Multi-Agent based governance model for Machine-to-Machine networks in a smart parking management system

Mustapha Bilal* Camille Persson*†, Fano Ramparany*, Gauthier Picard† and Olivier Boissier†
*Orange Labs Network and Carrier, TECH/MATIS Grenoble, France {firstname.lastname}@orange.com
†Fayol Institute, ISCOD, ENSM-SE, Saint-Etienne, France {lastname}@emse.fr

***Abstract*—Proposed in this paper is a multi-agent model that defines a set of global functioning rules for a flexible governance, adapted to parking management within a city. This is designed to aid drivers in finding a parking place, which satisfies a group of criteria, predefined in profiles, providing a better parking service to the public. The Multi-Agent model developed is integrated in the platform SensCity, which is dedicated to the development and deployment of Machine-to-Machine (M2M) systems. The city is divided into a number of parking areas that are equipped with sensors and actuators, which are responsible for transferring data from and to the parking places. Therefore, the agents can work to interpret and manipulate the governance principles modeled and implemented by the multi-agent model, independently from drivers and parking spaces. Moreover, this paper proposes an intelligent end-to-end management of parking system using the $\mathcal{MOISE}$ organization framework.**

***Index Terms*—Parking; Multi-Agent System; Governance; Machine-to-Machine; SensCity; agents; sensors.**

I. Introduction

Real-time parking information is a feature of Smart Cities that provide useful information to their citizens, in order to improve their quality of life and to help them meet important daily decisions. Modern cities have a strong need for advanced parking systems to provide drivers with information about any parking spaces. It is a common issue known by all drivers who have spent hours searching for a parking space (eg. 17.7 minutes in the neighborhood of Saint-Germain - Paris [1]). This search for a free parking place gives rise to traffic congestion, the frustration of driver and increases pollution.

Smart parking applications monitor the parkings availability in the city to guide drivers. Such an application raises three main issues: (i) how to monitor the parking places efficiently, (ii) how to select a suitable parking place and (III) how to guide the driver.

The contribution of this paper is to propose an end-to-end intelligent parking application based on a M2M platform, using a Multi-Agent System (MAS). This is done in order to meet drivers' preferences, and adapt its requirements to save sensors' batteries consequently.

[1]Survey done in France - year 2005

After presenting the existing parking systems in Section II, a proposal for intelligent management of parking places is presented in Section III. Then, Section IV discusses the approach. Finally, before an acknowledgment in Section VI, Section V concludes this paper, drawing the orientation to further work.

II. Related work

Finding a free parking place raises the importance of developing a suitable solution for the two different sub-problems: (i) Surveillance of parking places and (ii) Selection of parking places.

This section reviews the state of the art contributing to these two sub-problems, based on different technologies.

A. Surveillance of the parking space

Different approaches are used for parking surveillance: surveillance cameras, VANET (Vehicle Ad Hoc Networks) and wireless sensors networks.

Image processing of surveillance cameras video is used by [1] to monitor the parking spaces. This technology requires a sizeable computing power to run the image processing algorithms. However, taking advantage of the existing surveillance cameras can reduce the cost of such a system.

However, [2] found that the sensors can become inaccurate and would stop functioning easily when they age. Therefore, they used the VANET technology to track the available parking places through road side units, that survey a number of places and communicate with some embedded units (on board units - OBU) placed in each vehicle. A local communication takes place between the OBU and road side units (RSU), which continuously transmits the parking space occupancy status to the appropriate servers and consequently to the drivers. This keeps the information up-to-date, but requires the car to be equipped with a VANET connexion.

The WSN approach appears to be used the most by both academics [3] [4] [5] [6] and industrials [2] [3] [4] [5].

[2]http://www.intelligentparking.com
[3] http://www.lyberta.com/
[4]http://www.optipark.eu/
[5]http://www.streetlinenetworks.com/

Sensors are placed under parking spaces and communicate via a low power wireless network to a gateway. This gateway is connected by broadband to a server that allows transparent access to the sensors. Such sensors are very low cost and non-intrusive. Thus, they are suitable to be deployed on a large scale and have low power consumption [7]. Nevertheless, maintenance costs a lot, and therefore it is necessary to have an efficient way messages can be sent correctly, in order to save the sensors battery life.

B. *Selection of a suitable parking space*

The driver can be informed of the nearest parking place in advance [2] [3] [4]. This can reduce congestion, but it requires a complete communication system, in order to transmit the availability of places to drivers.

Another systems show the positions of available free places and leave it to the driver to decide where to park [1] [8]. The advantage to these kind of systems is that they are considered low cost, however, they are not fully automated parking solutions.

The use of MAS allows for predefined profiles of each driver, in order to set multiple criteria for the placing space [5] [9] [10]. Two parking information systems have to be distinguished by their behavior in making reservations [6]. One that guarantees the reservation and the other gives an indication of the possibility of finding a parking place. Such systems are fully automated, however, ignore the sensors battery life.

C. *Summary*

The scenario presented in this paper uses: (i) wireless sensors to monitor the parking spaces, (ii) an information and communication system to inform drivers about the nearest parking place to their destination, and (iii) a MAS which is a natural and effective solution to work with complex situations in distributed environments. The efficient management of sensors, and taking into consideration the generic preferences, are two related problems to be resolved within this paper.

Section (III) presents a proposal for the intelligent end-to-end management of the parking system.

III. Smart parking management

This section describes the proposal for a smart parking management, which is based on the SensCity[6] M2M platform. Following the M2M architecture [11], this proposal is divided into 3 subdomains: (i) the Application domain guides the drivers, (ii) the Network domain manages the devices and (iii) the Device domain monitors the parking spaces (see Figure 1).

Two main constraints must be taken into account: (i) sensors usage and (ii) driver preferences. Moreover, these two constraints are related, as the sensors usage can be determined depending on the drivers demand. The remainder of this section describes the Multi-Agent Organization of the proposal (Section III-A), then Section III-B describes how the driver's preferences are taken into account, finally Section III-C shows how the system adapts its needs to the sensors informations.

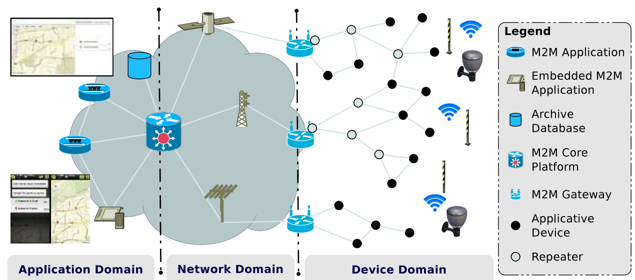

Fig. 1: End-to-end M2M architecture

A. *Smart parking governance principles*

Given the decentralized and dynamic characteristics of the application, the choice of using a MAS is made because it has properties that are specifically adapted to the governance of M2M systems. The governance's model is based on three dimensions:

- *Organization*: governance strategy. It defines the global functioning of the system by means of roles and goals organized into coordination schemes. The organization modeling language $\mathcal{MOISE}$ [12] is used to define an organization within a MAS according to three independent dimensions: (i) structural specification (SS), and (ii) functional specification (FS), connected together by the normative specification (NS).

- *Agents*: perform the algorithms and actions to fulfill the organization's goals, depending on their roles.

- *Artifacts*: represent the passive entities of the MAS. They are used as an abstraction of the users preferences and parking lots characteristics.

The roles *parkingAreaMgr*, *authenticationNAccountingMgr*, *driverGuide* and *driverPrefMgr* specify the behavior authorized for an agent in the organization, through the set of activities this agent can exert.

The parking system's organization is structured into two groups: gpSmartParking and gpDriver (see Figure 2).

The functional specification controls the goals and missions of the parking system, through Figure 3. This cooperation scheme defines the goals, in order to fulfill the primary goal (*driverParked*) as a sequence: (1)

[6]http://www.senscity-grenoble.com

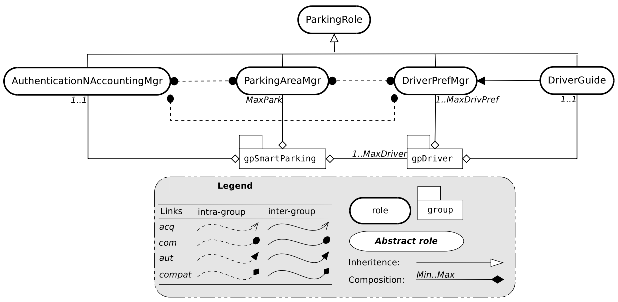

Fig. 2: The structural specification for the smart parking system.

driverAuthenticated, (2) *driverBill* which is a maintenance goal to bill the driver depending on his usage for the system, (3) *bidForPlacesResponsed* to inventory the parking spaces characteristics, (4) *bestPlaceSelected* following the users preferences and (5) *goneToPlace* which is satisfied when the driver reaches his destination.

An instance of this scheme is started each time a driver requests the smart parking service. The agents are then assigned missions, which group a consistent set of goals, by a norm: a deontic relation with a deadline, triggered by a condition.

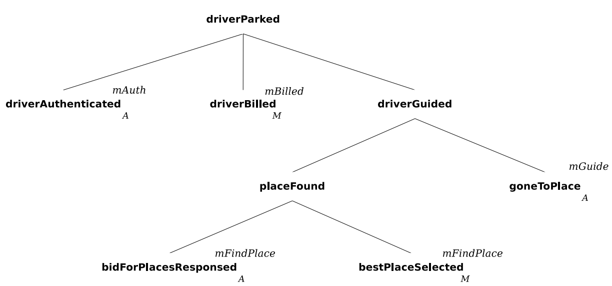

Fig. 3: The functional specification for the smart parking system.

B. Individualized parking selection process

This process corresponds to the role *driverPrefMgr*, which is played by *ParkSelectorAg*. *ParkSelectorAg* selects one of the parking places proposed by the agents playing the role *parkingAreaMgr* and always does this according to the user's profiles.

ParkSelectorAg selects its recipients that are located in a radius of the destination and groups all the agents *ParkingAreaAg* located within this radius. If a place is not found inside that radius, the area of search is increased.

The criteria form an evaluation function for a property. Some properties are defined in Table I, which groups examples of preferences criteria on the properties of zone parking places. Every user can have several predefined profiles (work, holiday, weekend, etc). Every profile forms a set of criteria on the property of the parking lots, and a criterion is a function of evaluation of a property that returns *1* if the criterion is satisfied and *0* otherwise.

TABLE I: Example of parking area properties

Proprieties	Description
$Location$	Parking area location
$Price$	Parking places' hourly price
$PlaceType$	Place's type (eg. Ordinary (O), for Handicaps (H), equiped with Electrical charger (E)...)
$Duration$	Maximum parking duration
$Availability$	Parking area occupancy rate

The system allows for the addition of any new criteria under the condition of having a property for the place, and to be able to put a criterion over this property.

The criteria are measured by the level of importance, which are ordered in a decreasing scale, shown as follows: *Mandatory (M)*, *Important (I)* and *Optional (O)*. For example, if a criterion on the property *distance* has an importance *(M)*, this property will be satisfied before the others, which are considered less important. Several criteria can be of the same importance. The system will select the place that satisfies the most criterion of the greatest importance. Equation 1 $eval(P)$ evaluates a place P compared to the list of the $criteria$, which corresponds to the preferences of the user. It returns a numerical value equal to the sum of the satisfied criteria weighted by their importance, where $cr_i(P)$ is equal to 1 if the criterion cr_i is satisfied by the place P. $N_{Cr}^{cr_i.imp}$ is the number of criteria N_{Cr} to the power of the criterion's importance $cr_i.imp$,.

$$eval(P) = \sum_{cr_i \in criteria} cr_i(P) * N_{cr}^{cr_i.imp} \tag{1}$$

$$eval_{Imp}(P) = \sum_{\substack{cr_i \\ cr_i.imp=Imp}} cr_i(P) \tag{2}$$

As described in Algorithm 1, the *ParkSelectorAg* agent compares the parking zones in order to select the best zone. A list of the places is sent to this algorithm, which will first consider that the first place on the list is the best place found. Then, it will look at all the places on the list, evaluating each time a place on the list, with the best place already found. The selection of the best place is done by *selectBest(BestPlace,P)*, that uses either the evaluation Equation 1 or Equation 2. If a better place P is found, this place will replace $BestPlace$. Following Equation 1 $eval(P)$, each homogeneous set of places P is evaluated by the number of satisfied user criteria, weighted by their importance.

Furthermore, a min-max optimization can be used to reduce the complexity of such an evaluation. This is done by comparing first the evaluations of primary importance, given by Equation 2 $eval_{Imp}$, of the two set of places.

Algorithm 1 The algorithm *findPlace* that determines the place which corresponds best to the preferences of the user.

Require: $Places = \{P_0..P_{max}\}$: list of possible places
Ensure: $BestPlace$: the chosen place which matches the best the user preferences

```
Imp ← {M, I, O} {Importance sequence: Mandatory, Important, Optional}
BestPlace ← P_0
for (P in Places\P_0) do
    BestPlace ← selectBest(BestPlace, P)
end for
return BestPlace
```

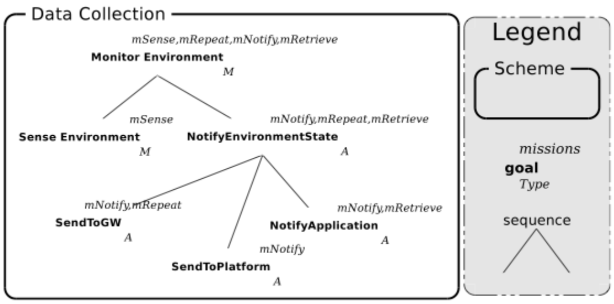

Fig. 4: DataCollectionScheme:"Goals to achive" in order to collect data from sensors to the application

C. Smart parking monitoring

This process corresponds to the role *parkingAreaMgr*, which is played by *ParkingAreaAg*. A *ParkingAreaAg* agent manages a parking area representation by using the sensors information. To do so, every *ParkingAreaAg* is considered a client application of the SensCity platform. This platform offers transparent access to the sensors and actuators deployed in the city.

A Multi-Agent Organization can be used to manage the SensCity M2M infrastructure [12]. This organization defines collaboration schemes involving agents managing the client applications, the M2M platform, the gateways and the devices (sensors and actuators). For example, Figure 4 gives a representation of the Data-CollectionScheme. The root goal *Monitor Environment*, is a maintenance goal which is satisfied by the sequence (i) *Sense Environment* and (ii) *NotifyEnvironmentState*. The first sub-goal is satisfied by the mission *mSense*. The notification of environment state can be done after each or multiple sensing. The second sub-goal is of type achievement, it is achieved by the sequence: (i) *SendToGW*, send to the gateway, (ii) *SendToPlatform* and (iii) *NotifyApplication*.

The SLA's can be defined as a set of norms (see Listing 1) specifying the frequency, delay of messages (for devices) and authorization requests (for applications). Examining the case of devices, each device has to periodically send a message in order to transmit the information about the status of the parking place. The interval of time separating each transmitted message (delay of messages) is defined by a set of norms. Norm n_{01} means that *ParkingApp* has permission to send commands to the devices. Norm n_{02} means that the sensors are forced to treat the command in time *ttfSensor*.

```
</normative-specification>

  <!-- Some norms relative to the schDevCtrl
       scheme -->

  <norm id="n01"  type="permission"
        role="parkingAppMgr"  mission="
             mAppCmd"/>
  <norm id="n02"  type="obligation"
        role="parkingSensor"  mission="
             mDevCmd"
        condition="command(sensor_request(Cmd
             ),Devs,_)"
        time-constraint="&ttfSensor;"/>

  <!-- Some norms relative to the schDevExec
       scheme -->

  <norm id="n06"  type="obligation"
        role="parkingSensor"  mission="
             mDevMonit"
        time-constraint="&ttfmDevMonit;"/>
  <norm id="n07"  type="obligation"
        role="parkingSensor"  mission="
             mDevRepeat"
        time-constraint="&ttfmDevRepeat;"/>
  <norm id="n08"  type="obligation"
        role="parkingSensor"  mission="
             mDevNotif"
        time-constraint="&ttfmDevNotif;"/>

</normative-specification>
```

Listing 1: Example of XML code for a set of norms in the Normative Specification (NF)

The *ParkingAreaAg* agents adopts the *parkingAppMgr* role in this organization. When instantiating a scheme, these agents can parameterize the norms to adapt the requirements to the demand.

IV. Discussion

This paper proposed an intelligent parking management system integrated in the platform SensCity. The model developed takes into account the user's and resources' constraints, providing a solution for the parking problem. This is done by taking advantage of the properties of a parking zone, to satisfy the user's requirements in the best possible way.

The method of evaluation of a list of parking places followed in this paper, refers to the fact that one satisfied criterion, having the importance *Mandatory*, can replace a larger number of satisfied criterion, that are considered less important. However, prioritization of criteria might be adapted, as a large number of satisfied low priority criteria could be preferable to a single high priority criterion. This could be achieved by using an agent per criterion, in order to negotiate with the other agents. But this may require the user to be able to prioritize his criteria more specifically.

Smartphones are a very convenient way to interact with the users. Such a deployment could be done using the JaCa-Android [7], which provides a framework for developing multi-agent systems on Android devices.

There should be a difference between the distance separating a driver from his destination and the time he should take to arrive to his destination. Taking into consideration the real time traffic information, can be a suitable solution to show these two properties (distance and time) in the presented system. Moreover, using the system can be extended to the use of *multimodal transport* and other resources of smart cities.

The infrastructure of SensCity allows the use of actuators and barriers. An agent can be added to the system to assure the reservation of a place for the drivers by the barriers.

V. Conclusion and perspectives

The continued problem of searching for parking places in the city, has lead to the need for the development of an intelligent parking place management system, which is based on Multi-Agent technologies. Implementation, testing and validation are done thanks to the tool JaCaMo. The agents are developed in Jason, the environment in CArtAgO and the organizational structure in $\mathcal{MOISE}$.

Updating the intelligent parking management system is possible. However, if the user does not find a place which corresponds completely to his criteria, the agent starts to eliminate a criterion, of least important, one after the other until ultimately the user manages to find a place that satisfies his preferences. Last but not least, an exchange of places between the drivers can be used to improve the entire system.

VI. Acknowledgment

This work took place in the context of the Minalogic SensCity project, funded by the Interministerial Unified Fund (FUI). The authors would like to thank Andrzej Duda for his comments and the computer science lab of Grenoble (LIG[8]) for the support and funding of this paper submission.

References

[1] M. Y. I. Idris, E. Tamil, Z. Razak, N. Noor, and L. Kin, "Smart Parking System using image processing techniques in wireless sensors network environment," 2009.

[2] R. Lu, X. Lin, H. Zhu, and X. Shen, *SPARK: A New VANET-Based Smart Parking Scheme for Large Parking Lots*. IEEE, Apr. 2009.

[3] G. Yan, S. Olariu, M. C. Weigle, and M. Abuelela, "SmartParking: A Secure and Intelligent Parking System Using NOTICE," *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pp. 569–574, Oct. 2008.

[4] S. Srikanth, P. Pramod, K. Dileep, S. Tapas, M. U. Patil, and S. C. Babu N, "Design and Implementation of a Prototype Smart PARKing (SPARK) System Using Wireless Sensor Networks," *International Conference on Advanced Information Networking and Applications Workshops*, pp. 401–406, May 2009.

[5] G. Ivan, M. O'Droma, and M. Damien, "Intelligent Car Parking Locator Service," *Information Technologies and Knowledge*, vol. 2, pp. 166–173, 2008.

[6] S.-Y. Chou, S.-W. Lin, and C.-C. Li, "Dynamic parking negotiation and guidance using an agent-based platform," *Expert Systems with Applications*, vol. 35, no. 3, pp. 805–817, Oct. 2008.

[7] S. Anandamurugan and C. Venkatesh, "Reducing Power Consumption in Wireless Sensor Networks by using Terrain Based Probabilistic Forwarding Scheme (TBPFS)," *Journal of Computer Science*, vol. 10, no. 2, pp. 208–217, 2010.

[8] Z. Pala and N. Inan, "Smart parking applications using RFID technology," *International Journal*, pp. 21–23, 2007.

[9] D. Teodorovit and M. Dell'Orco, "Multi Agent Systems Approach To Parking Facilities Management," in *Applied Research in Uncertainty Modeling and Analysis*. Springer, 2005, pp. 321–339.

[10] A. B. Juan and A. Muñoz, "Developing an Intelligent Parking Management Application Based on Multi-agent Systems and Semantic Web Technologies," *Hybrid Artificial Intelligence Systems*, pp. 64–72, 2010.

[11] ETSI, "ETSI TS 102 690 V<0.10.1>, Machine-to-Machine communications (M2M); Functional architecture," pp. 1–201, 2011.

[12] C. Persson, G. Picard, and F. Ramparany, "A multi-agent organization for the governance of machine-to-machine systems," in *IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'11)*. IEEE Computer Society, 2011, pp. 421–424.

[7] http://jaca-android.sourceforge.net/

[8] LIG, Drakkar team, ENSIMAG, Grenoble, France www.liglab.fr