An Approch for Monitoring and Smart Planning of Urban Solid Waste Management Using Smart-M3 Platform

Vincenzo Catania, Daniela Ventura
Department of Electrical, Electronics and Computer Engineering
Catania, Italy
first.last@dieei.unict.it

Abstract—**Solid waste management is one of the most important challenges in urban areas throughout the world and it is becoming a critical issue in developing countries where a rapid increase in population has been observed. Waste collection is a complex process that requires the use of large amount of money and an elaborate management of logistics. In this paper an approch to smart waste collection is proposed able to improve and optimize the handling of solid urban waste. Context of smart waste management requires interconnection among heterogeneous devices and data sharing involving a large amount of people. Smart-M3 platform solves these problems offering a high degree of decoupling and scalability. Waste collection is made by real-time monitoring the level of bin's fullness through sensors placed inside the containers. This method enables to exempt from collecting semi-empty bins. Furthermore, incoming data can be provided to decisional algorithms in order to determine the optimal number of waste vehicles or bins to distribute in the territory. The presented solution gives important advantages for both service providers and consumers. The formers could obtain a sensible cost reduction. On the other hand, users may benefit from a higher level of service quality. In order to make users feel closer to their community, they can interact with the system to be aware about the fulness state of the nearest bins. Finally, a mechanism for collecting "green points" was introduced for encouraging citizens to recycle.**

I. Introduction

Internet of Things will extend and revolutionise the concept of "cities" making them more comfortable and able to give intelligent responses to different kinds of events or needs, and also common "things" and infrastructures will be integrated with new hardware and software technologies. Interactive sensing streets or active buildings will gather environment data about the city and measure the level of noise, traffic, crowds, temperature - literally everything. Data will be transmitted to other "smart objects" or "smart controll systems" and processed by them. This information will be aggregated to achieve and provide more complete and intelligent services to citizens, monitor the environment and act quickly in case of natural disasters, improve decision-making both in public and private sectors.

Research on smart cities has been conducted by many organizations and many applications have already been implemented. Typical examples include: smart parking which supplies the position of a car park at any time [1]; smart agriculture to improve agroindustrial production based on weather and environmental conditions [2]; smart transport to find the best route taking into account the current traffic conditions [3]. All these applications are a step towards the realisation of a complete smart city.

Another field of interest that should be made smart concerns the waste collection. All cities, regardless their size, their geographical location or their economic level, spend huge amount of money every year for waste collection. The number of bins located in the streets and the number of vehicles used to empty them are generally estimated based on the number of citizens, but the resulting estimation is sometimes either too high or too low. The natural consequence is the provision of poor service or to incur in high costs (e.g. the cost of fuel for too many trucks). Furthermore, the collection of waste, regardless the type of material (recycling or unsorted), is typically fixed weekly without taking into account the actual state of the level of fullness for each bin. The result is the collection of semi-empty bins or the trash accumulation degrading conditions of hygiene of the city.

Predicting best time to make the garbage collection and optimizing the number of vehicles and containers placed on the streets became feasible operations only if a stream of constant information on the quantity of waste daily is provided. Less recent studies on waste management have considered the load-account analysis [4] considering the amount of waste collected and disposed in the landfills. Conversely, the diffusion of low cost mobile devices having longer battery life enables to data collection on the amount of waste produced directly on-site and in real-time. Monitoring the fullness of bins through the use of various types of sensors, it is possible to achieve a more efficient system. If the system is also able to respond appropriately to events that occur in real-time by applying different strategies depending on the event itself, the system can be defined "smart waste system management".

Moreover, in a smart city context, it is also important to allow users to interact with ubiquitous information produced by the city, anytime and from any device [5]. New user-centric applications will be developed and they will bring new forms of interaction, utilizing sensor data production and social networking. The role of user should be promoted in the field of the smart waste management as well.

In this paper we propose a solution for smart monitoring and planning of waste management. We adopted the Smart-M3 platform [6] that enables us to take advantage of some of its characteristics such as interoperability, uncoupled commu-

ISSN 2305-7254

nications, wide range of APIs and ease of implementation. This solution consists manly in two phases: a monitoring phase in which the fullness levels of the rubbish bins is constantly measured; a computation phase in which collected data is elaborated for the optimization of the trash collection. Beside, we have designed an application for users that provides information about the nearest bins and handles the collection of green-points with the aim to stimulate users in recycling.

The remainder of this paper is organised as follows: Section II discusses the domain of interest for this paper and gives some food for thought. Section III describes the case study dealt with and the components used. In Section IV, we summarize the basic concepts of Smart-M3 platform and the characteristics of which we have most benefited. In Section V, we discuss the architecture which we have defined and the ontology that we have used is examined in depth in Section VI. Finally, in Section VII, we talk about the implementation and some results obtained simulating the system.

II. Domain of interest

The planning of a smart waste system is a complex task and it does not have an univocal solution, but it depends on many factors and aspects which have to be considered and evaluated. First of all it is important to narrow the domain of interest and understand the goals to achieve. The following questions can help to design and implement a smart waste system. We tried to answer to these questions to better explain our domain of interest that we have been taking into account in this paper.

A. What category of waste is a subject matter of the study?

Waste is surplus of various human activities and it can be categorized in base of origin and degree of danger. The three main categories are:

- solid waste: they are a highly heterogeneous class and are non-hazardous waste produced by public or private (household waste). This category covers both the unsorted and the recycling and what differentiates these two types of collection is just the process of garbage disposal. So, the design of a smart waste system that deals with both types of waste is equal until to the transfer of waste in landfills;
- medical waste: this category includes waste from hospitals, pharmacies, veterinarians or other health care facilities. This type of waste may be liquid or solid as also infectious. To design a system that deals with the collection of medical waste, it is necessary to consider that the collection usually occurs not through public bins but directly from the manufacturers or sellers of medical equipment;
- hazardous waste: they include toxic, explosives and waste that can be harmful to humans, generally are produced by factories. Designing a smart system for these types of waste is very complex because the transport requires the use of special resources able to isolate them and to avoid that they could contaminate environments, lakes, rivers or aquifers and the collection process is highly dependent on the type of material to be collected.

In this paper, as said so far, we have dealt with only the management of solid waste. It is more simple to apply and to realize in a smart city context. The other two types of waste management require special licenses or vehicles different from one country to another and so it's difficult to achieve a general smart system for them.

B. What type of collection is used in the area where the smart waste management should be realized?

Once confined the problem to be addressed only to a specific category, for us the solid waste category, the next step is figuring out what type of collection is carried out in the place where the system should be achieved. In fact, various collection and container systems are used depending on the areas where reference is made to. So, it is possible to distinguish two different modalities:

- door-to-door collection: it is a technique which provides a periodic pick up service;
- indirect collection: it's based on the use of containers or communal bins placed near markets, in apartment complexes, and in other appropriate locations where special vehicles go to gather waste.

In both cases, the goal is to plan a system that would to make smart the collection in relation to the times and days of waste collection according to the different types of materials. In this paper, we focus primarily on indirect collection although we think that the proposed system can be easily adapted also to the door-to-door collection.

C. What goals does it want to obtain?

There are several aims that a smart system for urban waste management may want to achieve. We list the main:

- optimize the routing operations;
- sometimes the containers are moved, stolen or damaged. For companies responsible for the service is very important to be able to trace the positioning of the different waste bins within the urban or metropolitan areas to reduce theft or monitoring of position's change;
- reduce unnecessary costs of companies like the fuel of vehicle;
- provide a higher quality service to the citizen;
- identify and trace the profile of workers who work in the collection;
- increase recycling and so to reduce the environment impact of the waste dumping.

Being able to achieve all these objectives simultaneously is very difficult because some goals could be in conflict each other. In our system, we have focused on optimization of the trash collection, costs, quality of service and increase recycling.

III. Hardware Architecture

To achieve a smart waste system, we have used very different components each capable of interacting with the smart-space, which will be deepened in the next sections. Figure 1 shows graphically the hardware architecture used.

In accordance with the vision of the smart cities where it is required the insertion of components capable to measure and transmit data about the context in real time, inside the bins we have inserted two types of sensors: a proximity sensor located inside the lid or on the the internal and upper part of the bin; and a weight sensor in the bottom of the container. The proximity sensor has the task to measure the level of fullness of the container; the weight sensor is used instead to measure the amount of trash contained in the bin or thrown by a user. Each bin has a Zigbee module that is able to communicate the values of measured physical quantitaties to the nearest light pole. Alternative protocols, instead of Zigbee, could be used (e.g. BLE) because the system logic and the protocol to communicate sensors data are decoupled thanks to the use of a gateway component in the light pole.

The gateway was realized through a Raspberry PI and its most important task is to collect, process and then transmit data measured by the sensors to the control center. Each Raspberry PI is powered directly by the power source used to produce the light of the lamppost, has a Zigbee module to get the values from the sensors, and a network interface, i.e. one gprs or wifi shield. Furthermore it is also important to determine the position of the bins. Since the light pole is close to a very small group of bins, it is permissible to think that it is directly the Raspberry PI that associates to each bin the GPS coordinates recorded.

The control center is the component that uses sensor data to implement effective and efficient optimization strategies and to find solutions for problems linked resources organization to solid waste management. Also it informs any vehicle whether and when they have to empty bins.

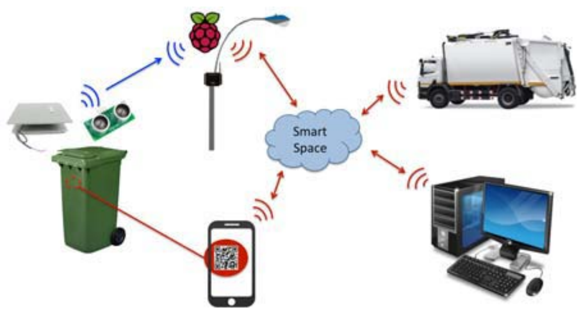

Fig. 1. Hardware components used in the smart waste system

In our system, a city is divided into areas. A list of coordinates' points is a polygon, i.e. an area. Each waste vehicle is mapped on an area of the city during the configuration of the system. The waste vehicles have an onboard computer or a tablet with the Internet connection. They receive in daily time slots the list of bins that have to be emptied, according to their area and to quantity of full bins.

Finally, we have considered each user can interact with the smart waste system through his smartphones or tablets. These devices are used for the authentication and login into the system, to collect "green points" and to know the state of the nearest bins and in base of waste types.

IV. Smart-M3 platform for Smart Waste Management

Smart-M3 platform is an open-source project that provides an environment in which different entities can share information and cooperate in a transparent way to the heterogeneities. The space in which the agents interact, called "smart-space", is a virtualization of the real environment where relevant informations about real world are stored and kept up-to-date at every moment. It is based on the use of ontologies to describe relationships between real entities and to contain shared data. The exchange of messages happens modifing, adding or removing subjects, predicates or complements from ontology itself.

The components of a smart space are: semantic information broker (SIB) which is the entry point to the RDF graph, and knowledge processors (KPs), which run on phisical devices. KPs comunicate to the SIB their data using the SSAP ("smart space access protocol") protocol which defines a set of messages for reading or modifying the smart-space. More details on the architecture of Smart-M3 platform are presented in [7], [8].

Many applications in different domains were implemented using Smart-M3 platform. Some of these use cases are: a cross domain application for wellness domain and home entertainment domain [9] that shows as independent agents can cooperate each other to provide a high user experience; another application is in the domain of smart-buildings where interoperability between furnitures and smart objects made by different manufacturers is rare [10]; and SmartRoom is a system that provides a set of digital services for activities localized in a room and during meetings [11].

The domain of waste management is a context characterized by a highly dynamic and fast data production, as well as by the presence of many users that require this data. The smart waste management proposed uses Smart-M3 platform to solve these problems. In fact, first of all, Smart-M3 platform gives a high level of decoupling between producer and consumer of data. The participants in the smart-space are unaware of other participants, their physical characteristics or capabilities. Therefore, the smart-space acts as an intermediary between different agents and provides a unique means of communication understood by all components. So, a producer can send data to a consumer without the latter having to directly ask the producer.

A second feature concerns the sharing of knowledge and access to information on the monitored environment. First of all, data sharing in the form of RDF triples and data access through semantic queries, make communication independent of the operating system or manufacturer. Also, Smart-m3 allows to have direct access to the freshest and updated data among multiple devices. An agent can make a subscription on an information so as it can be notified when data value changes without making an explicit request to obtain the newest data. Each agent can customize the type of subscription in order to obtain personalized services.

Moreover, Smart-M3 platform gives a good degree of ease to extend and integrate different applications in similar contexts, through integration of ontologies. This feature is important in smart cities context in which many innovative applications could be implemented. For example, our waste system gives data about position of waste vehicles on the streets. These data could be aggregate with other information to realized a smart traffic monitoring system through easily integration of two or more differents ontologies.

V. Software Architecture

According to the Smart-M3 view, devices that want to take part to the smart-space have to implement one or more of KPs. The smart waste system, showed in Figure 2, is composed by many types of KPs: two KPs for each light pole, a lot of KPs for the control center, two KPs for each vehicle and two for the users' mobile device. Each of them cooperate and share data through the smart space, when certain events occur.

In the system that we have developed, some constraints have been applied:

- waste collection frequencies are daily and it happens in fixed time slots. In particular to simplify our tests, we have supposed that each day there are two time slots: at 5 a.m. and at 2 p.m.. Obviously through a configuration operation it's possibile to chage these times;
- the types of materials collected are: glass, plastic, paper and general waste;
- residual waste and recyclables are collected separately;
- separate fleets of vehicles are employed for different types of materials.

To better describe the proposed architecture, we divide the description into two sub-sections.

A. Real-time monitoring and intelligent planning of daily collection operations

Each proximity and weight sensor, placed in each bin, transmits the measured values to the Raspberry PI with its own rate using the Zigbee protocol. But for the optimization of the logistics, we have considered only the values of proximity sensors, so as to identify which containers are close to their fill level, while the values of the weight sensors are used in the next subsection to interact with users. Each Raspberry PI, placed on a light pole, has two KPs: SensorsLightPole-KP and CoordsLightPole-KP. Each time that a SensorsLightPole-KP or a CoordsLightPole-KP perform an update-query on the smart-space, the control center is notified, but they have differents goals. The SensorsLightPole-KP updates the sensor data within the smart space. It may decide to adopt optimized strategies to perform this publishing operation. For example, if the value of a certain sensor does not change compared to the previous measurement, it makes no sense to act on the smart-space; or it might want to aggregate multiple data for the same sensor making an average first to communicate with the SIB. The CoordsLightPole-KP is responsible for updating the coordinates of the bins. If the bins are moved, a KP of another light pole will send a notification to the KP of the control center so to update its personal list.

The control center uses the GarbageLevelManager-KP to collect data from the various proximity sensors and evaluate the level of filling of a bin. We have considered four different levels: empty, half-empty, half-full and full. All information related to the level of bins' fullness are saved by GarbageLevelManager-KP in a local database with aims to provide useful information for offline processing of the data collected. The history of real data and correspondent timestamp can be useful to a decision support system that can find solutions to the problems of organization of resources related to the management of solid waste. For example analyzing the time taken to fill the bins in one of the areas of a city could help understanding the best number of bins that should be distributed in that area: if most of the bins are filled in a short time, the analysis of data suggests the need to add new bins in that area in order to provide a more efficient service to citizens, otherwise if the bins are filled in a quite long time, it means that it could reduce the amount of bins without affecting the service.

After having known the level of garbage of a bin, one of the following two cases can happen:

- if the level of a bin belongs to the full or half-full band, the GarbageLevelManager-KP upgrades the smart space by creating a connection between the bin and the vehicle which should gather the area of the city where the bin is located. Creating this connection does not mean that the vehicle will surely recover the bin. This phase of the decision will be implemented by another component of the control center, that is the VehicleStatusManager-KP.
- if the level of a bin is in the empty or half-empty band, the GarbageLevelManager-KP removes the connections between the bin and the associated vehicle to the area where the bin is located.

Whenever the time band in which the vehicles should carry out the collection springs, the VehicleStatusManager-KP applies an algorithm to determine if an area must, may, or doesn't require the collection. The algorithm applied by us compares the total number of bins in an area with full or half-full ones and marks all areas in clean, little dirty or completely dirty. More sophisticated algorithms can be integrated into the system without changing in the structure proposed in this paper. VehicleStatusManager-KP acts as publish to update the state of the vehicles of the smart-space. So it assigns the status in this way:

- All little dirty or completely dirty areas will make the status of the vehicle to "work".
- All clean areas will make the status of the vehicle to "not work".

The Vehicle-KP is subscribed to the status and according to this information will carry out the collection or remain in the garage. The result is a saving in terms of fuel costs for vehicles. Finally, each vehicle, through the VehicleCoords-KP, updates data about the position of the truck in real-time. At the moment this information is not used by our system, but we think it

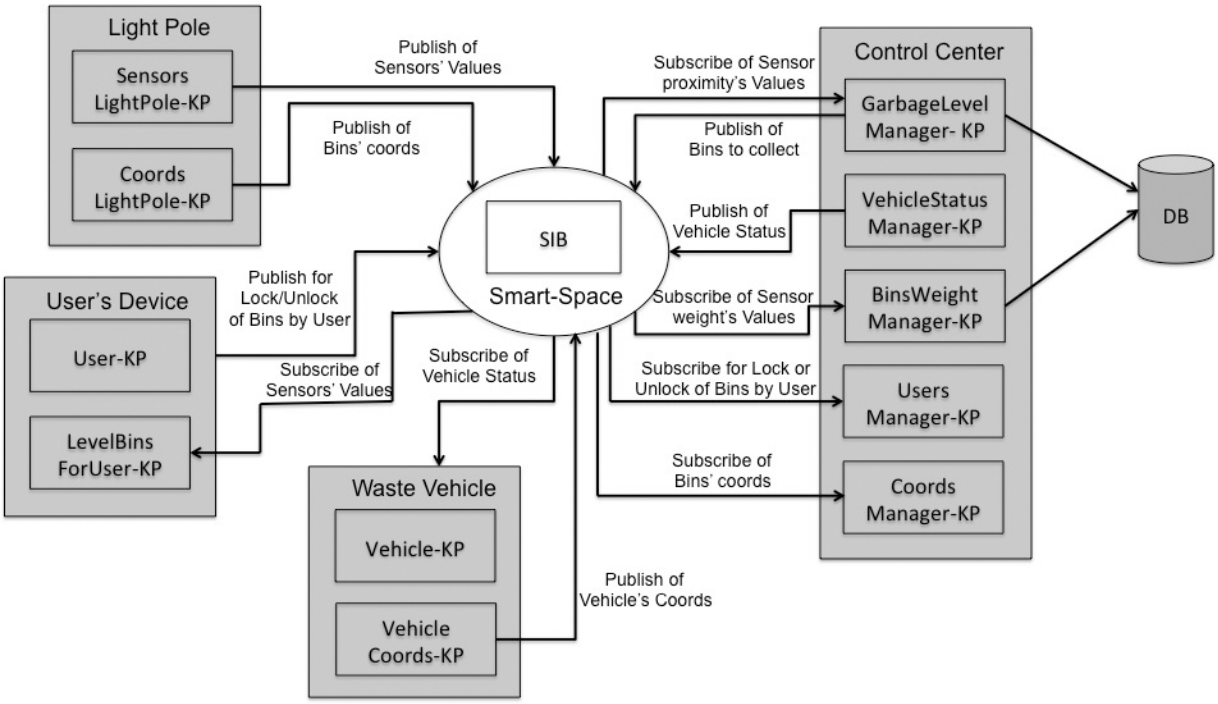

Fig. 2. Architecture of smart-space with all KPs and SIB

might be useful both to users, because if it is integrated with others applications, it gives usefull information to monitoring traffic on the streets, and to the control center, to track the movement of workers.

B. Real-time Monitoring and Incentives for Citizens

In accordance with the vision of smart user-centric cities, a user has to be able to know the measured values by the sensors in own city in accordance with his needs and interests. The first operation that a user can perform through his device, is to know the status of the closest bins to him. The LevelBinsForUser-KP has the task of carrying out this functions. It executes a subscription for the values of proximity sensors according to the type of collection and level of fullness selected as filter. It can be turned on and off by the user through the application. The time interval to receive notifications is fixed by the user.

The second operation that the user can do through his device, is log in to the system and to collect green-points. The idea of providing incentives to users who help to carry out a task or just to encourage them to use a service, it is still a very popular thing and widely used. The most obvious example is the collection of points provided by a supermarket chains. Who hasn't got at least one card to collect points in their own wallet? In this case, the user is rewarded with physical goods (e.g. household items), more quantity of money spent in the supermarket. Even on the web field, the use of incentives is a very common practice. In [12], it is investigated the effect of incentives in web-based surveys showing that monetary incentives increase the number of people who fill out a survey.

The Internet of Things will enable new type of incentives for new type of goals. In a smart city, we believe it is possible to implement a mechanism based on providing incentives to users in order to increase the amount of garbage to recycle. In fact, in the architecture described so far, it has been inserted the chance for citizens to obtain "green points" on the base of their behavior. At the moment the behavior of the user is measured based on the amount and the type of waste thrown. In the future, if it be possible to have a system to evaluate the material thrown by the user, it will be possible to evaluate also the qualitative aspect of the collection made for each user and possibly "punish" him if he commits frequent mistakes.

As previously mentioned, each bin has a sensor to measure the weight of waste in real-time and a QR-code, applied on different edges, which encodes the ID of the bin. For each new sensor values updated in the smart-space, the BinsWeightManager-KP saves the data in the database. When a user, through website of the company that manages the collection, does the authentication (e.g. OpenID), he obtains an access token. Through the webcam of his smartphone or tablet, he frames the QR-Code and, at the time when the QR code is decoded, the User-KP makes an insert-query in the smart-space adding the user's identifier connected with the bin's indentifier in the ontology. The UsersManager-KP, active in the control center, is notified and it sent the ID of bid and the ID of the user to the BinsWeightManager-KP. From this moment onwards the BinsWeightManager-KP measure the amount of trash thrown by the user and convert them in "green points" up to the moment in which the user will do the logout. Obviously these points are saved on the database of control center. The green points could be used to provide users benefits or fees discounts or simply by comparing each others achievements with friends so create a social real game. The study of finding the best financial incentives or other type's incentives, is an important area but it is altogether another area of research, not treated in this paper.

Instead of the QR-Code, it could also use authentication mechanisms through NFC tags. In this case, however, the proposed smart waste system couldn't change. The choice of using the QR-Code is needed by the fact that it is simple to use, all smartphones or tablet can decode it (while the NFC is present only on the latest generation devices) and is also very suitable for use with the so-called "wearable technology" (such as the "Google Glass").

As further improvements for user authentication management, we want to evaluate the possibility of implementing the mechanism of policies based on the exchange of public keys described in [13]. Moreover, to manage the atomic lock on a single RDF triple, we are considering to extend the architecture through the mechanism described in [14].

VI. Ontology

The Smart-M3 platform requires the smart-space is represented as a RDF ontology on which KPs acting through update, insert, or delete queries. The classes in the ontology are shown with the shape of rectangle in the Figure 3, while the circles are the data properties.

The *Sensor* class consists of two child classes that distinguish the two types of sensors in the system: *Proximity* and *Weight*. The choice of logically distinguish the sensors in smart-space is due to the fact that the two sensors are used for different tasks. The parent class has two data properties that allow to save the latest measured value and the corresponding timestamps. Since each bin contains a proximity and a weight sensor, it has been created the *Contains* object property, as it is possible to see in the Table I.

TABLE I. List of Object Properties used in the smart-space

Domain	Property	Range
Bin	Contains	Sensor
Bin	IsLockedBy	User
Vehicle	Collects	Bin
Vehicle	AssignedTo	Area

The class *Bin* has the *type* attribute to indicate the type of dumpster between glass, plastic, paper or general waste. In addition for the management of bin's localization, the Raspberry PI updates the *coords* attribute. The management of the incentives given by the user takes place after the phase of user login. During this operation, the system creates an entry of type *User* with the access token of the user and adds the object property *IsLockedBy* between this individual and the corresponding bin in which the user is throwing waste. The *status* of the bin is used by the control center to know when it has to start the collection of *green points* of the user. This attribute can assume the values "lock" when the user is logged and "unlock" after user's logout. During the logout operation, the instance of type *User* containing the access token is removed from the smart-space. It should also note that the *User* class has no properties because for security reasons we have avoided to enter passwords or other information characterizing the user directly in the ontology.

We have decided to divide a city, object of interest, in some areas, each of which is identified by a list of coordinates. For this reason, the *Area* class contains the list of points that form a polygon. The control center checks in which area the various bins are placed by comparing their coordinates with the coordinates of the area and updates its own local list. The control center for each "full" or "half-full" bin, acts on the smart-space by creating an object property *Collects* between this bin and its area. Obviously, when the real dumpster is emptied, the proximity sensor will report that the bin is now empty, and therefore the property *Collects* will be removed.

Finally, the class *Vehicle* has the *type* property to indicate what type of collection the vehicle is able to perform (depending on the material), the object property *AssignedTo* to indicate the collecting area in which the vehicle has been assigned, *coords* to indicate the current position of the truck and a *status* that indicates whether the vehicle is going to make a collection or not. In the case in which the vehicle is in the "work" status, it will know the list of bins to empty via the object property *Collects* made with the various bins.

VII. Results and Applications

All components of the system were implemented using Python language and the ontology was developed using Protege.

The implemented system was verified by testing several use cases. First, communication between bins and the smart-space was simulated using a weight and a proximity sensor and a Raspberry PI. In this first test, each sensor transmit its measured values to the Raspberry through a Xbee module at regular time intervals. Afterwards, the Raspberry behaves according the following rule: whenever the value measured by a sensor exceeds a certain threshold, the measure is updated on the smart-space along with a timestamp indicating the time when the measure itself was taken.

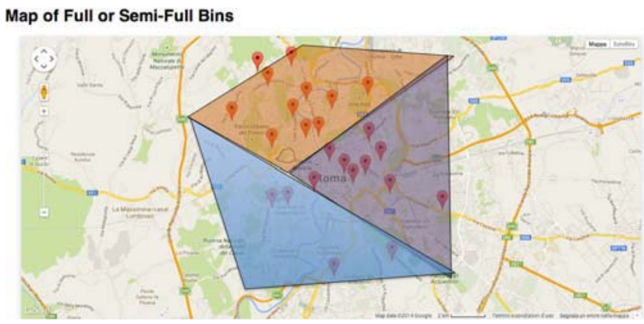

Fig. 4. Map resulting of a simulation with three vehicles

Furthermore, we have conducted some simulations to verify the behavior of the system with a greater amount of nodes. The Raspberry PI generated a random number of bins (and hence proximity and weight sensors) assigning coordinates also them random. The city taken into observation was divided manually into 3 different areas, visible in Figure 4.

For the control center and the vehicles we have used a PC. During the simulation, the values of proximity sensors have been evaluated by the control center and only those with a label full or half-full was sent to the vehicles. The system has properly set the status of each vehicle and recognized the dirtier areas (orange and purple in the image) respect to cleaner ones (in blue in the figure).

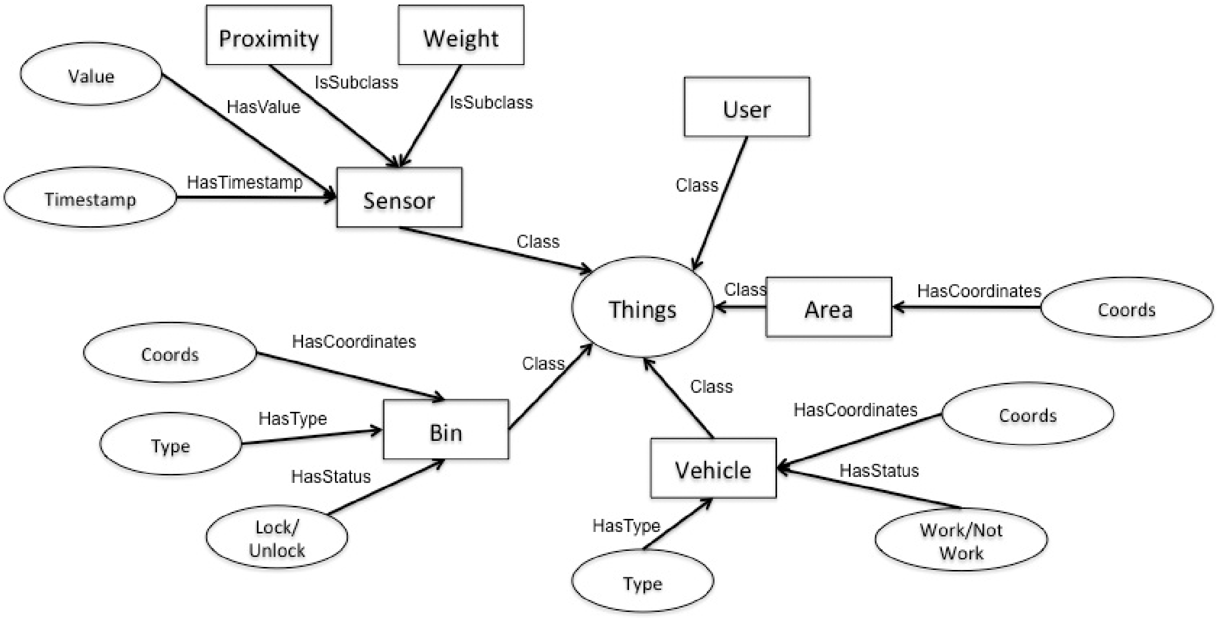

Fig. 3. Graphical representation of the Ontology used in the smart-space. The picture shows only the Data Properties

Two Android applications for users and vehicles are in phase of implementation. We are using Android SDK to realize the graphical interface of these applications. The logic is written in ANSI C. GUI elements are connected with the application logic using Java Native Interface (JNI), as explained in [15]. Moreover SmartSlog is used to develop data structures and variables related with the ontology entities.

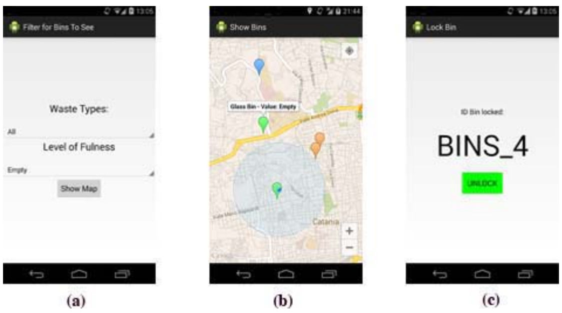

Fig. 5. Android Application for Users

In Figure 5, it shows the application for users. It consists of two main sections: the first allows to view on the map the closest bins respect to user's location based on the base of the filters inserted; the second the other allows to frame a qr-code and if it is recognized, to lock the bin (c). In order to provide the location of user we've integrated Google Map API. The different types of bins are identified by different colors on the map (b). A user can set filters (a) for a better search on the base of the type of bin (plastic, glass, paper, general) and on the level of fullness (empty, semi-empty, semi-full, full).

VIII. Conclusion

In this paper we have examined the issue of solid urban waste management. We have proposed an approch based on Smart-M3 platform through which it was possible to manage the sharing of data between devices very heterogeneous and achieve a high degree of scalability and flexibilty, important features for the management of highly dynamic and heterogeneous environments typical of smart city context. The smart system described focuses on two aspects: first of all, it is addressed to governments and private companies in order to plan a better management of resources to be deployed in city's areas and an optimal planning of waste collection; secondly, it is aimed at giving citizens the opportunity to know the position and conditions of the nearest bins and encourage them to recycling.

There are several future works and improvements for the proposed system: change the system of users authentication and atomic lock of bins during the collection of green-point in accordance with Smart-M3's features; implement graphical interfaces for the control center and complete Android applications; possibility of extending the system adding other use cases and applications for smart cities. Moreover, the proposed solution is flexible and decoupled respect to the algorithm to determine optimal number of bins and vehicles or to the algorithm to define the best route for vehicles. Therefore, future works can be made in the study of models that offer the best results in terms of decision-making.

References

[1] S. V. Srikanth, P. Pramod, K. Dileep, S. Tapas, M. Patil, and C. Sarat, "Design and implementation of a prototype smart parking (spark) system using wireless sensor networks," in *Advanced Information Networking and Applications Workshops, 2009. WAINA '09. International Conference on*, May 2009, pp. 401–406.

[2] T. Qiu, H. Xiao, and P. Zhou, "Framework and case studies of intelligence monitoring platform in facility agriculture ecosystem," in *Agro-Geoinformatics (Agro-Geoinformatics), 2013 Second International Conference on*, Aug 2013, pp. 522–525.

[3] W. Q. Wang, X. Zhang, J. Zhang, and H. Lim, "Smart traffic cloud: An infrastructure for traffic applications," in *Parallel and Distributed Systems (ICPADS), 2012 IEEE 18th International Conference on*, Dec 2012, pp. 822–827.

[4] Y. Huang, B. Baetz, G. Huang, and L. Liu, "Violation analysis for solid waste management systems: an interval fuzzy programming approach," *Journal of Environmental Management*, vol. 65, no. 4, pp. 431 – 446, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0301479702905669

[5] A. Vakali, L. Angelis, and M. Giatsoglou, "Sensors talk and humans sense towards a reciprocal collective awareness smart city framework," in *Communications Workshops (ICC), 2013 IEEE International Conference on*, June 2013, pp. 189–193.

[6] "Smart-m3: Free development software downloads at sourceforge.net," November 2013. [Online]. Available: http://sourceforge.net/projects/smart-m3/files/

[7] J. Honkola, H. Laine, R. Brown, and O. Tyrkko, "Smart-m3 information sharing platform," in *Computers and Communications (ISCC), 2010 IEEE Symposium on*, June 2010, pp. 1041–1046.

[8] Korzun, Lomov, Vanag, Balandin, and Honkola, "Multilingual ontology library generator for smart-m3 information sharing platform," *International Journal On Advances in Intelligent Systems*, vol. 4, no. 4, pp. 68–81, 2011.

[9] J. Honkola, H. Laine, R. Brown, and I. Oliver, "Cross-domain interoperability: A case study." ser. Lecture Notes in Computer Science, vol. 5764. Springer, 2009, pp. 22–31.

[10] K. Framling, A. Kaustell, I. Oliver, J. Honkola, and J. Nyman, "Sharing building information with smart-m3," *International Journal on Advances in Intelligent Systems*, vol. 3, no. 4, p. 347357, 2010.

[11] D. Korzun, I. Galov, and S. Balandin, "Development of smart room services on top of smart-m3," in *Open Innovations Association (FRUCT), 2013 14th Conference of*, Nov 2013, pp. 37–44.

[12] J. Su, P. Shao, and J. Fang, "Effect of incentives on web-based surveys," *Tsinghua Science and Technology*, vol. 13, no. 3, pp. 344–347, June 2008.

[13] Boccacci, Bordoni, Cicora, Fabbri, and Prati, "Smart m3 a secure chat application," 2013.

[14] DElia, Honkola, Manzaroli, and Cinotti, "Access control at triple level: Specification and enforcement of a simple rdf model to support concurrent applications in smart environments," in *Smart Spaces and Next Generation Wired/Wireless Networking*, ser. Lecture Notes in Computer Science, S. Balandin, Y. Koucheryavy, and H. Hu, Eds. Springer Berlin Heidelberg, 2011, vol. 6869, pp. 63–74.

[15] Kovyrshin and Korzun, "Programming android client for smart-m3 applications: Smartroom case study," 2013.