Swarm-Based Smart City Platform: A Traffic Application

Pablo Chamoso[a] and Fernando de la Prieta[b]

[a] Daegu University, Republic of Korea
[b] Sunchon National University, Republic of Korea
chamoso@gmail.com

KEYWORD

Agents, Swarm Intel-li-gence; Locating System;, Smart Cities

ABSTRACT

Smart cities are proposed as a medium-term option for all cities. This article aims to propose an architecture that allows cities to provide solutions to in-terconnect all their elements. The study case focuses in locating and opti-mized regulation of traffic in cities. However, thanks to the pro-posed struc-ture and the applied algorithms, the architecture is scalable in size of the sensor net-work, in functionality or even in the use of resources. A simulation environment that is able to show the operation of the architecture in the same way that a real city would, is presented.

1. Introduction

The social, economic, environmental, and engineering challenges of this transformation will shape the 21st century. The lives of the people living in those cities can be improved – and the impact of this growth on the environment reduced – by the use of “smart” technologies that can improve the efficiency and effectiveness of urban systems. A smart city can be defined as the integration of technology into a strategic approach to achieve sustainability, citizen well being, and economic development [1]. The smart city offers a coherent vision for bringing together innovative solutions that address the issues facing the modern city, but there are still many challenges to be dealt with.

Through the use of sensor networks, the building automation and control systems, which started with wired technology, have now entered the era of wireless technology, having produced technologies such as ZigBee, Z-Wave, EnOcean, among others. Wireless Sensor Networks (WSN) are used for gathering the information needed by intelligent environments, whether in urban construction and smart cities, home and building automation, industrial applications or smart hospitals [2]. However, the growing heterogeneity of this type of wireless network protocol makes them difficult to use. WSNs make it possible to build a wide range of applications, such as those used to control energy costs, monitor environmental data, implement security and access control in buildings, or automate industrial and home environments, and, Indoor Locating System (ILS).

Although outdoor locating is well covered by systems such as the current GPS or the future Galileo, ILS still needs further development, especially with respect to accuracy and the use of low-cost and efficient infrastructures. Some of the applications of ILSs include among others tracking (people, assets and animals), access control, wander prevention, warning and alert systems, controlling security perimeters and optimization of resources.

Thus, at present there is a growing need for versatile open platforms capable of integrating different sensing technology over wireless technologies, merging data sets from heterogeneous sources, and intelligently managing the generated information. Although there are already some architectures or frameworks that allow the interconnection of sensors, both in academia and in business [3], the reality is that existing platforms are designed for a specific end, using a specific technology stack for each deployment, and offering a very specific set of services whose functionality is very limited. Therefore, current systems are limited by the pre-installation of infrastructure, and integrators have to face the decision of choosing between other technologies or adapting their existing systems and infrastructure. It is also difficult for integrators to combine the information obtained from heterogeneous wireless sensor networks since there are no adequate tools to do so.

2. Agents' Theory and Swarm Intelligence

2.1. Agents' Theory

There are many complexities in software agents. In this paper we focus on two completely opposite ends that are described below and take shape in the proposed case study. These ends are lightweight and heavyweight agents.

Lightweight agents have their origins in the study of animal behavior [4] where, compared to the case of ants, the intelligence of a single ant is far from the great skills that the colony shows. Therefore, the interactions between agents are the place where the intelligent part is centralized.

The study of behavior has to be done by analyzing the environment of the entities and observing their actions, so that the representation of this behavior is mainly focused on sensory inputs and signals sent to actuators.

Computationally, their representation is essentially numerical and it is based on the assignation of values to different parameters. This makes them more efficient than heavyweight agents.

On the other hand, heavyweight agents are based on the cognitive part of artificial intelligence, where each agent tries to reach the level of human intelligence [5]. Therefore, in this case the intelligence resides in each of the agents that take part in the system. They have the advantage over lightweight agents, that the fact of being developed taking into account human knowledge, facilitates understanding and communication with users. For example, the concept of "emergence" of an agent corresponds to the concept of human "emergency" resulting easier to understand.

Other examples of conceptual representations developed by heavyweight agents are beliefs, desires, intentions or goals, plans, etc. [6].

2.2. Swarm Intelligence

As is known, a swarm is a set of single agents (such as the case of the mentioned lightweight agents) usually identical, cooperating among each other and their environment to complete a common objective or task with the ability of intelligent behavior [7].

In recent years, swarm based algorithms have become more and more popular due to their capabilities to provide solutions to complex problems. Some of these algorithms can be found in [8]. Problems trying to provide solutions swarm intelligence are mainly related to artificial intelligence (AI), data mining [9] and robotics [10]. These problems include optimization tasks, data analysis and pictures, optimal routes, etc. As shown in [7], the main advantages of swarm intelligence are the robustness, adaptability and self-organization, flexibility, scalability and decentralization, getting a good solution to distributed problems.

3. Architecture Overview

To design the platform proposed in this project, special attention is paid to open systems. Open systems exist in dynamic operating environments, in which new components may continuously join the system, or already existing components may leave it, and in which the operation conditions might unpredictability change. Therefore, open systems are characterized by the heterogeneity of their participants, their limited confidence, possible individual goals in conflict and a great probability of non-accordance with specifications.

3.1. Platform Layers Description

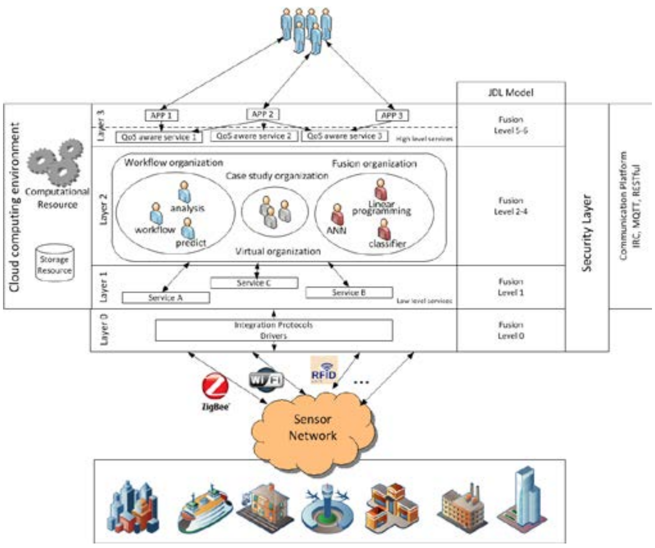

Fig. 1 - Platform architecture

Under the frame of this work, we proposed a new open platform, especially designed for the integration of heterogeneous sensor, through lightweight agents, and intelligent management of automated environments be means of heavyweight agents. The platform will incorporate advanced algorithms for intelligent information management and novel mechanisms to provide security for the information and communication within the platform. We propose to use a VO of MAS to facilitate the design of the platform, which will be deployed in a cloud environment allowing sustainable growth, and adapted to the needs of deployment.

The platform is divided into a layered architecture as shown in Fig. 1. As we can see, the platform is composed of different layers associated with the different functional blocks. These levels are distributed in the different layers of the architecture shown in Fig. 1.

The following section describes the components and main features of the architecture:

- **Layer 0. Integrate sensing/performance technologies**. The platform will facilitate integration sensing technologies currently available, regardless of their nature, and will provide an open environment that allows the dynamic addition of new sensor systems and technologies. In order to accomplish this, the platform will provide data encapsulation mechanisms that standardize the information received from sources such as Wi-Fi, ZigBee, etc. Layer 0 of the platform is a broker that defines communication with sensor networks of different natures (Wi-Fi, ZigBee, Bluetooth, etc.), and obtains the raw data from sensor networks. The main novelty of this layer is the ability to provide the platform and the upper layers with openness regarding the connection to sensor networks of different natures, and thus ensure that upper layers of the architecture have access to information and are able to perform data fusion at different levels.
- **Layer 1. Low-level services**. Given the information exchanged with the environment through layer 0 as described above, the existing functional requirements and a set of low-level services will now be defined; specifically those that depend on the types of networks and technologies integrated into every deployment. After obtaining the raw data, a gateway is provided, defined through adapters that allow the information received to be standardized. In this first stage, the platform provides services such as filtering of signals, normalization services or other treatment services at the basic level signals. These services are provided by the adapters and will be associated with algorithms that perform initial treatment of the data, so that these data can be presented to higher layers in a more homogenized way.
- **Layer 2. Information fusion algorithms**. The platform is structured as a VO of MultiAgent System (MAS). Each organization includes the roles required to facilitate an intelligent management of the information obtained from the lower levels of the architecture. The MAS incorporates agents specifically designed to interact with low-level services. In addition, we propose the design of intelligent agents specialized in Information Fusion (IF). For this purpose, roles that allow merging information automatically through supervised learning and previous training have been included. This procedure can be applied in cases such as the mixture of experts. Statistical techniques will be added based on linear programming, neural networks and classifiers. Thus, specialized agents in IF can combine different sensing technologies that provide heterogeneous data and provide more accurate information to the upper layer services. In addition to providing IF techniques, this layer will provide an automatic generation of IF flows between different levels by including the organization of agents. This organization will contain roles such as analysis workflow, prediction workflow and workflow processing. The analysis workflow is responsible for analyzing a given workflow and for creating a new workflow automatically. The workflows will combine different techniques and algorithms integrated into the virtual organization for data processing for a specific case study. To carry out the analysis and prediction of workflow, statistical techniques based on Bayesian networks were applied to estimate the most appropriate execution flows.
- **Layer 3. High-level services**. The top layer of the platform provides an innovative module that allows management and customization of services to end users, due to the capabilities of the MAS which is deployed over the cloud computing environment. The tasks associated with the man-

machine interfaces are performed at these levels in order to adapt according to the characteristics of the user and facilitate decision making by the user.

4. Case Study

4.1. Smart City Platform: A Traffic Application

Nowadays traffic flow control is a major problem both in large cities as in smaller cities. The main problem in this area is jams formed due to a non-optimal utilization of the pathways' capacity. Within this scope, the proposed architecture is adapted to the application environment, allowing address this problem. It addresses the problem of gathering data from multiple sensors, automate the processing and generate useful information that can be used effectively in managing traffic flow automatically.

To carry out the demonstration, and since test the system in a real environment would be a high cost when necessary to establish the specific sensing or adapt existing ones, an environment that simulates the operation of a real city has been developed. This environment consists of a 3D printed model. With this recreation, real data are received with which we are able to make appropriate tests to validate the system before deploying it in a real city.

- **Sensor network:** In order to understand the architecture, it is necessary to start with the lower part, the sensor network. The first problem that arises in the case study is the need of tracking the vehicle flow across the stretch of roads that integrate the system.

Although tests will be done with the simulated environment, there are different alternatives sensing adaptable to real environments, able to interact with the system in the same way. To locate vehicles in the simulation environment, a series of magnets are distributed parallel to steel rails on which vehicles circulate. These vehicles are equipped with a hall sensor that detects the presence of those magnets, placed in a specific way and order that defines the streets of the city where they are and the position of each street with an accuracy of 1 to 2 mm.

Each vehicle in the city is part of the swarm and is considered a lightweight agent with his position as the only knowledge.

For the real system, cameras deployed over the traffic lights will be used. This way vehicles flow is analyzed with a piece of software (Fig. 2), which processes the images in real time.

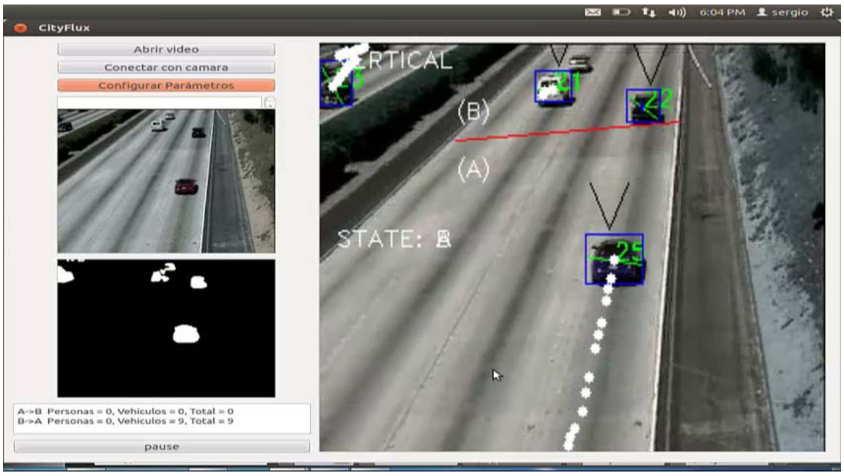

Fig. 2 - CityFlux, developed software for real traffic detection

- **Layer 0:** This layer is responsible for receiving data from different protocols through communication standards used to subsequently serve to a higher layer. It refers to a broker where a specific software for each standard runs. In all cases, software have been developed using different libraries and tools available for each technology. For example, the case study of BLE is based on the Gattool, which allows us to use BLE services through a series of simple commands. Just a simple configuration that varies depending on the type of communication to use is needed. For example, to BLE one needs to know the MAC address of the device. In the case study BLE is present in the vehicle. So far, in the middleware are integrated communications via Wi-Fi (a, b, g, n and c are treated the same way), RFID, BLE and ZigBee.
- **Layer 1:** This layer translates all data received from the layer 0 to a common protocol. So, at this level, no matter how sensors obtained the data but the data itself, which must be encapsulated to provide it in a well defined manner to the upper layer by low-level services. In this case, the protocol used to serve data in a homogeneous manner is MQTT (Message Queuing Telemetry Transport). This open protocol connectivity M2M (Machine-to-Machine) allows sending telemetry data as a message. It is also a very lightweight protocol. Data are available in a well-defined way in the system hierarchical topics. These topics are conceptually associated with the information they represent, for example, the position of the vehicle is published (served at higher layers) on the topic '/vehicles/location' where 'vehicles' is the root topic of our vehicle system and 'location' is one of the topics that are within 'vehicles' topic. These data represent the information lightweight agents.
- **Layer 2:** The second layer acts directly on the previous layer, obtaining the services corresponding to the needed information. In this layer heavyweight agents are located and they are structured in VO. Thus the system is divided into less complex structures and simplicity. In this layer, heavyweight agents receive the information. They have more functionality than lightweight agents and they are specialized in IF. Their mission is to communicate this information gained from other agents in this layer, apply the necessary data transformations and serve them to the layer 3. For the present case, moreover, there VO composed of agents that apply statistical techniques and, specifically to this case study, there are agents responsible for implementing algorithm H-ABC (Hierar-

chical Ant Based Control) [11], based algorithm Ant Based Control (ABC) but it can be applied to large networks as in the case of city streets.

This system is scalable and therefore is applicable both to the demonstration environment, and in a real city. Its implementation is based on the authors presented in [12], but with several differences. The main new proposal is that there is no need for vehicles to carry a GPS sensor, it is the system itself which locates them. Furthermore, this proposal is able to automatically detect events such as accidents or other situations that block access in some way by analyzing the flow, which will be reduced in a sounded way and algorithms based on neural networks will detect the alert, acting accordingly.
The system performs dynamic routing system maintaining a route table for each intersection based on probabilistic results provided by the corresponding agents.
Once the decision is made, the states of the traffic lights will be modified using the services defined high level in the upper layer.

- **Layer 3:** In the top layer of the architecture a series of high-level services are presented to end users to monitor and control the system. System status can be accessed via web services or using the Node.js technology for current study. This allows real-time interaction with the system, either by defining different routes, managing existing intersections or adding new ones or monitoring the state of the whole system. Different apps developed for this purpose.

4.2. Results

When carrying out initial simulation testing with the model, different vehicle location benefits appear over existing commercial systems. Precisely, this system does not require any sensor addition in the environment as in [13]. As already mentioned, it is the vehicle itself that knows, with a pinpoint accuracy, the exact point where it is. This provides a higher accuracy in the most critical points and an economic saving due to the fact that this system is based in magnets whose magnetic field is read by a hall sensor, as well as, the vehicle reaction time is immediate.

Nevertheless, real systems are affected by additional variables such as rail speed or number of lanes. Also, to test the real system effectiveness, it is necessary to know the required time to do each one of the possible routes. Due to the fact that speed is constant in the simulation environment the tests are performed in a second simulation system. In this occasion, the system will be software simulation where new real scenery needs are added. To be able to compare the results, the simulation in [12] has been followed using the same algorithm. Fig. 3 shows a schema of the simulated scenery.

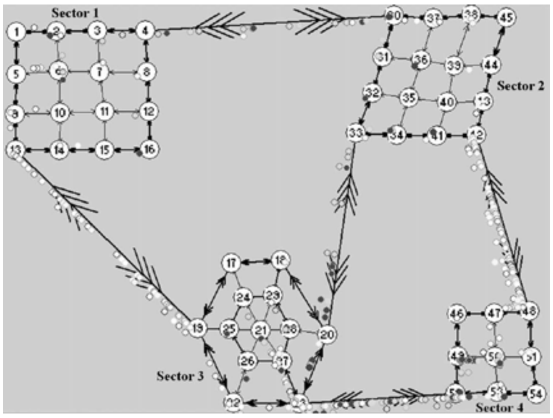

Fig. 3 - Simulated scenario with the simulation software

Fig. 4 shows a graphic with the average route time for standard cars, cars with an intelligent device with GPS and standard cars located by the system following the proposal for real environments.

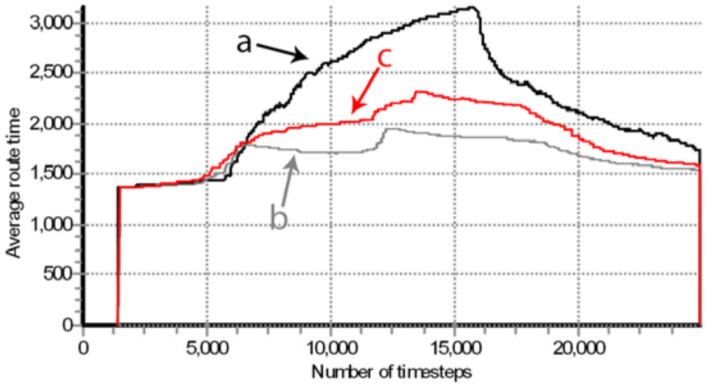

Fig. 4 - Trip time of standard vehicles (a), vehicles with location device (b) and standard vehicles with the proposed system (c)

The total number of time steps in the simulation corresponds to 5000 seconds. The average 'smart car' time grows up to 1800 time steps and then it stabilizes around 1700 time steps. As we can see, very similar results are obtained with the proposed system. This system constitutes an economic saving for the drivers as well as an added safety for them thanks to not having to use digital devices to, for instance, indicate accidents.

4.3. Conclusions and Future Work

As a result, a scalable architecture capable of dealing jointly or independently all kinds of sensor that smart city may need, regardless of the technology used. The size of the swarm formed is not a problem for the architecture and the cost is directly linked to the use of the system, which allows it to be implemented both in small and medium cities.

As future work, we intend to implement the system in a real environment, starting with small neighborhoods or towns of Salamanca (Spain). This will serve to make some studies of the installation, service and maintenance costs when implementing the proposed platform for small and medium cities to be smarter cities and the saving that the system can offer to every kind of town.

5. Acknowledgements

The research of Pablo Chamoso has been financed by the Regional Ministry of Education in Castilla y León and the European Social Fund (Operational Programme 2014-2020 for Castilla y León, EDU/310/2015 BOCYL) and the project EKRUCAmI: Europe-Korea Research on Ubiquitous Computing and Ambient Intelligence. Ref. 318878. FP7-PEOPLE-2012-IRSES.

6. References

[1]. Nam, T., & Pardo, T. A. (2011, June). Conceptualizing smart city with dimensions of technology, people, and institutions. In Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times (pp. 282-291). ACM. http://dx.doi.org/10.1145/2037556.2037602

[2]. Renuka, N., Nan, N. C., & Ismail, W. (2013, September). Embedded RFID tracking system for hospital application using WSN platform. In RFID-Technologies and Applications (RFID-TA), 2013 IEEE International Conference on (pp. 1-5). IEEE.

[3]. Daniel, F., Eriksson, J., Finne, N., Fuchs, H., Gaglione, A., Karnouskos, S., ... & Voigt, T. (2013, April). makeSense: Real-world Business Processes through Wireless Sensor Networks. In CONET/UBICITEC (pp. 58-72).

[4]. Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). Swarm intelligence: from natural to artificial systems (No. 1). Oxford university press.

[5]. Wooldridge, M. J. (2000). Reasoning about rational agents. MIT press.

[6]. Rao, A. S., & Georgeff, M. P. (1991). Modeling rational agents within a BDI-architecture. KR, 91, 473-484.

[7]. Bin, W., & Zhongzhi, S. (2001). A clustering algorithm based on swarm intelligence. In Info-tech and Info-net, 2001. Proceedings. ICII 2001-Beijing. 2001 International Conferences on (Vol. 3, pp. 58-66). IEEE. http://dx.doi.org/10.1109/ICII.2001.983036

[8]. Marciniak, A., Kowal, M., Filipczuk, P., & Korbicz, J. (2014). Swarm Intelligence Algorithms for Multi-level Image Thresholding. In Intelligent Systems in Technical and Medical Diagnostics (pp. 301-311). Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-39881-0_25

[9]. Martens, D., Baesens, B., & Fawcett, T. (2011). Editorial survey: swarm intelligence for data mining. Machine Learning, 82(1), 1-42. http://dx.doi.org/10.1007/s10994-010-5216-5

[10]. Chamoso, P., Perez, A., Rodriguez, S., Corchado, J. M., Sempere, M., Rizo, R., ... & Pujol, M. (2014, July). Modeling Oil-Spill Detection with multirotor systems based on multi-agent systems. In Information Fusion (FUSION), 2014 17th International Conference on (pp. 1-8). IEEE.

[11]. Tatomir, B., & Rothkrantz, L. J. M. (2005). H-ABC: A scalable dynamic routing algorithm. Recent Advances in Artificial Life, 5, 8. ISO 690. http://dx.doi.org/10.1142/9789812701497_0021

[12]. Tatomir, B., & Rothkrantz, L. (2006, September). Hierarchical routing in traffic using swarm-intelligence. In Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE (pp. 230-235). IEEE. http://dx.doi.org/10.1109/itsc.2006.1706747

Wagner, M. (19/03/2015) Car System Digital Forum. Germany.Gebr. FALLER GmbH. https://www.car-system-digital.de/de/