On the Network Proximity in City-Scale Ubiquitous Systems

Manfred Sneps-Sneppe[1], Dmitry Namiot[2]

[1] Ventspils University College
Ventspils International Radioastronomy Centre
Ventspils, Latvia,
{ manfreds.sneps@gmail.com }

[2] Lomonosov Moscow State University
Faculty of Computational Mathematics and Cybernetics
Moscow, Russia
{ dnamiot@gmail.com }

Abstract. This paper discusses the city-scale context-aware mobile information system and programming interfaces - CityProximus. Our project is based on the ideas of network proximity. De-facto, network related information is the easiest way for getting context-related information on mobile devices. It lets us replace geo-information in location based services with data about available networks and network nodes. So, in the proposed system user-defined information could be directly associated with existing and specially created network nodes. Later, this information could be delivered to mobile users being in the proximity to the above-mentioned network nodes. In this paper, we discuss the technical aspects of implementation for such system on the city level.

Keywords: Bluetooth; Wi-Fi; context-aware; network-proximity; smart city.

1 Introduction

The classical definition for the term 'context-aware' [1] describes context as a location, identities of nearby people and objects, and changes to those objects. Therefore, we consider context-awareness in the context of proximity. There are several definitions from other authors, but most of them define context awareness as a complementary element to location awareness. The location serves as a determinant for the main processes, and context adds more flexibility with mobile computing and smart communicators [2].

There are many practical use cases, where the concept of the location can be replaced by that of proximity. On one hand, this applies to use cases where the detection for an exact location is difficult, even impossible, or not economically viable. Very often, this replacement is related to privacy. For example, a privacy-aware proximity-detection service determines if two mobile users are close to each other, without requiring them to disclose their exact locations [3]. Proximity can be used as a main formation for context-aware browsers [4]. The context-aware browser will reveal data chunks depending on the current context.

The usage of network proximity for context-aware systems is very transparent. At this moment, network modules in mobile phones are most widely used "sensors". All smartphones nowadays have Wi-Fi (Bluetooth) modules and Wi-Fi (Bluetooth) related measurements (e.g., visible Wi-Fi access points and Bluetooth nodes, signal strength) are standard. So, if treat the above-mentioned data (nodes visibility, signal strength) as measurements, we can treat network modules in mobile devices as sensors. By the definition, Bluetooth signal, for example, is local. So, if any Bluetooth node is visible from a mobile device (a mobile phone, for example), then this device is somewhere nearby that node (so-called Bluetooth distance). The same is true for Wi-Fi. And this proximity information (network proximity) can replace location data. There are two main reasons for this replacement. At the first hand, we can target here all indoor application [5]. Obtaining GPS (Global Positioning System) data indoor is not reliable and sometimes even impossible. At the same time, modern offices have a number of wireless nodes. The second reason is much more interesting. The wireless node could be moveable. So, our context-aware information linked to some moveable wireless node will "follow" to object's movements.

For network proximity-based context-aware applications, any existing or even especially created Wi-Fi hot spot, or Bluetooth node, could be used as a presence-sensor that can play the role of a trigger. This trigger can open access to some content, discover existing content, as well as cluster nearby mobile users [6,7]. In this paper, we present CityProximus projects – network proximity system on the city-level. This project has been presented to H2020 program [8].

The rest of the paper is organized as follows. In Section 2, we discuss related works. In Section 3, we present working projects within CityProximus. In Section 4, we discuss the open issues and challenges.

2 Related works

At the first hand, our proposal for H2020 is based on our own projects for network proximity. We can mention here Spot Expert [2] and Bluetooth Data Points [9]. Spot Expert presents an expert system for existing and especially created Wi-Fi access points. Mobile users will be able to see user-defined content depends on the visibility of Wi-Fi access points. The typical rule in this systems looks so [2]:

```
  IF Access_Point with SSID Café IS visible AND RSSI
(signal strength) is within the given interval THEN
  {activate some predefined content for mobile users}
```

So, this service contains the following components:

- The database (data store) with the productions (rules) associated with Wi-Fi networks;
- The rule editor. It is a web application (including mobile web) which lets users add (edit) rule-set, associated with Wi-Fi networks;
- The mobile application (context-aware browser). This browser detects Wi-Fi networks, checks the current conditions against the database of productions and executes rules (activates content).

Figure 1 illustrates the typical output from context-aware browser:

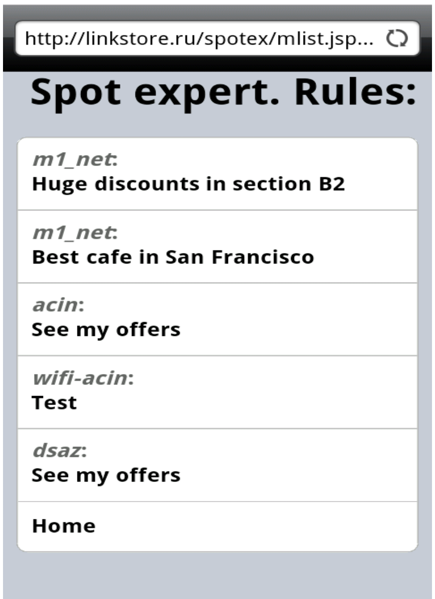

Fig.1 SpotExpert context-aware browser

By the similar principles, it could be done for Bluetooth nodes in the discoverable mode [9]. Bluetooth Data Points (BDP) are Core Bluetooth nodes in the discoverable mode which have some data associated with them.

The next example is an iBeacon from Apple [10]. The iBeacon is a tag with Bluetooth Low Energy chip [11]. The iBeacon simply broadcasts a presence message once per second to other devices within range of the Bluetooth radio.

SITA [12] provides common use Beacon Registry. The Beacon Registry is a registry of common use iBeacons for the Air Transport Industry. The registry provides the following services:

It allows beacon owners (airlines, airports or 3rd parties) to manage their beacon infrastructure and track where they are placed in an airport.

It enables airports to monitor beacon deployment to prevent radio interference with existing Wi-Fi access points

It provides beacons owners with a simple mechanism to set the 'meta-data' associated with beacons.

It provides an API for app developers who want to use these beacons for developing travel and other related apps [12]. It is illustrated in Figure 2.

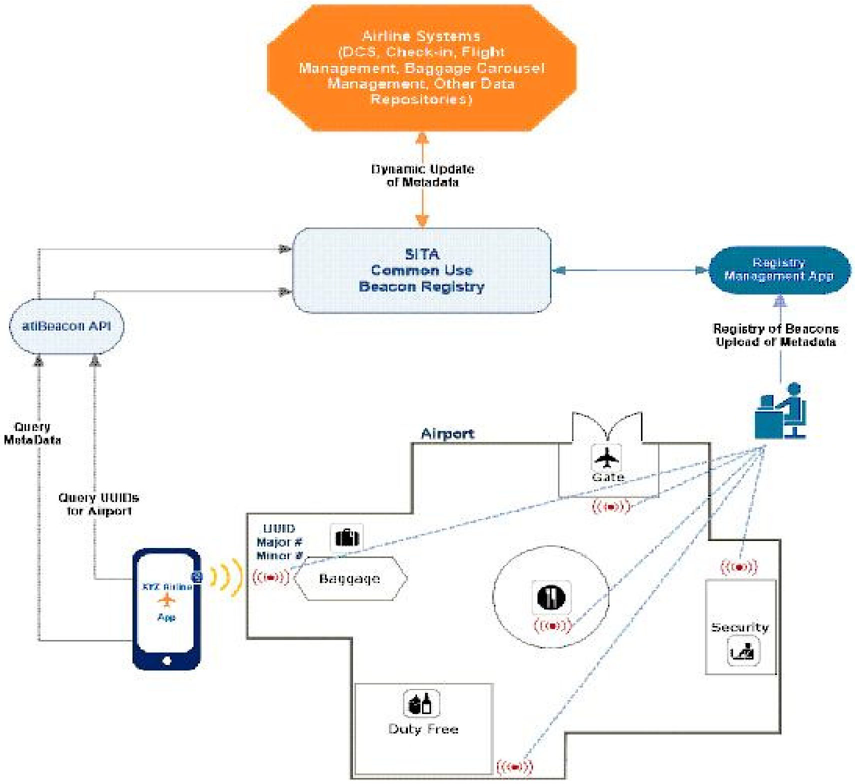

Fig. 2. SITA wireless tags infrastructure [12]

The next example is Navizon Proximity Engine [13]. It triggers proximity-based actions for registered users. This engine is a system designed for geo-fencing and enables delineating a virtual boundary around an area of interest. The engine triggers actions when known mobile devices enter or exit this area. A wide range of actions can be triggered, such as sending SMS messages to smart phones, delivering media to a user's device at an exhibit, opening doors activated by a Wi-Fi tag, sending email, or displaying a personalized message on an electronic billboard [13]. It is illustrated in Figure 3.

Navizon uses so-called passive Wi-Fi monitoring. It analyzes Wi-FI beacon frames. When a Wi-Fi device operating in scanning mode enters the target area, its MAC address is automatically acquired by the engine and uploaded to the cloud. The cloud-based software will identify any registered devices associated with a category defined for that location and trigger the corresponding action. Note that this approach depends on the scanning devices. Another problem could be associated with the newest iOS devices. Apple can use random MAC-address for beacon frames. This decision eliminates on practice personal identification for passive monitoring. And, of course, this functionality depends on the special nodes for collecting beacon frames.

Our approach is infrastructure-less. On the city-level, we simply cannot rely on installed tags.

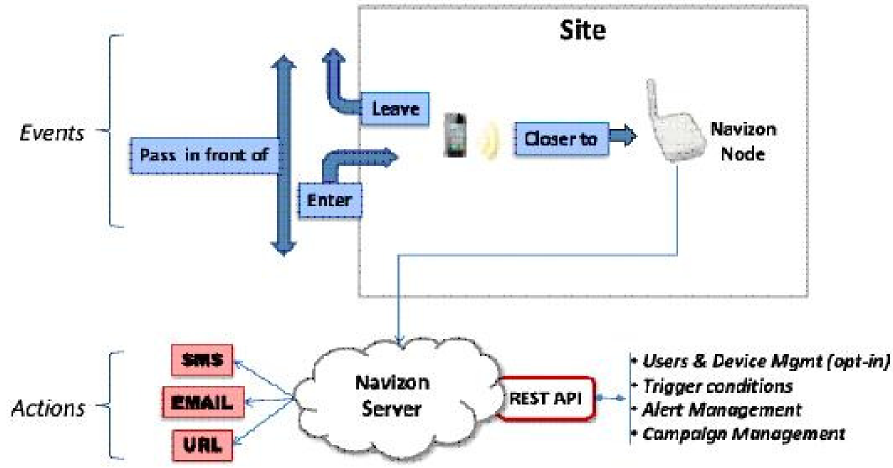

Fig.3 Navizon Proximity Engine

In our system, we do not use dedicated tags. The idea is to rely completely on the existing wireless nodes as well as on the nodes, especially created by mobile users on their own devices for sharing local data.

3. CityProximus

According to our previous development, CityProximus will include the following elements:

- a crowd-sensing system for collecting a city-level database of wireless networks;
- a data persistence solution (database) for saving and maintaining wireless networks data;
- an editor for rule settings;
- a context-aware browser

Formally, our proposal to H2020 includes the following working packages:

- Description of the current state-of-the-art of technology and protocols (deliverable overview): technological alternatives, wireless technologies, communication protocols and data structures.

- Analysis of Smart Cities Enablers from FI-WARE project and de-facto standards (e.g. Bluetooth low energy).

- CityProximus core concepts, requirements and architecture draft including Open Call requirements.

- Analysis of micro-service approach versus monolithic architecture, communications in Micro-services architecture.
- Smart Cities standard proposals: Data Programming Interfaces for Smart Cities.
- Machine to Machine (M2M) communications security standards.
- M2M communications wireless network infrastructure security problems (the use of wireless network infrastructure to emit presence information and to explore the environment without (application) data exchange).
- Network proximity-based services security (against attackers, network nodes used for a proximity analysis, authentication, Car-2-Car (C2C) applications).
- End-To-End Security Architecture in the context of SmartCity infrastructure.
- The current state-of-the-art of Open source libraries for mobile phone sensing (e.g. Funf Open Sensing package).
- Development of data mining tools (Discovery of Convoys in Network Proximity Log, Trajectories in the proximity log).
- Network proximity measuring mathematical tools.
- The specification of the smart city service.
- Socio-economic factors affecting the Smart cities network proximity services.
- Network proximity services socio-economic impact and sustainability assessment.

For collecting wireless info data, we are planning to use a custom version of Funf package [14]. A detailed review of existing Open Source crowd-sensing frameworks is provided in our paper [15]. As per data persistence, our initial experiments were organized with Apache Accumulo [16]. The simplest formation for proximity based rules is a visibility for the particular wireless node. In other words, each rule looks like this:

```
IF (Node_is_visible) THEN
{  activate some content }
```

Actually, iBeacons from Apple follow this model. The visibility for any node could be described by its MAC-address. It means that any rule could be presented in the following form:

```
MAC-address -> content for activation
```

Rules will be described individually for the each wireless node. So, we have a key (MAC-address) and a vector of data chunks (texts, images, etc.) It is a typical key-value data model. This data model is one of the most used models for NoSQL approach. And Apache Accumulo is a distributed key-value store. The details for NoSQL data model for network proximity are presented in our paper [17]. In CityProximus project, we are planning to use data persistence solutions from Fraunhofer FOKUS [18]. In order to ensure the interoperability across data sources, Fraunhofer FOKUS is going to research and develop formal knowledge structures based on semantic for the sensor data and computing and communication resources. Appropriate ontological structures will be adapted or created, to provide the backbone of the semantic unification of physical and virtual devices and resources and services.

Existing foundational ontologies and design patterns will be reused and extended towards semantic interoperability among the various entities, to effectively capture and interlink contextual information across all layers of the smart city platform, which will be a significant advancement over existing syntactic interactions.

As a tool for data storage, we suppose to use TheOpenCityDatabase (OCDB), Fraunhofer FOKUS – an important enabler of FI-Content (as a part of FI-WARE project) [19], and Open Machine Type Communications (OpenMTC) Platform [20] from the same source (Figure 4).

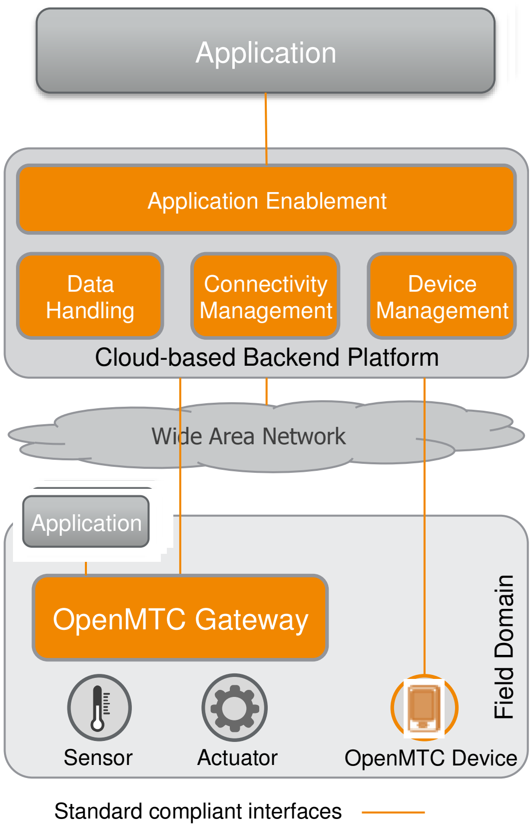

Fig. 4 OpenMTC

The OpenMTC provides the most open, cloud-enabled solution for conducting applied research activities and developing innovative M2M applications. The OpenMTC consists of two common capability layers that can be deployed in the front-end as M2M device or M2M gateways or in the infrastructure as a cloud platform. The OpenMTC features are aligned with M2M-related standards defined by oneM2M, ETSI, OMA, and 3GPP groups. The platform can be applied to many different vertical domains such as industry 4.0, smart city, eHealth, logistic, etc.

Heterogeneous sensors and actuators are accessed via M2M gateways, which expose a unified view of the resources in the field domain towards the infrastructure platform, using various wired or wireless communication technologies. Both front-end and backend nodes support common service capabilities that can be used by the applications through open interfaces.

There are mainly three features:

- Connectivity Management: Where we support various transport protocols such as HTTP, CoAP, WebSocket and (soon) MQTT, secure transmission based on TLS and DTLS, Diameter protocol to interwork with 3GPP network-based components such as PCRF and HSS.
- Device Management: Where front-end devices (gateways, sensors, and actuators) are managed through a central component on the backend following OMA lightweight M2M DM.
- Data Management: where data is processed and aggregated on the front-end and backend.

The OpenMTC offers SDK to access the OpenMTC capabilities through REST APIs compliant with ETSI M2M and oneM2M specifications. The SDK allows developing applications for the front-end (running e.g. on OpenMTC devices or gateway) or on the top of the backend platform.

In addition, the project is going to use the OpenIoT platform [21] to ensure semantic interoperability. The OpenIoT project has developed the OpenIoT platform as open source middleware for Internet of Things (IoT) and cloud integration, which is based on the integration of W3C SSN compliant IoT/data streams. OpenIoT is also a showcase of the merits of the semantic interoperability in the IoT domain. It will be reused as a basis for realizing the semantic interoperability of diverse data sources on different levels.

4. The challenges

In this section, we would like discuss the challenges for network proximity tasks.

If network proximity-based services don't use any kind of authentication there might be an incentive and in particular possibilities for attackers to inject false information into the system. A node could either inject bogus network node availability information or it could selectively affect the availability of other network nodes used for a proximity analysis. In either case, the context "seen" by a node is not authentic.

A network proximity scan is likely to gather MAC-addresses of network nodes, even though they are probably not directly relatable to real-life identities. Similarly to Car-2-Car (C2C) applications this might allow for tracking of individual nodes and thereby for violating privacy requirements of users. It might be interesting to study the applicability of C2C-style pseudonymization techniques [22] for network proximity-based services.

For the type of communication mentioned above, the lightweight security mechanisms are desirable that allow for a secure (encrypted, authenticated, and integrity checked) communication between network nodes directly and from network nodes to the backend system. Since nodes might have significantly constrained resources the overhead (power, bandwidth, CPU, and memory usage) must be as small as possible. Particularly in dynamic networks like the ones constituted by SmartCity infrastructure connectivity is expected to suffer from interruption. Thus, the applicability of DTLS-like security mechanisms [23] could be analyzed in the context of SmartCity infrastructure.

5. Conclusion

In this paper, we present CityProximus projects devoted to city-level information systems, based on the network proximity. The proposed project will deploy and develop tools for information gathering based on existing participatory and opportunistic sensing approach, where citizens, with participatory sensing, the costumers choose an application out of their interests and actively collect the proximity data, process them and take decisions. This approach keeps the ability to the opportunistic sensing too, where the citizens may not be voluntary directly in the sensing activity, but their own devices or external device decide when sensors and actuators are needed to be activated. The recent proliferation of mobile devices equipped with their own sensors and functionalities (e.g., gyroscope, compass, accelerometer, proximity sensor, front- and back-facing camera, microphone, GPS, Wi-Fi, and Bluetooth interfaces) has fostered the emergence of more sophisticated Smart Cities applications. In addition, intermediate network nodes (such as M2M gateway or WiFi access points) can periodically scan the network proximity passively and provide additional information to the system. The collected information will facilitate the creation of real-time information space for big-data analytics based on multiple sources managed either by local devices or backend platform.

6. References

[1] G. Schilit and B. Theimer Disseminating Active Map Information to Mobile Hosts. IEEE Network, 8(5) (1994) pp. 22–32

[2] D. Namiot “Context-Aware Browsing -- A Practical Approach”, Next Generation Mobile Applications, Services and Technologies (NGMAST), 2012 6th International Conference on, pp. 18–23, DOI: 10.1109/NGMAST.2012.13

[2] Sundmaeker, H., Guillemin, P., Friess, P., Woelfflé, S. (eds.): Vision and Challenges for Realising the Internet of Things, CERP-IoT, March 2010. European Commission, Brussels (2010)

[3] L. Šikšnys, J. Thomsen, S. Šaltenis, and M. Yiu Private and Flexible Proximity Detection in Mobile Social Networks, Mobile Data Management (MDM), 2010 Eleventh International Conference on, 2010, pp. 75–84.

[4] Namiot, Dmitry. "Network Proximity on Practice: Context-aware Applications and Wi-Fi Proximity." International Journal of Open Information Technologies 1.3 (2013): 1-4.

[5] Namiot, Dmitry. "On Indoor Positioning." International Journal of Open Information Technologies 3.3 (2015): 23-26.
[6] Namiot, Dmitry, and Manfred Sneps-Sneppe. "Geofence and network proximity." Internet of Things, Smart Spaces, and Next Generation Networking. Springer Berlin Heidelberg, 2013. 117-127.
[7] Sneps-Sneppe, Manfred, and Dmitry Namiot. "Spotique: A new approach to local messaging." Wired/Wireless Internet Communication. Springer Berlin Heidelberg, 2013. 192-203.
[8] Anastasi, Giuseppe, et al. "Urban and social sensing for sustainable mobility in smart cities." Sustainable Internet and ICT for Sustainability (SustainIT), 2013. IEEE, 2013.
[9] Namiot D., Sneps-Sneppe M. On Mobile Bluetooth Tags //arXiv preprint arXiv:1502.05321. – 2015.
[10] N.Newman, “Apple iBeacon technology briefing”, Journal of Direct, Data and Digital Marketing Practice, vol. 15, no. 3, 2014, pp. 222-225.
[11] C.Gomez, J. Oller, J. Paradells, “Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology”, Sensors, vol. 12, no. 9, 2012, pp. 11734-11753.
[12] SITA iBeacon Registry https://www.developer.aero/Beacon-Registry-API/API-Overview
[13] Navizon Proximity Engine https://www.navizon.com/product-navizon-hub
[14] Aharony, N., Gardner, A., Sumter, C., & Pentland, A. (2011). Funf: Open sensing framework.
[15] Namiot, Dmitry, and Manfred Sneps-Sneppe. "On Open Source Mobile Sensing." Internet of Things, Smart Spaces, and Next Generation Networks and Systems. Springer International Publishing, 2014. 82-94.
[16] J.Kepner, W.Arcand, W.Bergeron, N.Bliss, R.Bond, C.Byun, C.Yee, “Dynamic distributed dimensional data model (D4M) database and computation system”, In Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, pp. 5349-5352. IEEE.
[17] Namiot, D., & Sneps-Sneppe, M. (2015). On Mobile Bluetooth Tags. arXiv preprint arXiv:1502.05321.
[18] Schieferdecker, I., & Mattauch, W. (2013). ICT for Smart Cities: Innovative Solutions in the Public Space. Computation for Humanity: Information Technology to Advance Society, 127.
[19] The Open City Database http://wiki.mediafi.org/doku.php/ficontent.smartcity.enabler.opencitydatabase
[20] The Open Machine Type Communications (OpenMTC) http://www.open-mtc.org/index.html
[21] Kim, J., & Lee, J. W. (2014, March). OpenIoT: An open service framework for the Internet of Things. In Internet of Things (WF-IoT), 2014 IEEE World Forum on (pp. 89-93). IEEE.
[22] Stübing, Hagen. Multilayered security and privacy protection in Car-to-X networks: solutions from application down to physical layer. Springer Science & Business Media, 2013.
[23] McGrew, David, and Eric Rescorla. "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for Secure Real-time Transport Protocol (SRTP)." (2010).