Towards the Future Internet
G. Tselentis et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-539-6-247

The SENSEI Real World Internet Architecture

Vlasios Tsiatsis [a,1], Alexander Gluhak [b], Tim Bauge [c], Frederic Montagut [d],Jesus Bernat [e], Martin Bauer [f], Claudia Villalonga [g, d], Payam Barnaghi [b], Srdjan Krco [h]

[a] *Ericsson Research, Ericsson AB,* [b] *University of Surrey, UK,* [c] *Thales Research and Technology,* [d] *SAP AG, Switzerland,* [e] *Telefonica I+D,* [f] *NEC Europe Ltd.,* [g] *ETH Zurich,* [h] *Ericsson d.o.o., vlasios.tsiatsis@ericsson.com, a.gluhak@surrey.ac.uk, timothy.bauge@thalesgroup.com, frederic.montagut@sap.com, bernat@tid.es, martin.bauer@nw.neclab.eu, claudia.villalonga@sap.com, p.barnaghi@surrey.ac.uk, Srdjan.krco@ericsson.com*

Abstract. The integration of the physical world into the digital world is an important requirement for a Future Internet, as an increasing number of services and applications are relying on real world information and interaction capabilities. Sensor and actuator networks (SAN) are the current means of interacting with the real world although most of the current deployments represent closed vertically integrated solutions. In this paper we present an architecture that enables efficient integration of these heterogeneous and distributed SAN islands into a homogeneous framework for real world information and interactions, contributing to a horizontal reuse of the deployed infrastructure across a variety of application domains. We present the main concepts, their relationships and the proposed real world resource based architecture. Finally, we outline an initial implementation of the architecture based on the current Internet and web technologies.

Keywords. Real World Internet, Sensor and Actuator Networks, Resource, Resource Discovery, Context-awareness

1. Introduction

Internetworking physical world objects and integrating them into the network of the future is one of the most important aspects of a Future Internet [7]. The objective of our research is to enable this real world dimension of future networking. Wireless Sensor and Actuator Networks (WS&ANs)[6], ubiquitously deployed at the edges of the networks, will provide an infrastructure that enables the augmentation of the physical world and interaction with it, without the need for direct human intervention. Sensing interactions are concerned with capturing information about the state of the physical world and making it available as contextual information at different levels of granularity. This information can be further augmented and processed by the core infrastructure before being delivered to interested consumer endpoints. Actuation interactions, on the other hand, enable actions upon the real world that may have a

[1] Vlasios Tsiatsis, Ericsson Research, Ericsson AB, Faeroegatan 6, SE-16480, Stockholm, Sweden

direct or indirect impact on the physical state of the real world objects which can range from simple activations to complex control loops.

In this paper we present the architecture [2][4] of a framework designed to enable efficient integration of smart objects, which has been developed as part of the European ICT-FP7 SENSEI [10] project. Building on top of the communication service layers of the current or future Internet, our architecture weaves heterogeneous SAN and processing resources into a homogeneous resource fabric, enabling an open market space for real world context and interaction. The underlying design principles of our architecture [1] are heavily influenced by service oriented architecture [9], the REST architectural style [3] and evolutionary design [8].

Our contribution is the adaptation of these principles and the synthesis of these concepts into a coherent architecture. Another contribution described in this paper is the realisation of the architecture, which includes the specification and design of a set of key enabling services and an initial implementation of those based on current Internet and web technologies.

1.1. Key Innovations

The proposed architecture offers basic sensor, actuator and processing services with which a user can access sensor data, control actuators or process raw sensor data. This is done by a user contacting directly a communication endpoint locator or address (e.g. an IP address) for a sensor, actuator or processing element. For users that do not have these communication endpoint locators the proposed framework offers discovery services of varying type of sophistication. The simple discovery services in our framework are not different from other relevant systems such as the Open Geospatial Consortium (OGC) SWE [15] (e.g. Sensor Observation Service). However, our framework builds on these primitive types of services to offer advanced discovery services as well as context and actuation management services by providing an abstraction level corresponding to the real world consisting of Real World Entities or Entities of Interest (EoI). An EoI represents an element of the real world such as a person, place or object. These entities are linked to sensor or processing services that provide information about the real world. Similar models are introduced to allow sophisticated actuation tasks and control loops on top of primitive sensor, actuator and processing services. Given the fact that such context services and actuation tasks can include sophisticated processing of sensor data and actuation commands that is not always known at the time of the user request, SENSEI offers architectural support for dynamic service composition. The semantically-rich models and descriptions of sensors, actuators and processing elements are designed to enable machine processing that facilitates the dynamic composition and instantiation of new services. To the best of our knowledge the context model, context services, actuation tasks and dynamic service composition of both primitive and advanced services is the first of its kind for the realization of the Real World Internet.

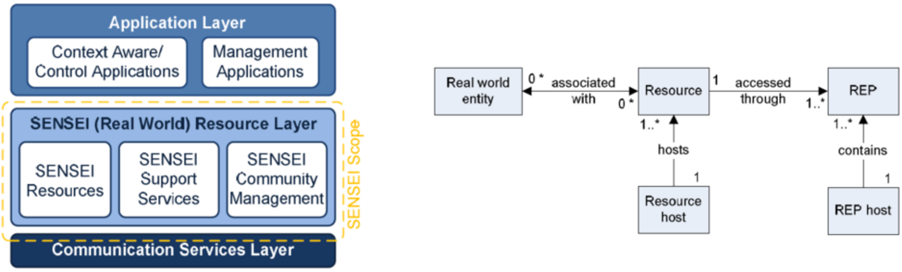
A comprehensive Authentication, Authorisation and Accounting (AAA) solution has been specifically designed to provide more efficient access to WS&AN resources than possible with existing approaches.

In today's mobile networking environment, one cannot exclude the possibility of the providers of context or control services of being mobile. Mobility of sensors and actuators is an assumption that a framework such as the one described in this paper factored in the framework architecture. As a result, our framework offers an innovative mobility management mechanism residing in a higher layer than the network mobility management and operates in a complementary fashion.

2. Architecture Overview

2.1. High level overview

On a high level the SENSEI architecture follows a layered model with three major horizontal layers, as shown in Figure 1(a). It extends the Internet architecture by adding a (Real World) Resource Layer on top of the Communication Service Layer, which forms the connectivity substrate of the current or Future Internet. The Resource Layer encompasses all necessary functions that facilitate the interaction of applications and services with the Real World Resources. The concept of the Resource is central to the proposed architecture as it provides a unifying abstraction for simple devices such as sensors, actuators, processors or software components (e.g. data fusion). Apart from the Resources, the Resource Layer encompasses Support services that enable discovery, composition and dynamic creation of Resources and support long term interactions (sessions). Furthermore it includes a set of functions that provide identity management for all entities in the system, consumer and provider account handling, privacy and security features summarized as the Community Management.



(a) (b)

Figure 1 : High level overview of the SENSEI Architecture and SENSEI core concepts

The separation between a Resource Layer and the Communication Services Layer allows mapping of the Communication Services Layer either to a connectivity substrate based on the evolution of the current Internet for near-term deployment or to a more revolutionary clean slate design of the underlying network infrastructure.

On the Application Layer multiple diverse applications and services gain unified access to real world information and interaction capabilities via interfaces provided by the Resource Layer. The Resource Layer also aims to support system management applications in a unified manner. As a result the Resource layer enables horizontalisation through the reuse of constrained devices such as sensors or actuators by a multiple different types of applications.

This paper introduces the resource concept and focuses specifically on the support services. For information on other aspects of the architecture readers can refer to [16].

2.2. The resource concept

The resource concept lies at the heart of our architecture [2][4]. Resources provide capabilities such as sensing, actuation, processing of context and sensor data or actuation loops, and management information concerning sensor/actuator nodes, gateway devices or entire collections of those that form a common administrative domain.

Figure 1 (b) provides an overview of the key concepts that relate to the resource concept. A resource has an embodiment in the physical world, the Resource, which could be sensor, actuator a processing component, or a combination of these. A Resource End Point (REP) is the software process which implements one or more Resource Access Interfaces (RAIs) to the resource. A Resource Access Interface is used by a user (referred to as Resource User or RU) to access the services that a resource provides (e.g. sensor data, actuator commands). A REP is uniquely addressable within the real world resource layer. Differentiation between, REP and resource and the roles of REP host and Resource host allows a separation of concern between different required mechanisms in the architecture. The device hosting a resource is referred to as the resource host. A REP host is a device that executes the software process representing the REP. Resources may be associated with one or more Real World Entities (e.g. person, place or objects) for which they can either provide information or for provide control over their surrounding environment.

3. Support Services

3.1. Resource layer components

The Support Services are depicted in Figure 2. For simplicity we depict the REP as being *inside* the Resource. Resource Users are typically context-aware applications that require real world information, applications that require real world actuation, management applications that are used for the management of one or more WS&ANs, or composite Resources that make use of other Resources. A Resource User interacts with a Resource via the Resource endpoint (REP) using the Resource Access Interface (RAI) [11].

The Resource Directory implements a loosely coupled rendezvous mechanism between Resource Users and Resources. In order to be discoverable by Resource Users in the Resource Layer, a Resource must be registered with the Resource Directory [2]. Using the Resource Publication Interface (RPI), the Resource Publication function (RP), that the resource provider controls, registers one or more Resources with the Resource Directory. This information is maintained for lookup within a structured Resource repository. Using the Resource Lookup Interface (RLI), Resource Users can interact with the Resource Lookup function to discover Resources of interest in the Resource Layer.

The Entity Directory maintains the associations and dependencies between Real-World Entities and Resources. The Entity Directory complements the Resource Directory in that the focus of the Resource Directory is on Resources themselves while the Entity Directory focuses on Entities. To be discoverable through the Entity Lookup Interface (ELI), a link between a Real World Entity and a Resource has to be registered using the Entity Publication Interface (EPI) through the Entity Publication function (EP) [11].

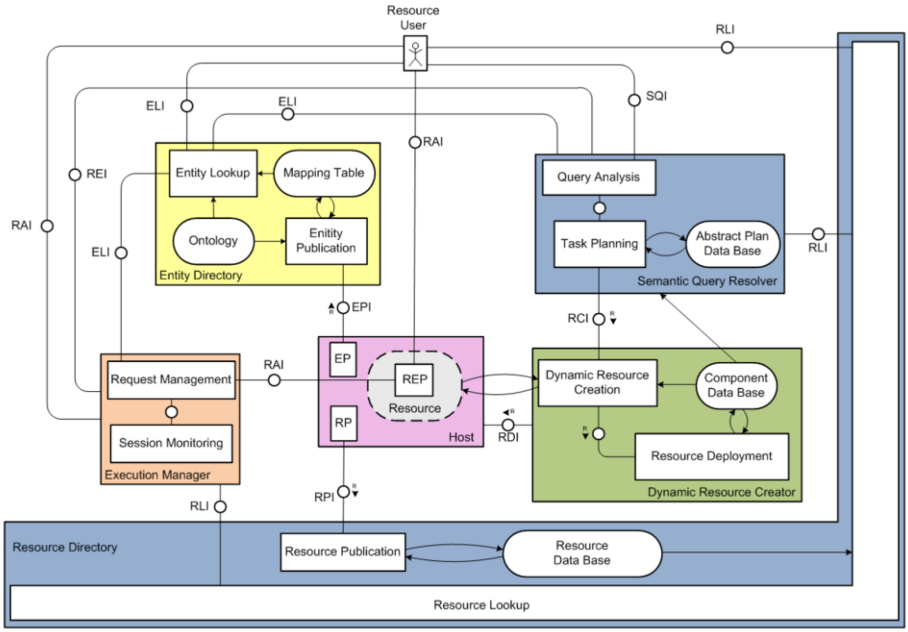


Figure 2 : Support Services in the Resource Layer Architecture

The Semantic Query Resolver receives queries from Users interested in particular Resources or Entities of Interest. High level declarative queries can be submitted via the Semantic Query Interface (SQI). These queries are analysed by the Query Analysis function and decomposed in (a set of) Resources and a corresponding execution plan with the help of the Task Planning function. An Abstract Task Plan Database can store templates to assist this process. The Semantic Query Resolver builds on the information available in the Resource Directory and uses the RLI interface for interactions.

In the case that no adequate Resource exists to answer a high level query of a Resource User, the Semantic Query Resolver can trigger the Dynamic Resource Creator to create the required Resource via the Resource Creation Interface (RCI) on the fly. A Dynamic Resource Creation function creates a new Resource that may be based on available Resources in the system and components (e.g. algorithms) from the Component Database. The newly created Resource is instantiated and deployed by the Resource Deployment function at a Resource Host using the Resource Deployment Interface (RDI). At the same time a new REP is also instantiated. The host registers the newly created Resource with the Resource Directory and the associated Real World Entity to the Entity Directory through the RP and the EP, respectively.

The functions of the execution management system component are implemented by the Execution Manager [11]. Using the Request Execution Interface (REI), the Semantic Query Resolver can invoke the Execution Manager to process requests for real world context and actuation tasks. A Request Management function keeps track of incoming queries and invokes appropriate Resources. Additionally, it can set up sessions between Resource Users and Resources using the RAI in case of long lasting requests (e.g. events). In this case, the Session Monitoring performs the adequate adaptation when resource availability changes in the system.

3.2. System Interactions

To illustrate the operation of the system, this section provides a brief description of some example use cases with the required system interactions. The system architecture supports a wide range of different resource users with different requirements. It provides support at design time and, more importantly, at runtime.

In the basic scenario, resource users need to interact with a fixed set of known resources. These resource users assume a rather static setting with stationary sensor and actuator resources. This fits the more traditional setting where applications are directly deployed on top of wireless sensor networks. Here, the system supports the developer to find the required resources through Resource Directory (Figure 3, 1.1). The RAI description provides the required information for implementing the client-side. At runtime, the resource user directly interacts with the required resources (Figure 3, x.2) and the system only provides support functionalities such as authentication, authorisation, and accounting.

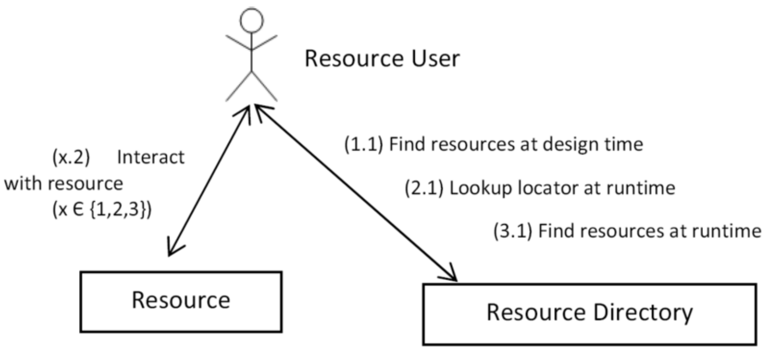


Figure 3 : Basic System Interactions

In cases where resources are mobile, their point of attachment to the network can change. This means that the REP locator through which they are reachable changes; however the resource identifiers remain unchanged. Whenever the network attachment changes, the resource host updates the REP locator information in the resource description stored at the Resource Directory. Using the resource identifier, resource users can look up the current REP locator of resource in the Resource Directory at runtime (Figure 3, 2.1) and interact with it as before (Figure 3, x.2). This is similar to the use of the Home Location Register in GSM, where information about the current attachment of a mobile phone is updated and can be looked up [13].

The system also supports cases where the set of resources is not known at design time, but has to be determined at runtime. For example, this is needed when the resource user is utilising a mobile device and needs to interact with available resources in its respective environment that provide the required information - which we expect to be a common case when we achieve our goal of horizontalisation. In this case the lookup functionality is utilised for discovering the required resources. In the basic scenario, the Resource Directory is queried for resources of interest (Figure 3, 3.1). To query a resource, users can submit a set of tags that are matched against the tags provided in the resource descriptions.

Resource users that are designed to operate across heterogeneous environments do not need to know the details of all resources in the different environments. For these resource users it is advantageous to specify what information they require or what actuation task they need to execute rather than having to know how this information is provided by different resources or how certain actuator resources execute an actuation task. The Semantic Query Resolver allows resource users to issue declarative requests specifying what context or sensor information is required or what actuation task is to be executed (Figure 4, 4.1). The Semantic Query Resolver then analyzes the request and finds resources that can be used to satisfy the request. In the planning step, the Semantic Query Resolver creates an execution plan. If all required resources do not exist, it is checked whether the Dynamic Resource Creator can create them (Figure 4, x.2, x.3); otherwise the request cannot be executed. Finally, the information about the required resource(s) can be returned to the resource user.

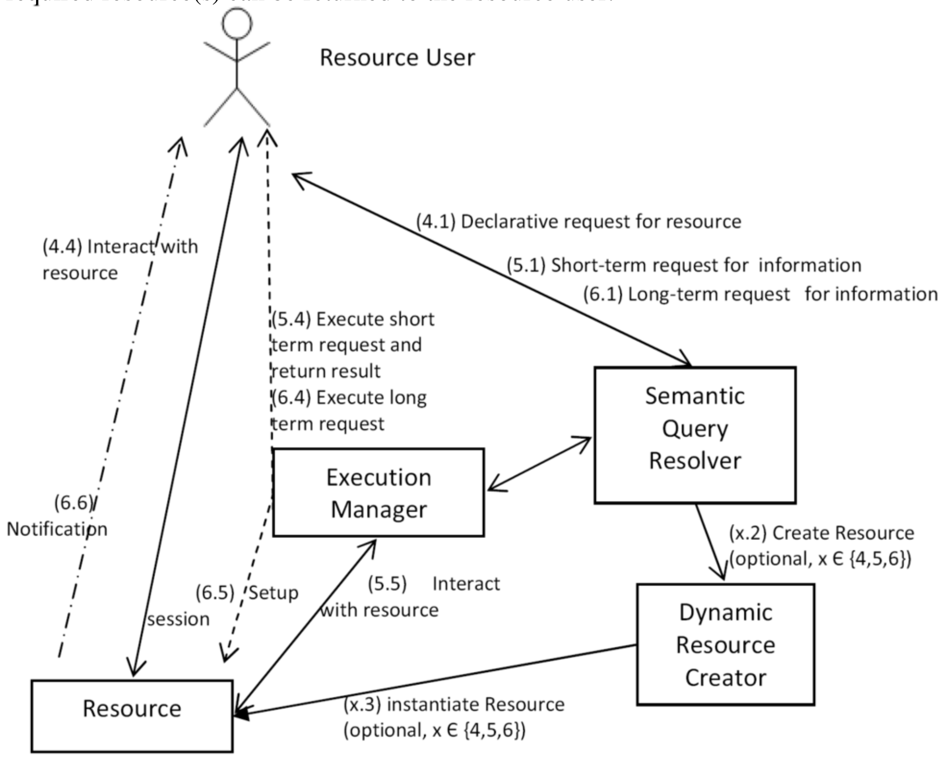


Figure 4 : Advanced System Interactions

In the described case, it would still be up to the resource user to interact with the resources directly for executing the request (Figure 4, 4.4). Alternatively, the Execution Manager can execute the request on behalf of the resource users (Figure 4, 5.4, 5.5). This simplifies the writing of high-level applications as they do not have to deal with the execution of the request itself.

In the simple one-time request scenario, the Execution Manager directly executes the request and returns the information. In the long-term subscription scenario, the Execution Manager sets up sessions [14] between the resources and the resource user according to the execution plan (Figure 4, 6.5). This separation of control and data flow leads to a more scalable system. The Execution Manager only handles the control aspects, whereas the data flow happens directly between the resources and resource users (Figure 4, 6.6) and thus it is completely decentralised.

For long-term requests, important aspects for the request execution may change over time, e.g., resources become unavailable, new resources become available, resources are no longer suitable because resource or resource user are mobile and have moved. It is important to ensure the continuity of information provisioning and actuation task execution in such cases. The Execution Manager allows setting up the monitoring of these aspects. If a relevant change is detected, the re-planning of the request by the Semantic Query Resolver is triggered. If these results are in a changed execution plan, the request execution will be changed without the need for an intervention of the resource user.

4. Realisation on the Current Internet

In this section we briefly describe a realisation of the SENSEI architecture which makes use of the existing web technologies to implement the main service functions of the architecture. The implementation is based on a RESTful design [3] using the RESTlet framework [5]. Each component is available as a REST resource on the web, identifiable by a unique URI and accessible through the HTTP protocol main primitives: GET, POST, PUT, DELETE that allow manipulation of the resource (get information about the resource, set or update the resource parameters or delete resource).

The first version of the implementation illustrates simple scenarios of the Resource Layer using a set of heterogeneous resources, framework components involving the Semantic Query Resolver, the Entity Directory, the Resource Directory and resource users as depicted in Figure 5. The resources represent a mix of native SENSEI WS&AN islands, legacy WS&ANs and mobile phone based gateway:

- A 6LowPAN enabled sensor network represents a native SENSEI WS&AN. Each sensor node is able to directly host an end-point (i.e. REP) for the resource it provides. End-to-end transparent access is enabled with the implementation of Binary Web Service protocol. This uses Efficient XML Interchange (EXI) encoding to compress the exchanged messages for efficient transfer through the WSN.
- A WS&AN island based on ZigBee protocol stack is used as a legacy type of resource. We enhanced a Zigbee gateway to host a Resource End Point for each available Zigbee sensor and provide resource and Entity Publication functionality.
- An Android phone served as an example of a mobile platform, hosting a Resource End Point for its built-in sensors.
- A context processing server hosted composite processing resources that were able to combine input form multiple Resource End Points to provide higher layer context information.

Context aware applications, acting as resource users, are implemented on an application server, which provides a browser based access interface to fixed and mobile clients.

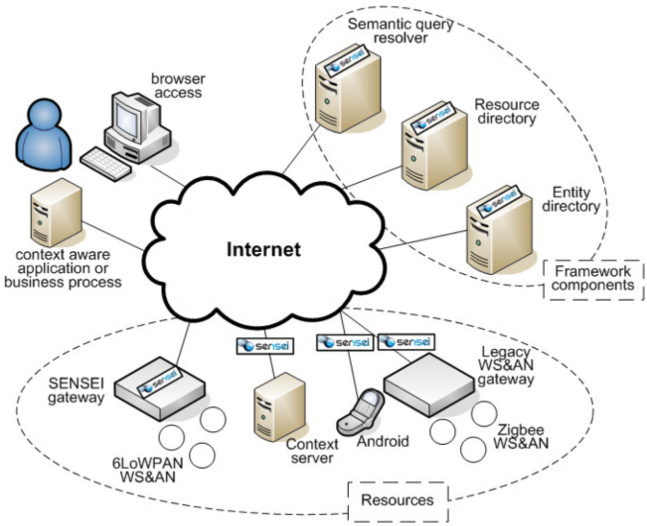


Figure 5 : Setup of test deployment

A typical interaction in the setup involved a context aware application issuing a semantic query to the Semantic Query Resolver. Upon receipt of a request, the query is analysed and the relevant Real World Entity is identified. The Semantic Query Resolver looks up the Entity Directory in order to retrieve the Resource associated with the latter Entity of Interest. The Resource Directory is then queried to retrieve the URI of the relevant resource. The latter information is finally sent back to the client who can in turn invoke the adequate resource.

5. Conclusions and Outlook

This paper presents the design and initial implementation of an architecture that provides the enabling foundation of a Real World Internet. The architecture has been developed as a part of a large co-ordinated effort in Europe to create an underlying architecture and corresponding services for the Future Networked Society. The architecture provides the necessary functionality to enable a range of services for accessing sensor data, actuator and processing commands as well as offering context management for real world entities. Our architecture being the results of both technical and business-oriented research enables a Real World information marketplace and the creation of a business ecosystem around Real World Services.

We have demonstrated the feasibility of the Real World Internet concept and an initial realisation of the key architectural components in lab deployments. The work is ongoing to enhance the existing components, complete the implementation of the architecture with new components and evaluate the presented architecture and its services on larger scale. For this purpose a PAN European testbed is currently being created, integrating heterogeneous sensor and actuator networks across the facilities of 19 partners, including sensor network testbed deployments from ISSNIP, an affiliated Australian research network.

Acknowledgment

This paper describes work undertaken in the context of the SENSEI project, Integrating the Physical with the Digital World of the Network of the Future (www.sensei-project.eu). SENSEI is a Large Scale Collaborative Project supported by the European 7th Framework Programme, contract number: 215923.

References

[1] A. Gluhak and M. Bauer and M. Johansson and J. Bernat and V. Stirbu and F. Montagut and M. Presser. Towards an Architecture for a Real World Internet, “Towards the Future Internet – A European Research perspective. IOS Press, 2009.
[2] F. Carrez (Editor), T. D.3.2 –Reference Architecture. SENSEI, Public Deliverable D.3.2, 2009, http://www.sensei-project.eu/.
[3] R. T. Fielding, and R. N. Taylor, Principled design of the modern Web architecture. ACM Trans. Internet Technol., 2(2):115--150, 2002.
[4] T. Bauge (Editor), D.3.3 - Components for End-to-End Networking, Management and Security and Accounting. SENSEI, Public Deliverable D.3.3, 2009, http://www.sensei-project.eu/.
[5] RESTlet framework. http://www.restlet.org/.
[6] I. F. Akyildiz,, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. Comput. Netw., 38(4):393--422, 2002.
[7] D. Clark, C. Partridge, R. Braden, B. Davie, S. Floyd, V. Jacobson, D. Katabi, G. Minshall, K. Ramakrishnan, T. Roscoe, I. Stoica, J. Wroclawski, and L. Zhang. Making the world (of communications) a different place. SIGCOMM Comput. Commun. Rev., 35(3):91--96, 2005.
[8] D. Clark, J. Wroclawski, K. Sollins, and R. Braden . Tussle in cyberspace: defining tomorrow's internet. IEEE/ACM Trans. Netw., 13(3):462--475, 2005.
[9] T. Erl. Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall, 2005.
[10] ICT FP7 SENSEI Project: http://www.sensei-project.eu/
[11] M. Rossi (Editor), “D.2.3 - Preliminary Context Model, Interfaces and Processing Mechanisms for Sensor Information Services”, SENSEI Public Deliverable D.2.3, 2009, http://www.sensei-project.eu/.
[12] Z. Shelby, M.I. Ashraf, M. Luimula, J. Yli-Hemminki, and A.P. Castellani, "BinaryWS: Enabling the Embedded Web" submitted to EWSN2010, FWN2010
[13] M. Mouly and M. Pautet. The GSM System for Mobile Communications. Telecom Publishing, 1992
[14] M. Strohbach, M. Bauer, E. Kovacs, C. Villalonga and N. Richter. Context Sessions - A Novel Approach for Scalable Context Management in NGN Networks. Proceedings of Workshop on Middleware for Next-generation Converged Services and Applications (MNCNA), held in conjunction with Middleware 2007, 2007.
[15] M. Botts, G. Percivall, C. Reed and J. Davidson. OGC Sensor Web Enablement: Overview and High Level Architecture. Technical report, OGC, 2007.
[16] T. Bauge (Editor), “D.3.3 - Components for End-to-End Networking, Management and Security and Accounting”, SENSEI, Public Deliverable D.3.3, 2009, http://www.sensei-project.eu/.