Implementing a holistic approach for the Smart City

Roberto Requena, Antonio Agudo, Alba Baron,
Maria Campos, Carlos Guijarro, Jose Puche, David Villa, Felix Villanueva, and
Juan Carlos Lopez

School of Computer Science, University of Castilla-La Mancha,
Ciudad Real,Spain
{roberto.requena,jose.puche,alba.baron,
antonio.agudo,maria.campos,carlos.guijarro}@alu.uclm.es,
{david.villa,felix.villanueva,juancarlos.lopez}@uclm.es

Abstract. Extending the services offered by the city requires, most of the times, reimplementation efforts. This paper presents our on-going efforts to develop a platform for the Smart City that focuses in providing the appropriate solutions for an easy integration of new services and devices. This endeavor is accomplished by abstracting communication issues using a middleware platform and by standardizing the way services are instantiating.

1 Introduction

The Smart City paradigm is gaining credit as technology improvements are enabling new and more ubiquitous ways of interaction. In the Spanish context, several cities are leading the way towards the Smart City such as Malaga, Barcelona, or Santander. However, most of these cities make the emphasis in deploying *black boxes* of sensors and services. On the contrary, the Smart City paradigm envisions more versatile and flexible solutions than just a set of services attached to a specific set of sensors. A Smart City should tackle a broader objective as it is providing a basic platform where new devices and services could be easily deployed and integrated.

Ideally, new devices could be integrated in the Smart City by simply implementing a *plug-and-play* approach. Similarly, new services could be implemented, that would make use of the data gathered by those devices, in a standardized manner without having to be aware of the low-level details of every different device available in the Smart City.

However, this is not a trivial matter since several issues need to be addressed, such as a mechanism for abstracting communication between different protocols and technologies or a syntactic and semantic standardization to support an orthogonal instantiation of services.

This paper presents a holistic approach to the Smart City, called Civitas, that implements a bottom-up approach. The proposed solution consists in a layered framework, in which different functionalities are organized by levels of complexity. In this sense, the lowermost level deals with device-integration aspects. Then, an intermediary layer is taking care of device-communication issues, abstracting device-like particularities. Finally, the uppermost layer provides a set of services built upon the previous layers. Figure 1 depicts the different layers along with the different devices and services supported by each of them. Examples of every layer are provided underneath.

This paper is organized as follows. First, the state of the art for Smart City framework is studied, summarizing not only specific solutions for the Smart City but also comprehensive frameworks upon which devices and services can be deployed. This section will pose the most relevant challenges that need to be addressed from the point of view of the Smart City paradigm, as well describes the specific details of how standardization is accomplished, providing concrete examples of how this standardization is implemented in the different layers of the proposed architecture. Finally, the most relevant conclusions drawn from the proposed framework will be presented in Section 5.

2 Smart Cities platforms

Lately, a lot of services are appearing devoted to provide smart functionality in smart city field. There is still a lack of standards or platforms that we could take as facto standards in service development and specification.

One of the main efforts in smart city application field is SmartSantander project [7] where one of the most large testbed has been deployed in Santander (Spain). This project can be seen as an city-scale experimental facility which enables research in smart city field identifying key technologies, services and socioeconomic factors involved in smart city ecosystem.

One of the main issues of smart city is "information island" problem which appear if we develop services as standalone applications. Examples of this type of services are traffic management [8], human dynamics [9], bicycle routes [10], mobile crowdsensing [11], etc.

We can clasify the approach followed by these works since middleware point of view in the following solutions:

- A complete design solution from scratch focusing in the functionality of the service designed without middleware considerations. They do not usually deal with common problems in a real deployment (e.g scale, basic security mechanism, etc.)
- A middleware designed from scratch ignoring existent solutions.
- The use of specific protocols as a middleware, for example XMPP (Extensible Messaging and Presence Protocol) in [11], without any other consideration.

Our approach is different to any other, we take an efficient, robust, well-proved generic object-oriented distributed middleware and we add specific smart-city related layers. Currently we are working with Internet Communication Engine (ICE) [6], an object-oriented distributed middleware which provide us with basic security mechanisms, scalability mechanisms, event broker, etc.

With this starting point, we add what we consider essential common services for future smart city services (e.g. reasoning layer or visualization layer) and we integrate other protocols in order to provide a flexible framework for service development (e.g MQTT).

3 The Civitas Framework

Figure 1 shows an overview of Civitas framework, the lowest layer is a sensor/actuator layer since a lot of process in the city has to be monitorized and controlled by mean sensors and actuators. As we will see in the next subsection by mean a pollution service monitoring example, according to the type of service, a set of appropriate sensors/actuators are deployed in the city. The information from sensors/actuators, together with other type of information (e.g. from citizens), is gathered using an Information and Communications Technologies (ICT) Infrastructure.

We strongly believe that, similarly to services which are provided using common infrastructures (electric/water supply network, transport networks, etc.), an smart city should provide with an ICT layer to help service and device deployment. At logical level, Civitas backbone layer has similar aim that ICT layer does at physical level. In the Civitas backbone, information is gathered and distributed according to each application field (Homeland Security, Waste management, social media, etc.). These application fields, in turn, are divided into specific services (semaphore control, pollution monitoring, traffic monitoring, parking management, etc.)

Over these application field Civitas offers a reasoning and analytics layer in order to deal with information extraction from raw data. Again, this layer is specialized by application field and the information is provided to different entities (Government, corporations and citizens) which also provide with information.

Finally, an advanced visualization and control layer enables human advanced interaction in order to monitor and to control different process in the city.

Every layer and service in Civitas exposes one or several object-oriented interfaces in order to interact with its functionality. We describe these interfaces by mean an Interface Description Language (IDL) which provide us with an excellent tool to specify a "contract" between the developer who provides the service and the developer who uses the service.

Additionally, as we will see later and in order to easily integrate sensors from different partners, in the sensor/actuator layer we also support other protocols.

About the infrastructure of the smart city we are deploying a set of Service nodes with four interfaces each one(figure 2):

- 802.15.4 [12] at 868.6 MHz for information gathering coming from sensors. We use this technology for sensor/actuator wireless network.
- 802.11 at 2.4 GHz for citizen public service enabling them to connect to the smart city infrastructure at specific places in order to access directly to the services.

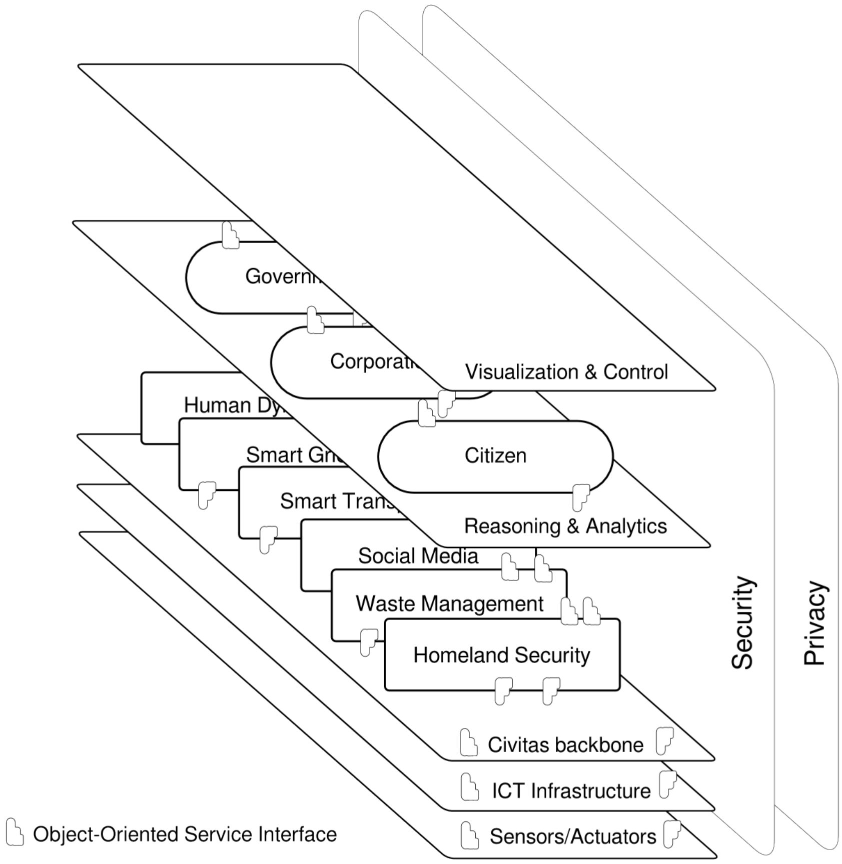

Fig. 1. Civitas framework overview

– 802.11 at 5.8 GHz for connecting the service node between them in a mesh network if there is no Ethernet connection (e.g ADSL modem) available.
– Ethernet interface for connecting the service node to the rest of the infrastructure.

The best way to understand Civitas is through the current implementation of services and layers described in the following subsections.

3.1 Sensor integration: pollution monitoring

The lowermost layer of the framework integrates different devices, in order to create a "*wireless sensor/actuator network*" (WSN). In the case of monitoring air pollution, the aim of this WSN is to gather and monitor the values of some pollution parameters like O3 or CO2. This activity is justified because there are several worrying information about pollution, and it is known that big cities generate approximately 70% of the gases that cause the greenhouse effect [2].

Fig. 2. Civitas pollution monitoring node (left) and service node (right) deployed in the school of computer science

This layer describes how to integrate different technologies into this WSN, and how to offer the pollution service as a Civitas service. One of the devices integrating this WSN is "*Waspmote Plug&Sense Smart Environment*"[1], in charge of periodically gathering air pollution values using different sensors(Figure 2). Then, the gathered data are sent to a Civitas service node[2], that receives frames with the values and some other extra information, like battery level or "waspmote" coordinates. Finally, a Civitas node device with a "Node.js" web server consults a MySQL database that contains sensor values sent by the gateways.

Gathering pollution values would not have sense if these values could not be seen. Consequently, an implementation of the visualization layer and accessible way to observe and monitor them is needed. In our case, we choose as implementation of visualization layer a Node.js web server which is responsible for serving html5 pages, containing updated polluting values, to the users (Figure 3). Citizens can therefore use laptops, smartphones or tablets to visualize them in a web browser. Node.js is a platform built on Chrome's JavaScript run-time that is highly scalable. This is very important to satisfy all user's requests. Moreover,

[1] http://www.libelium.com/development/plug-sense/
[2] `http://www.libelium.com/development/meshlium`

a library of OpenStreetMaps [13] project is used in the html5 page to show a map with the real location of the Waspmotes and their values, providing to the citizen a more integrated experience with the city environment. Additionally, we expose the collected values by mean an object-oriented interface specified in the mentioned interface definition language.

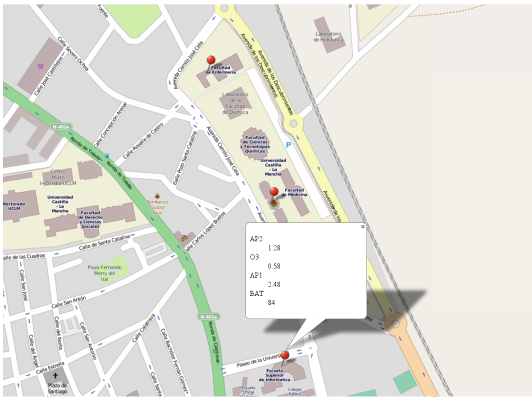

Fig. 3. Civitas visualization layer example

The proposed interface for this service should look:

```
module Pollution {
    struct Coord {
        double latitude;
        double longitude;
        double altitude;
    };
    struct Gas {
        long sensorID;
        long timestamp;
        Coord point;
        double CO;
        double CO2;

    };
    interface PollutionListener {
        void report(Gas m);
    };
};
```

3.2 Protocol integration: MQTT

From the pollution monitoring service we extracted a set of requirements for Civitas. The Civitas framework, as a middleware for the Smart Cities, needs to be able to adapt to new technologies in order to easily adopt new ways of collecting/distributing data. The Sensor/Actuator and the backbone layer of the Civitas framework addresses the integration of an event-based infrastructure which uses the deployed WSN in order to publish information about the environment. This information will be disseminated to those parties subscribed to the communication channel (e.g emergency teams, security people, etc.). The previous subsection has introduced different types of devices comprising a WSN. This section focuses in the dissemination infrastructure and more specifically, in the use of MQTT-SN and MQTT protocols. MQTT [3] is a royalty free, very lightweight, publish/subscribe, asynchronous messaging transport protocol designed by IBM company. Its design principles makes MQTT ideal for emerging paradigms of connected devices like M2M or IoT and also for mobile applications and sensor networks. MQTT is undergoing a standardization process at OASIS[3] and the protocol specification has been openly published with a royalty-free license. It is also important to take into account the more than possible increased use of MQTT due to Eclipse Paho project which is currently developing some greats MQTT clients. But WSNs usually do not have TCP/IP as transport layer. They have their own protocol stack such ZigBee on top of IEEE 802.15.4 layer. Thus, MQTT which is based on TCP/IP cannot be directly run on WSNs. For this reason MQTT-SN was developed, which is an extension of MQTT for WSNs.

MQTT-SN is designed to be as close as possible to MQTT, but it is adapted to the peculiarities of a wireless communication environment such as low bandwidth, high link failures, short message length, etc. Clients are WSN nodes, which communicate via a gateway to a broker on IP network [3].

According to the infrastructure described at the beginning of this section, Figure 4 describes the communication architecture. As it can be seen the Waspmote Plug&Sense devices (recall section 3.1) forms a WSN with Meshlium gateways using MQTT-SN over 802.15.4 layer like the protocol communication. The gateway is the responsible for communicate the WSN with a TCP/IP network. So in the Civitas Service node, a protocol gateway translation between MQTT-SN and MQTT is needed. With this process of translation a point-to-point communication with the MQTT broker can be established.

All of this involves an infrastructure which allows WSN devices to publish environment data to a server in a TCP/IP network. The only task that it needed to make is translate this information for Civitas platform format.

3.3 Service Instantiation: Action recognition

Civitas is here considered not only as a middleware framework where services and devices can be deployed, but also as an urban area consisting of infrastructures, smart platforms and networks (Civitas backbone and ICT Infrastructure). In this scenario, citizens play an essential role, increasingly connected one to each other, allowing themselves and their mobile phones to work as sensors.

[3] `https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=mqtt`

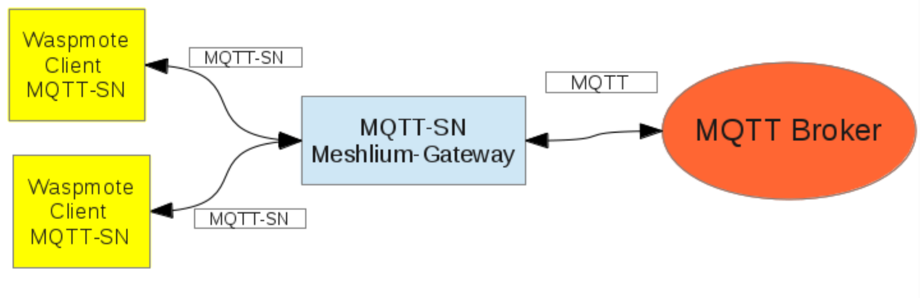

Fig. 4. Communication architecture

Making the most out of this circumstance, Civitas has devised a way of integrating mobile phones as though they were traditional sensors. These sensors provide context-aware services since they accompany their owners all day long. In order to prove the capability of such devices, Civitas provides a service that identifies the output of a given set of actions regarding the tasks being performed by the phone holders, we use this implementation as example of reasoning & analytics layer. Particular attention is paid at those tasks that involved transportation.

A machine learning approach is followed in this service, based in a learning by example technique. First of all, in order to state different means of transport, an application has been develop in order to collect values from sensors available in a smart phone. This application has been developed considering a Samsung Galaxy S4 and therefore the sensors this phone provides (gestural sensor, proximity sensor, gyroscope, accelerometer, geomagnetic sensor, temperature and humidity sensor, barometer, RGB light sensor, GPS and positioning). The application is in charge of automatically collecting these sensor values and characterize the signal through a feature extraction process. The resulting feature vectors will be used first for a training process and once the system has been trained, and a model for each means of transport has been computerized, feature vectors will be used for the recognition process. Please note that the training process is carried out just once, offline. Once the service is deployed in Civitas, sensor values are used for recognizing means of transportation. The result of the recognition process is communicated to the middleware framework using a Publish/Subscribe approach, so that other services interested in such information can subscribe to these channels and therefore use this information for more elaborated services.

Examples of services that could make use of this basic means of transport recognition are services analyzing traffic congestion in a city, public safety, or even private services. Regardless, the result of the recognition tasks should be provided in a standardized way to the Smart City management framework, so these can be automatically used by other specialized services.

This service is therefore useful not only for the Smart City but also for the Ambient Assisted Living (AAL) [4] and Ambient Intelligence (AmI) [5]. The main objective of the AAL is to provide support services to elder people, it uses technological devices that are present in the environment naturally. This is where comes the second paradigm, which promotes an environment where technology is always available and fully integrated, almost imperceptible way for its users.

If we also take into account that human attention is a limited resource, these technologies can help overcoming this limitation by providing daily assistance based on these ubiquitous technologies such as mobile phones.

The role that smart phones can play in enabling contexts conceived by AAL or AmI is critical because they often come with us in our daily lives and their capabilities are constantly increasing.

We could recognize the activity the person is performing at each moment in time and the location in which it is being carried out, so that any other system can provide more sophisticated functionalities based on these data, adjusting to the peculiarities of each user. Some examples might be:

- Make a call to the emergency services if it detects that a person takes too many hours without moving and may be seriously ill.
- Notify parents or guardians of a child if caught on his mobile phone that is involved in a fight.
- Submit a attendance unit looking for an Alzheimer's patient that has been lost and walking away from home.

4 A standardization mechanism for services and devices in the Smart City

The Smart City concept presents a challenge in the current conception of cities through the most efficient creation or renovation of infrastructures, intelligent and respectful with the environment. By means of a combination of technologies the Smart City tries to offer a better quality of life to the citizen and a reduction of the environmental impact. All these changes do not consist simply of a technical procedure, but also it is necessary to bear in mind the change produced in, for example, a company in general.

The proposed Smart City paradigm is not solely based on the employment of new technological devices, but it is also based on information that these devices can provide and how to treat such information. Therefore, it is possible to define the Smart City as a knowledge system, whose sources of information are all devices or sensors deployed throughout the city. However, it is important to highlight that the majority of the collected information from the Smart City do not follow a concrete standard, showing formats difficult to handle and originating information islands.

To be able to work with the information generated by the Smart City, it is necessary to initially identify the concepts that compose the area of the city, so that later on, it might be possible to relate and provide them with the appropriate semantics and a common vocabulary. It is therefore necessary to count on an ontology for the Smart City, so that it could carry out the standardization of the different services that the city offers, obtaining this way a deeper knowledge and a precise specification of the concepts that these include.

An ontology defines a common vocabulary to describe the concepts of a certain context, in this case the domain of the city, through a set of basic terms and relations between them. The use of an ontology in the Smart City will provide better communication, integration and sharing between the different services it could offer. Across the ontology one tries to reach the semantic interoperability of the smart cities, carrying out a homogenization of the diverse services they offer. Implementing this approach a global solution is obtained that can be apply to any city.

However, the employment of an ontology is not sufficient. Once the information offered by the city is collected, it is necessary to be later on processed in order to support intelligent decisions-making, giving the impression that it is the city itself the one reasoning, similarly as a person would do. To this end, the approach presented here proposes the use of a common-sense reasoning[1] as a way of enhancing the semantic held by the ontological concepts and providing means to infer new knowledge out the information present in the knowledge base.

4.1 Service Composition: A service based on other services

As we mention before, the Smart City paradigm heavily depends on how to obtain information about the city and how to process it, and in this endeavour, Civitas can play an essential role by providing a great number of ways of integrating and providing devices and services. Previous sections have proved how different technologies, protocols, and services could be integrated into Civitas. This section is intended to prove how citizens can use all that information held in Civitas.

The interaction between the Smart City and citizens is one of the most interesting application conceived by the Smart City paradigm. Citizens are a continuous source of information, as they give life to the city while they enjoy it. So, the data they produce is a big valuable resource that the Smart City should collect. However, at the same time, the Smart City cannot be conceived merely as an information consumer, but on the contrary it has to provide citizens with useful services. This can easily be achieved in Civitas by developing applications built upon the functionality provided by previously studied layers.

This subsection shows a Civitas application in which interaction between the citizen and the Smart City is leveraged. Potentially, thousands of applications can be deployed into the Smart City framework provided by Civitas. The one presented here is a prototype that can be used as a model of how this interaction is supported.

First of all, the application must provide a service to the user. In this case, this application proposes a geolocalization service and an interaction application. This means, using all the services deployed over the Smart City (restaurants, bus stops, parkings, etc.), this application prototype should be able to detect and geolocalize them. Note that, when the application works for one service – for example, the restaurant service–, it can be widely scalable for the rest of them. The more services the Smart City presents, the more interactions and geolocalizations the application could implement.

Regarding the technologies available for developing this application, many available ones could be employed, however, the following ones have been proposed to this end:

- Bluetooth technology is chosen for service discovery and communication support between them and the smartphone. Bluetooth is a very strong and stable specification that allows data transmission between many different devices. Moreover, it can be easily implemented in a smartphone.
- For the geolocalization task, the already mentioned OpenStreetMap project comes into scene. It is an open-source project which provides free geographic data for all over the world.

The use of these two technologies support geolocalization and communication between Smart City services. This information can be exchanged with citizens which while providing citizens with these two services can collect user data. Using this approach, both communication sides obtain useful data from the interaction. The user information obtained and stored by the Smart City can be sold to external companies interested in which are citizens' customs, habits or preferences. Moreover, these data increase the knowledge held by the Smart City.

5 Conclusions

This paper presents a description of how to implement different granularity services for Civitas, a Smart City framework. To this end, different technologies and communication protocols have been employed, therefore demonstrating how new services and devices can be easily integrated into the framework.

Starting from an existent middleware we are more close to a final solution than other approaches. We can also focus in the research of new middleware services as reasoning layers or protocol integration. This ongoing work has to be validated in real testbeds and provide with examples of integration of final services and devices.

Acknowledgments. This research was supported by the Spanish Ministry of Science and Innovation through project DREAMS (TEC2011-28666-C04-03).

References

1. Fahlman S.E.: Marker-Passing Inference in the Scone Knowledge-Base System, First International Conference on Knowledge Science, Engineering and Management (KSEM'06), Guilin, China,Springer-Verlag. 2006.

2. Trenberth K.E., et al. Observations: surface and atmospheric climate change. Climate Change 2007: the Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge, Cambridge University
3. Stanford-Clark et al., MQTT for Sensor Networks (MQTTs), Oct. 17, 2007 Press, 2007; pp. 235-6
4. Pieper, M., Antona, M., Cortes, U.: Editorial: Ambient Assisted Living. ERCIM News, Special Theme: Ambient Assisted Living, 87, 18-19, 2011.
5. Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J.C.: Scenarios for ambient intelligence in 2010. Brussels Belgium: ISTAG. 2010
6. Michi Henning and Mark Spruiell, *Distributed Programming with Ice* Revision 3.4, June 2010, ZeroC Inc.
7. Theodoridis, E.; Mylonas, G.; Chatzigiannakis, I., "Developing an IoT Smart City framework," Information, Intelligence, Systems and Applications (IISA), 2013 Fourth International Conference on, July 2013
8. Khekare, G.S.; Sakhare, A.V., "A smart city framework for intelligent traffic system using VANET," Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on, pp.302-305, March 2013.
9. Vakali, A.; Angelis, L.; Giatsoglou, M., "Sensors talk and humans sense Towards a reciprocal collective awareness smart city framework," Communications Workshops (ICC), 2013 IEEE International Conference on, 189,193, 9-13 June 2013.
10. Cosido, O.; Loucera, C.; Iglesias, A., "Automatic calculation of bicycle routes by combining meta-heuristics and GIS techniques within the framework of smart cities," New Concepts in Smart Cities: Fostering Public and Private Alliances (SmartMILE), 2013
11. Szabo, R. et al., "Framework for smart city applications based on participatory sensing," Cognitive Infocommunications (CogInfoCom), 2013 IEEE 4th International Conference on, pp.295,300, 2-5 Dec. 2013
12. IEEE 802.15.4-2006 standard, http://www.ieee802.org/15/pub/TG4.html.
13. *OpenStreetMap Wiki* [online]. 2010 [cit.2014-03-15].http://wiki.openstreetmap.org/wiki/Main_Page