Agent Negotiation for Different Needs in Smart Parking Allocation*

Claudia Di Napoli[1], Dario Di Nocera[2,**], and Silvia Rossi[3]

[1] Istituto di Calcolo e Reti ad Alte Prestazioni
Consiglio Nazionale delle Ricerche, Napoli, Italy
claudia.dinapoli@cnr.it

[2] Dipartimento di Matematica
Università degli Studi di Napoli "Federico II", Napoli, Italy
dario.dinocera@unina.it

[3] Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione
Università degli Studi di Napoli "Federico II", Napoli, Italy
silvia.rossi@unina.it

Abstract. Smart Cities are experiencing a growing interest from different research areas. One of the challenges of Smart Cities is the design of an effective City Parking System that may contribute to improve the city life in terms of gas emission and air pollution in city centers, but also the everyday life of city dwellers by facilitating to park with the support of automatic parking services. In this work, an investigation on the use of software agents negotiation to accommodate both user and vendor requirements on a parking space is carried out. It is shown that agent negotiation allows to assign parking spaces in an automatic and intelligent manner by taking into account that users have their own needs regarding parking location and price, while parking vendors have their own needs regarding efficient allocation of parking spaces, and city regulations.

Keywords: Agent negotiation, multi-agent systems, smart parking, smart cities.

1 Introduction

Smart Cities initiatives are focused on different themes relevant to increase the state of innovation of European and worldwide cities in order to: increase the quality of life of city-dwellers, enhance the efficiency and competitiveness of the economy, move towards the sustainability of cities by improving resource efficiency and meeting emission reduction targets.

* The research leading to these results has received funding from the EU FP7-ICT-2012-8 under the MIDAS Project (Model and Inference Driven - Automated testing of Services architectures), Grant Agreement no. 318786, and the Italian Ministry of University and Research and EU under the PON OR.C.HE.S.T.R.A. project (ORganization of Cultural HEritage for Smart Tourism and Real-time Accessibility).

** Ph.D. scholarship funded by Media Motive S.r.l Napoli (Italy), POR Campania FSE 2007-2013.

Y. Demazeau et al. (Eds.): PAAMS 2014, LNAI 8473, pp. 98–109, 2014.

One of the challenging problems to be addressed is parking in urban areas. It is widely recognized that drivers searching for a parking space in wide urban areas waste time and fuel, so increasing traffic congestion and air pollution [7]. It is not always possible to address the problem by creating more parking spaces, but rather "intelligent" parking facilities are necessary.

The use of advanced technologies, including vehicle sensors, wireless communications, and data analytics, is the base for the efficient allocation, monitoring, and management of smart parking solutions for future Smart Cities in order to improve urban mobility strategies. Most of the research projects concerning smart parking systems focus on ways to collect and publish live parking information to drivers so they can be informed of available parking spaces near to the destination they require. At the same time, many companies are developing electronic parking systems allowing for a wide variety of available payment methods in conjunction with the dissemination of parking availability information. Nevertheless, they lack of intelligent features allowing not only to advise motorists of available car parks in multiple zones, but more importantly to help them in making decisions on where to park.

Mechanisms to manage the relationship between supply and demand are necessary to provide user-oriented automatic parking services that take into account both drivers preferences, and parking vendors requirements together with social benefits for the city, such as a reduction of traffic in city centers by limiting parking in that area [8].

In this context, we investigate the possibility to use software agent negotiation to address some of the challenges concerning smart parking and mobility pricing strategies. Software agents are software programs situated in some environment, continuously active, capable of autonomous actions (either proactive or reactive), and of working on tasks on behalf of users. These programs differ from regular software because they are personalized, continuously running, and to a certain extent autonomous, so making them suitable to assist buyers in the search and selection of products [5]. Software agents are able to communicate with other agents, and to negotiate over a set of issues [3]. Automated software agent negotiation is crucial to address the demands for systems composed of agents that represent different individuals or organizations and that are capable of reaching agreements through negotiation [4].

The present work proposes an automated negotiation mechanism among a software agent that models a Parking Manager responsible for providing parking spaces, and a software agent acting on behalf of a motorist user searching for a parking space in the city center of a urban area. Negotiation is used in order to accommodate both users and providers needs that are different and, more importantly, conflicting. In fact, the Parking Manager has the objective to sell parking spaces to make a profit, but to prevent, as much as possible, motorists to park in the city center, while users would prefer to save as much money as possible, but to park close to the city center location they require. The allocation of the parking space is the result of a negotiation process between the Parking

Manager and the user having their own private utility functions respectively to make a parking space offer, and to evaluate whether to accept a received offer.

2 Automated Negotiation for Parking Allocation

Usually, parking applications provide users with available parking spaces among which to select the preferred one according to their own preferences, if possible. In the Smart Cities of the future, users should be equipped with applications able to carry out this selection automatically, and more importantly, to take into account different requirements for a parking space based on user profiles (e.g. business, tourist, generic) that may have different preferences on parking attributes. Furthermore, in order to help refining the selection process, additional information may be used (that could come from other sources of information), such as unavailability of public transportation at the required time, the necessity to reach different locations once the car has been parked, the possibility to find other attractions in the area, and so on.

Another problem of parking in big cities is the fragmentation of public and private parking providers, each one adopting their own technology to collect occupancy data that, as such, cannot be easily shared among different owners or made accessible by user-friendly applications. In order to provide motorists with smart parking applications, the first step would be to encourage public and private parking providers to share their data and to build smart parking software applications that coordinate individual parking solutions for end users without involving them in the fragmentation of parking owners. At the same time, individual parking owners should be made aware of the benefits of providing such a global parking provision showing them that the coordinated provision of parking solutions still guarantees their individual income and fair competition by better exploiting the parking spaces offered in a city. Furthermore, a coordinated parking system allows to gather information to dynamically change the price of the offered parking spaces according to market-based evaluations based on the flow of user requests and the occupancy of the car parks in a given time interval (e.g., the price could decrease according to the occupancy of the parking, or to the time requested by the user), their geographical location, and so on.

In this context, automated negotiation may address some of these issues by allowing car park owners and users to negotiate over parking space attributes whose values may depend on dynamic information and on users' and car park owners' preferences. Different user profiles may be modeled by using different utility functions to evaluate parking offers. It is assumed that car park owners (that can be both public and private) agree to subscribe to a Coordinated Parking System by making it available a given number of parking spaces managed by a Parking Manager Agent (PM). It is responsible for their coordinated reselling to provide a better distribution of vehicles in the managed car parks. Its objective is to sell parking spaces to make a profit, but also to prevent, as much as possible, motorists to park in the city center, so improving the city life by decreasing the circulation of cars in the city center. Motorists are modeled as User Agents (UAs) interacting with the PM to submit requests for parking spaces specifying their own preferences on where to park, but also trying to pay as little as possible. Automated negotiation between the PM and the UA is used to find a parking space allocation that accommodates their needs up to a certain extent, i.e. by finding an acceptable compromise for the involved negotiators.

The length of the negotiation process could prevent its use in real-world scenarios, so we adopt a flexible negotiation mechanism, proposed in [1], that allows to dynamically set the negotiation duration according to the number of available parking spaces that is known only at the time of a request, and so it cannot statically included in the negotiation mechanism.

2.1 The Negotiation Model

The adopted negotiation mechanism, reported in [1], is used in the present work as a bi-lateral negotiation whose protocol is based on a Contract Net Iterated Protocol, and it may be iterated for a variable number of times until a deadline is reached or the negotiation is successful. Each iteration is referred to as a negotiation *round*, and the deadline is the number of allowed rounds.

According to the protocol, at the first negotiation round the UA submits its request for a parking space specifying the preferred location area in the city center, and the requested time interval. The PM replies sending an offer for a parking space, waiting for an acceptance or rejection from the UA. If the offer is accepted the negotiation ends successfully, otherwise a new round is started, if allowed by the protocol. The PM will send as many offers as the number of allowed rounds, that of course cannot be greater than the number of available parking spaces.

In the proposed negotiation, utility functions are used to model the different needs of the PM and the UA: the PM uses the value of the utility functions to decide which offer to send, while the UA uses the utility function to evaluate whether accept or reject the received offer. The utility U for an agent x is a function that depends on the specific agent x, and on an offer o_i made by the agent y (with $x = y$ or $x \neq y$) such as $U_x(o_y) : D_1 \times \cdots \times D_r \rightarrow [0, 1]$, where $D_1, \ldots, D_r$ are the value domains of the r negotiation issues. The issues for the PM are the car park availability and its distance from the city center, while the issues for the UA are the parking space price, the distance of the car park from the requested location, and the same distance evaluated in terms of travel time from the requested location. So, the utility functions for the PM and the UA have the following domains:

$$U_{PM}(offer_{PM}(k)) : \textit{availability} \times \textit{distance\_from\_city\_center} \rightarrow [0, 1]$$
$$U_{UA}(offer_{PM}(k)) : \textit{price} \times \textit{GPS\_distance} \times \textit{time\_distance} \rightarrow [0, 1]$$

where, the co-domain $[0, 1]$ indicates that the functions are normalized.

Utility functions are modeled as linear functions (as will be explained in the following Sections) resulting from the weighted sum of the considered issues. Different *weights* can be associated to the considered parking attributes, so modeling the different importance of the attributes for different classes of users, and even

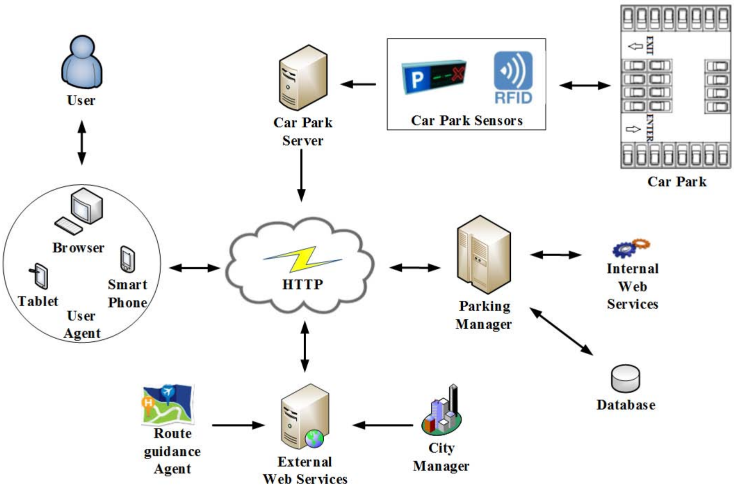

Fig. 1. Coordinated Car Parking Service architecture

for different Parking Managers. It should be noted that an offer proposed by the PM in a negotiation round cannot be considered available in the successive rounds once rejected by the UA, since it may be allocated to a different user, or its price may change according to the number of requests.

3 The Coordinated Parking System

In order to provide motorists with an automatic parking system, first of all it is necessary to provide them with logistic information about available car parks in a specific area, upon a user request. It is assumed that motorists interact with a Coordinated Parking System, as shown in Figure 1, by submitting a request for a parking space to the Car Park Server through several devices (e.g. Tablet, Smart-Phone, PDA or PC) using a city map to select the area where he/she would like to park, and an interface to indicate his/her parking preferences. The PM is responsible for processing the request: it queries an internal database (Database) to retrieve information on the available car parks, and it relies on specific applications to extract car park availability when the request is processed, and to collect relevant information on city regulations, or on events that may affect public transportation.

Each car park is characterized by the following parameters:

car_park= <park_id, park_GPS_location, ref_price_unit,
park_capacity, sector>

where park_id is the unique identifier of the car park, park_GPS_location is its GPS location, ref_price_unit is the default time unit price for a parking space, park_capacity is the total number of parking spaces in the car park, and

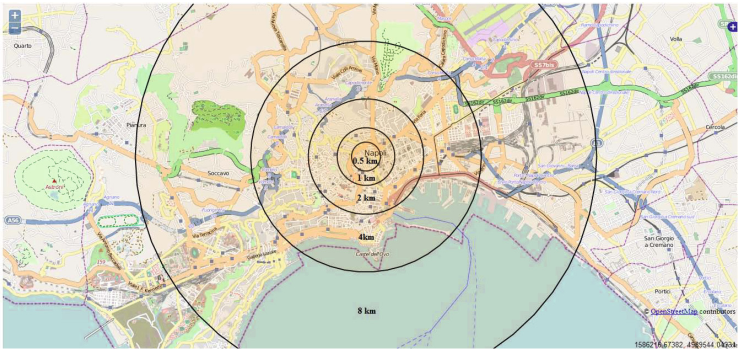

Fig. 2. A representation of city sectors

`sector` represents the geographical location of the car park with respect to the city center. A `sector` identifies a ring and its value is an integer computed as follows:

$$sector = \begin{cases} 0 & distance\_from\_city\_center \quad < min\_range \\ 1 + \lfloor log_2(distance\_from\_city\_center/min\_range) \rfloor & otherwise \end{cases}$$

where min_range is the radius of the area of (`sector=0`), and $distance\_from\_city$ represents the distance between the car park location and the city center (located in the area of `sector=0`).

The distribution of sectors starting from the city center is shown in Fig. 2 and it is used to model the reliance of the price offered for a parking space on the distance between the car park and the city center (as it will be shown in Section 3.1).

A UA request `park_req` is composed of values referred to the parking space attributes that are relevant for the user to decide where to park:

```
park_req= <id_req, dest_GPS_location, start_time,
end_time, reserv_time>
```

where `id_req` is the unique identifier of the user request, `dest_location` represents the GPS location of the destination the user wants to reach, the interval (`end_time - start_time`) represents the time the user wants to park for, and `reserv_time` is a flag used to distinguish between on-demand or advance requests. For the time being, only advance requests are considered since for on-demand requests different assumption on the evaluation of car park occupancy should be considered.

With a static selection, the PM will select car parks considering only to meet the user requirements in terms of location, and available parking spaces for the required time interval. If there is no parking space meeting the requirements, a static mechanism will end up with no solutions for the driver request.

A dynamic selection of parking spaces implies the evaluation of criteria that may not be explicitly expressed by the user, and that can influence both the selection of parking spaces offered by the PM, and the evaluation of the received offer. By using an automated negotiation mechanism for a dynamic selection of parking spaces, it is possible to propose offers that do not strictly meet the user requirements, but that are a result of an evaluation of the available parking spaces against parking space attributes that are relevant to the PM, and whose values may depend on dynamic information, such as the car park occupancy. On the other hand, a received offer is evaluated by the UA against parking space attributes that are relevant to the UA and whose importance (i.e. the weight associated to each attribute) may vary for different users.

3.1 The Parking Manager Model

As described earlier, the proposed negotiation mechanism is not based on the exchange of offers and counteroffers, since UA may only accept or reject offers. So, the PM may compute the set of offers it will propose during negotiation, at the first round. The set of possible offers is computed by selecting first a set of car parks that meet the following requirements:

- the distance (referred to as `park_GPS_distance`) of the car park location (`park_GPS_location`) from the destination (`dest_GPS_location`) set by the user, is within a given distance (referred to as the `location_tolerance`),
- the car park have spaces available for the time interval specified by the user at the time `t` the request is issued.

The `location_tolerance` is set by the PM in such a way to include also car parks that are not in the city center, and consequently they may be far from the `dest_GPS_location` specified by the user, since the PM tries to prevent users from parking in the city center and to maximize the occupancy of car parks not located in the city center. In order to incentivize users to park outside the city center and in car parks with more parking spaces available, the PM calculates the unit price to offer for a parking space by considering that car parks located in the city center are more expensive (according to the distribution reported in Figure 1), and by applying a discount factor that depends on the car park occupancy at the time the request is processed, related to the its total capacity. Hence, the `park_price_unit` for a selected car park is dynamically computed as follows:

$$\texttt{park\_price\_unit} = \texttt{max\_price} - 2^{\texttt{sector}} \cdot (u_p/2) + \left(1 - \frac{\texttt{park\_availability}}{\texttt{park\_capacity}}\right) \cdot u_p$$

where, `max_price` is the maximum time unit price among the selected car parks, `park_availability` is the number of parking spaces available for the time interval (`end_time - start_time`) requested by the UA, `park_capacity` is the total number of parking spaces, and u_p is a unit of price (e.g., 1 euro). It is assumed that `park_availability` is retrieved through a specific service invoked by the PM at the time the request is processed. The PM includes in the offer also the time necessary to travel from `park_GPS_location` to the `dest_GPS_location` by using public transportation (`dest_time_distance`). It is assumed that this information is retrieved with the support of external services. So, an offer of the PM is:

$$\texttt{offer(k)} = < \texttt{park\_id}, \texttt{park\_GPS\_distance}, \texttt{dest\_time\_distance}, \texttt{park\_price\_unit} >$$

Once the PM computes the set of possible offers, it needs to establish which one to offer at each negotiation round, i.e. it needs to establish its concession strategy during negotiation. In order to do so, the PM uses a private utility function to rank the selected car parks. The evaluation function used by the PM to compute the utility of each car parking ($offer_{PM}(k)$) is the following:

$$U_{PM}(offer_{PM}(k)) = \sum_{i=1}^{n}(\alpha_i * \frac{q_{i,k} - \min_j(q_{i,j})}{\max_j(q_{i,j}) - \min_j(q_{i,j})}) \quad (1)$$

where n is the number of issues the agent is evaluating, $q_{i,k}$ is the value of the i-th issue of the k-th car park, $\min_j(q_{i,j})$ and $\max_j(q_{i,j})$ are respectively the minimum and the maximum values of the i-th issue among all the car parks selected by the PM, and the constants α_i are weights associates to the different issues with $\sum_{i=1}^{n}\alpha_i = 1$. As previously described, the issues for the PM are:

$q_1 = dist(\texttt{park\_GPS\_location}, \texttt{center\_GPS\_location})$
$q_2 = \texttt{park\_availability}$

Once the set of offers is ordered according to the utility values of Eq. 1, the PM sends as first offer the one with the highest utility value, and it concedes in utility offering, at each negotiation round, parking spaces with a monotonically decreasing value of its own utility. The PM will end the negotiation with a failure if all the car parks selected have been offered and not accepted. If an offer is accepted by the UA, then the negotiation ends successfully.

3.2 The User Agent Model

The evaluation function used by the UA to compute the utility of each offer proposed by PM is the following:

$$U_{UA}(offer_{PM}(k)) = \left[1 - \sum_{i=1}^{m}\beta_i * \frac{q_{i,k} - c_i}{h_i - c_i}\right] \quad (2)$$

where, m is the number of issues the agent is evaluating, $q_{i,k}$ the value i-th issue of the k-th offer, c_i is the preferred value over the i-th issue, h_i are constant values introduced for normalizing each term of the formula into the set [0,1], and β_i are weights associates to the different issues with $\sum_{i=1}^{m}\beta_i = 1$. Moreover, we assume that the preferred c_i values are reasonable with respect to each considered issue, i. e. the preferred user values are not unreasonable in relation to the issue (this means that the user cannot ask for a parking space for free!). If $q_{i,k} - c_i < 0$ than the term is set to zero.

As previously described, the issues for the UA are:

$q_1 =$ `park_price_unit`
$q_2 =$ `park_GPS_distance`
$q_3 =$ `park_time_distance`

At each round, the UA calculates its utility for the received offer according to Eq. 2, and it accepts it only if the utility value is greater then a predefined threshold. Otherwise, it rejects the offer and waits for another offer, or for a message of negotiation end.

4 Experimental Analysis

A preliminary set of experiments was carried out in order to determine whether the negotiation is a viable approach in order to meet both users and parking managers requirements.

The experiments simulate 150 different queries made by users by selecting a destination on the interactive map of the city provided by the Coordinated Parking System, and associating to the destination the time interval which the user wants to park for. The destinations selected by the users are located in sectors two and three on the city map. For each query a negotiation run takes place. At the first negotiation round, the PM selects the car parks according to the query as reported in Section 3.1. Parking identifiers and locations are extracted from the OpenStreetMap database [2] of the city of Naples (Italy), while routing information (`dest_GPS_distance` and `dest_time_distance`) are retrieved through the use of Google MAPs API [6]. The occupancy of car parks is randomly generated for each negotiation run.

The weights in the utility formulas are equally distributed among issues ($\alpha_i = 0.5$ and $\beta_i = 0.33$ for all i), while, for each issue i, h_i and c_i are dynamically set respectively to $max_j(q_{i,j})$ and $med_j(q_{i,j})$ (i.e., the maximum and the medium value for the current issue). The UA accepts an offer if its utility for that offer is greater than a threshold value set to 0.6 for the experiments.

4.1 Experimental Results

The first experimental results are summarized in Table 1 in case of successful negotiations. In particular, the table reports the maximum, the minimum and the mean value (with the standard deviation), obtained at the end of each negotiation run, of the number of selected car parks (# available parks), the number of negotiation rounds (# Rounds), the parking spaces available in the car park (Availability), the distance between the selected car park and the city center (Distance), the distance between the selected car park and the user's destination (Route), the parking space unit price (Price), the travel time to reach the destination from the car park (Time), the PM utility (PM Utility), and the UA utility (UA Utility).

The mean value of rounds (that is the number of offers sent by the PM) is very low with respect to the mean number of car parks selected by PM for the

Table 1. Experimental Data collected in 150 runs

	max_value	**min_value**	**mean_value**
# Available parks	14	10	11 ± 2
# Rounds	9	1	3.3 ± 2.5
Availability	237	1	110 ± 58
Distance (m)	7339	1948	3495 ± 360
Route (m)	4355	649	1105 ± 160
Price (u_p)	8.9	5.1	7.6 ± 0.3
Time (s)	3046	457	927 ± 211
PM Utility	0.97	0.03	0.62 ± 0.22
UA Utility	0.75	0.10	0.68 ± 0.06
PM Utility without Neg			0.35 ± 0.27
UA Utility without Neg			0.71 ± 0.04

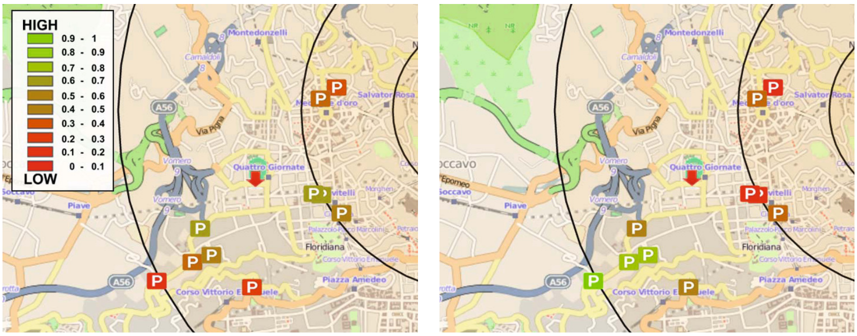

Fig. 3. User Agent and Parking Manager Utilities

experiments. This means that the negotiation ends before the PM offers all the selected car parks, and the obtained mean utilities values for the UA and PM show that the requirements of both parties can be met in a satisfactory way.

With the same settings we evaluated the PM and the UA mean value utilities obtained in the case the complete set of offers selected by the PM is known to the UA as well (the last two rows in Table 1), as shown in Figure 3 that reports a graphical representation of the different utility values respectively for the PM and the UA on the interactive city map. In this case the UA would select the offer that maximizes its own utility (in the average 0.71), that corresponds to a low utility for the PM (in the average 0.35). As expected, in this way, the UA requirement are privileged with respect to the PM ones.

In Table 2 experimental results are reported for two negotiation runs with the same query, but varying the occupancy of the selected car parks. The Table reports the values of the issues of each offer for both the PM and UA and their utilities. According to the negotiation mechanism, at each negotiation round the PM selects the offer with the best utility value, among the remaining offers.

Table 2. Negotiation on a single query

# Rounds	ID	Availability	Distance	Price	Route	Time	PM Utility	UA Utility
1°	417856728	78	3530	7,78	1849	1676	0.77	0.19
2°	2204657189	27	4389	5,14	2151	1951	0.65	0.18
3°	1495201878	40	3719	7,30	1442	1110	0.59	0.45
4°	2245281153	87	2357	7,59	1030	720	0.58	0.62
# Rounds	ID	Availability	Distance	Price	Route	Time	PM Utility	UA Utility
1°	2204658556	171	3712	7,46	1126	848	0.72	0.53
2°	2239471042	237	2273	7,99	1263	1013	0.56	0.43
3°	2204657189	2	4389	5,82	2151	1951	0.50	0.11
4°	2204657190	7	3946	7,86	1525	1790	0.41	0.19
5°	1495201878	18	3719	7,52	1442	1110	0.40	0.39
6°	417856728	36	3530	7,92	1849	1676	0.40	0.18
7°	2245281149	138	2434	7,17	883	725	0.39	0.63

The negotiation ends as soon as the UA utility for an offer is greater than its threshold value. As shown in Table 2, varying the occupancy of the selected car parks impacts the length of the negotiation (i.e., the number of rounds necessary to reach an agreement).

5 Conclusions

Parking in populated urban areas is becoming a challenging problem requiring smart technologies in order to assist users in finding parking solutions, so improving the time necessary to find parking spaces. In this way, it is possible to decrease traffic congestion, and to improve the everyday life of city dwellers. In the present work, we investigated the possibility to use software agent negotiation to address the parking problem by taking into account not only motorists preferences regarding parking location, but also parking vendors preferences regarding car park occupancy, and social city benefits by incentivizing to park outside the city center. We use a flexible negotiation mechanism to find parking solutions that represent a compromise among different needs: a user who prefers to park close to the city center, the car park vendors who prefer to sell parking spaces in less occupied car parks, and a city manager who tries to limit the circulation of cars in city centers. At this purpose, a Coordinated Car Park System is proposed in order to provide a coordinated selling of parking spaces belonging to different car parks, managed by a single software entity, the Parking Manager.

We show that an automated negotiation mechanism between the Parking Manager and motorists represented by User Agents, allows to find this compromise, through the use of utility functions for the involved negotiators that manage different needs to be dynamically evaluated, and help users in their decision making process. The first experiments carried out shows that negotiation is a viable and promising approach since a solution is found before all the selected car parks are proposed to users. The second experimental result shows that car parks occupancy have an impact on the length of negotiation and further experiments will be carried out to find the relation between the occupancy percentage and the length of negotiation.

We plan to extend the experimentation by including different User Agents with different utility functions and weights for the issues that negotiate with the Parking Manager, so to show the suitability of multi-agent negotiation to model real-world scenarios.

References

1. Di Napoli, C., Pisa, P., Rossi, S.: Towards a dynamic negotiation mechanism for qos-aware service markets. In: Pérez, J.B., et al. (eds.) Trends in Practical Applications of Agents and Multiagent Systems. AISC, vol. 221, pp. 9–16. Springer International Publishing, Switzerland (2013)
2. Haklay, M., Weber, P.: Openstreetmap: User-generated street maps. IEEE Pervasive Computing 7(4), 12–18 (2008)
3. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C., Wooldridge, M.: Automated negotiation: prospects, methods and challenges. Int. Journal of Group Decision and Negotiation 10(2), 199–215 (2001)
4. Lopes, F., Wooldridge, M., Novais, A.: Negotiation among autonomous computational agents: principles, analysis and challenges. Artificial Intelligence Review 29(1), 1–44 (2008)
5. Nwana, J.S.: Software agents: An overview. Int. Journal of Group Decision and Negotiation 11(3), 205–244 (1996)
6. Pan, B., Crotts, J., Muller, B.: Developing web-based tourist information tools using google map. In: Sigala, M., Mich, L., Murphy, J. (eds.) Information and Communication Technologies in Tourism 2007, pp. 503–512. Springer Vienna (2007)
7. Polycarpou, E., Lambrinos, L., Protopapadakis, E.: Smart parking solutions for urban areas. In: 2013 IEEE 14th International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1–6 (2013)
8. Teodorović, D., Lučić, P.: Intelligent parking systems. European Journal of Operational Research 175(3), 1666–1681 (2006)