International Journal of Geo-Information

MDPI

Article

An Occupancy Simulator for a Smart Parking System: Developmental Design and Experimental Considerations

Germán Martín Mendoza-Silva *, Michael Gould, Raul Montoliu and Joaquín Torres-Sospedra and Joaquín Huerta

Institute of New Imaging Technologies, Universitat Jaume I, Avda. Vicente Sos Baynat S/N, 12071 Castellón, Spain; gould@uji.es (M.G.); montoliu@uji.es (R.M.); jtorres@uji.es (J.T.-S.); huerta@uji.es (J.H.)

* Correspondence: gmendoza@uji.es

Received: 2 April 2019; Accepted: 3 May 2019; Published: 7 May 2019

Abstract: This paper presents the development of a parking occupancy simulator to support a smart parking system. The simulator uses an agent-based approach to model drivers who follow activity plans and who may or may not use the smart parking system. We illustrate how the process of developing our simulator helped in the design and implementation of the smart parking system components. The paper also shows how the simulator was used to study the possible usage of the smart parking system in a university campus, foreseeing (1) support for the smart parking system's overall suitability, (2) reservation guarantee violation problems, and (3) the value of using total traveled distance as a metric for the smart parking evaluation. The experience presented in this paper may prove valuable to teams planning the development of a smart parking system for similar contexts.

Keywords: on-street parking simulators; smart parking systems; agent-based modeling

1. Introduction

Agent-based simulations have proven valuable for studying traffic and mobility phenomena, including parking search and availability, without disrupting the actual traffic in a city. Many cities assign great importance to solutions to parking-related problems [1]. Those solutions include smart parking systems (SPS) which, mainly found as Parking Guidance and Information (PGI) systems, determine the parking occupancy and provide suggestions about parking availability [2]. SPS have been shown to have positive effects on driver parking success and on traffic flow [3]. SPS development has also significantly benefited from behavior models for parking search that help in analyzing the underlying phenomenon [4], and in testing of design and implementation choices [5,6]. Drivers are commonly represented as agents, using Agent-Based Modeling (ABM) [7,8], located in an urban environment. The environment can be represented using data from Geographic Information Systems (GIS), which provide a digital representation of the urban environment [9] and are already in place in many city governments.

When off-the-shelf simulation software is used to build a parking simulation linked to an SPS, the software often can impose data format and scripting restrictions that create conflicts with the SPS design and implementation. These conflicts prevent or complicate the SPS and simulator from sharing GIS data and algorithm implementation [5,9], affecting resource sharing and thus reutilization. Building or adapting a parking simulator so that it can share data and software components with its related SPS can benefit an SPS project beyond what is explored in previous parking simulations studies. Such benefits are important considering the increasing availability of city geospatial data [10] and

the increase in government interest in parking optimization [11]. For example, an SPS development project for a city may benefit from (1) the incorporation of available geospatial services into the SPS development, and (2) the early evaluation of the SPS design and feasibility by means of a parking occupancy simulation.

This paper describes our experiences in building an agent-based parking occupancy simulator. The simulator had two major goals: (1) testing suitability of an SPS in the context of a university campus and (2) reuse of its development efforts (and code) during the SPS development. The SPS targeted by our simulator checks the occupancy of on-street parking spots using sensors, and handles logical spot reservation upon request. The simulator allows the exploration of parking occupancy patterns created by agents that either use or decline to use the SPS. Agents represent drivers of the most typical profiles of people who drive to and within the campus.

As a case study, experimentation was performed using the simulator to explore situations with different levels of parking demand and SPS usage. The results provide insights into a metric for SPS suitability evaluation from a driver's point of view. Also, the experiments allowed exploration of the reservation guarantee problem (someone stealing your assigned spot while you are en route to it), which arises due to the lack of a physical reservation enforcement. In summary, the main experiences and recommendations in this paper are:

1. The methodology we employed to increase re-usability of software development efforts for a parking simulator, applied to a related SPS development;
2. how to explore the reservation guarantee concept for an SPS without physical reservation enforcement; and
3. how to use the total driving distance metric for making credible comparisons when evaluating an SPS usage benefits.

To the best of our knowledge, no previous study has proposed the mentioned reutilization methodology relating an SPS and a parking occupancy simulator. The design proposal allows novel traits like running a simulation from current parking state data, or automatically using the latest environment information. Likewise, despite the fact that the reservation guarantee problem has been acknowledged by other studies, they did not study the problem incidence under several levels of SPS usage. Our analysis questions the acceptability of an SPS that promises a reservation to drivers and does not physically enforce the reservation. Furthermore, the parking studies mainly analyze parking search distance, which is only a part of the total driving distance. Additionally, the code of our simulator is freely available in a public repository.

The remaining sections of this article are as follows. Section 2 presents relevant previous studies and supports our design considerations. Section 3 describes the relationship between the SPS and the parking simulator. Section 4 explains the simulator's details. Section 5 describes the case study of the simulator for the experimental evaluation of SPS usage. Finally, conclusions and acknowledgement sections are presented.

2. Background

This section demonstrates how our design approach is unique by reviewing previous studies. We also review previous SPS evaluation metrics to highlight the relevance of our proposed metrics.

2.1. Smart Parking Systems and Parking Simulations

Modeling and simulation have been used for knowledge discovery applicable to parking systems, as well as for testing already implemented SPS. Some examples of the first approach are: testing a utility function that involves factors affecting parking choice [4], collaborative path-finding in a multi-agent context applied to an SPS [12], testing a parking planning algorithm [13,14], and an SPS evaluation considering several vehicle categories [15]. Examples of the second approach include:

testing an SPS model to explore factors like distance to building entrances [16], testing dynamic prices assignment [5], and parking guidance evaluation [6].

Despite the fact that modeling and simulation techniques are often related to SPS design or evaluation [2,17–21], to the best of our knowledge efforts devoted to the simulator/simulation's development are not reutilized in SPS development. A simulator and its related SPS are generally built following distinct goals: The former's development commonly seeks a fast way to study the parking phenomenon, while the latter's development pays more attention to common software concerns like robustness, efficiency and load handling. During our work, we noticed an interesting opportunity to reuse the simulation software modules in a related SPS by combining and linking the simulator development with the development of the guidance/reservation algorithm and other software components for the SPS. The abundance and variety of available agent-based modeling toolkits [17,22] may facilitate such co-developments. The algorithms or other components required in the SPS may be implemented using the same programming tools in the simulator. In a general sense, doing so may require:

1. building or adapting parking simulator software and not just defining a model to run in available modelers,
2. working alongside the SPS development team, and
3. applying software design techniques that assure robust re-usability.

The team that developed the simulator described in this paper was also part of the team developing the targeted SPS. Its members had software design and team skills that enabled them to meet the previous requirements. The software components reutilized in this work are the parking reservation component and the components for accessing the related data and external services. Section 3 presents the selected design decisions that assured the sought-after component reutilization.

2.2. Agent-Based Parking Models And Gis

ABM applications to traffic and transportation, including parking-related phenomena, are significant and numerous [7,8], with several popular ABM toolkits being spatially explicit [23] or including extensions that provide support for GIS data usage [22]. Agents represent drivers in cars moving across an environment, which is usually composed of a network of road, target destinations, and parking spaces.

In research literature related to parking studies, the bridging of ABM and GIS is addressed either using particular spatial data formats or by having the simulation built within a GIS platform. Works like SUSTAPARK [24], TRANSIMS [25], MATSim [26], PARKGRID [27] and PARKAGENT [13,28,29] are good examples. SUSTAPARK and PARKGRID load the roads and parking data from GIS layers stored in files, e.g., in shapefile format. TRANSIMS and MATSim read their input data—e.g., network, destinations, and activity plans, from files that follow their own specification, though they include some GIS tools for importing, exporting, or visualizing other formats. PARKAGENT was implemented as an ArcGIS© application so as to have direct access to GIS data and services.

Following the reutilization goal presented in Section 2.1, we decided to assure the GIS data and services were accessible online and decoupled from the simulator. This decision allowed reutilization of data access, as well as software components. A GIS server providing data access through web services enabled data sharing between several applications, and more specifically, between the simulator and the SPS, thus allowing interesting new considerations—such as running a simulation from the actual parking state detected by the SPS. Our proposal includes the parking spot information and the car and pedestrian route determination hosted as services in a GIS server. The ability to run a simulation from current parking state data, and to automatically use the latest environment information, which is provided by a GIS server, is a distinctive and novel trait of our methodology proposal.

2.3. SPS Evaluation Metrics

The metrics used for evaluating the benefits of using a Smart Parking System include parking search time [5,30], mean driving distance [31], wandering ratio [5], walking time [32], and travel density and average speed [33]. Measurements (most often distance or time) usually start when a car enters the simulated area [31], when it is near to its destination or makes a request for parking [5], or when it arrives to a parking lot [30]. In our model, agents using the SPS are guided as soon as they enter the walled university campus, a city surrogate. The use of total driving distance to achieve credible comparisons of SPS usage benefits is not recommended, given that drivers travel certain distances to their destination regardless of their SPS usage. The total distance is generally valuable, however, and simple to calculate; therefore, we devised a methodology to use the total driving distance in our experiments. Section 5 shows how we arrived at the methodology through a series of experiments using our simulator.

In our model, the driver's travel distance while trying to park might be affected not only by the driver's parking choice, destination, and parking availability. An agent that does not use our SPS may occupy a parking spot already reserved for another agent, forcing the latter to issue a new spot reservation. The effects of reservation without a physical guarantee, i.e., nothing stopping a driver from grabbing someone's reserved spot, are not explored in the literature we reviewed, though reservation guarantee is a key aspect of SPS [5]. The literature does suggest approaches to enforce the reservation, but they are either relatively expensive (physical barriers) or not fully effective [2,5,32]. Section 5 shows how we explored the reservation guarantee problem, and comments on the negative impact it may have for the SPS' usage. To the best of our knowledge, no previous study has hinted on the level of SPS utilization under which the reservation guarantee problem is the most notorious for a given environment.

3. The SPS and the Parking Simulator

The SPS was created and tested at the campus of the Universitat Jaume I (UJI) in Spain. It is a walled complex and has four vehicle entrances. Its parking spots are free and on-street, with some areas similar to those in a small town neighborhood and other larger areas similar to that near a sporting facility. The SPS has detection sensors, smart parking services, and client applications. The magnetic sensors detect the parking occupancy and deliver that information to the smart parking services through a wireless network. These services, exposed as REST web services, handle occupancy data storage and provide search, reservation, and routing functionalities to a smartphone client application. The application allows visualization of available parking spots, spot reservation, and driver guidance.

Our simulator represents drivers that move across the campus to reach their destinations and park in spots convenient to them. The simulator's design allows the usage of the simulated parking occupancy data as a fake (surrogate) input from the SPS sensors. Therefore, the simulator became a valuable tool for the SPS development team for testing the SPS before deployment of the actual occupancy detection sensors in the university campus. Also, the SPS' parking reservation component was implemented and tested as a part of the simulator. Therefore, any further refinement to it could be easily tested through simulations and later be directly used in the SPS. Figure 1 shows the layered design, after Fowler [34], of our simulator software. The Data layer obtains the necessary information for running the simulation. The Model layer represents the actual model (its implementation is aided by an ABM library) along with the implementation of some parts of the SPS. Finally, the Presentation layer has components that handle the model's output and user interaction. The layers vertically communicate using facades [34].

The GIS data and services were hosted using commercial, off-the-shelf GIS server software (Esri (Esri software company (http://www.esri.com/)) ArcGIS for Server, now ArcGIS Enterprise), which is used by many city governments and it is available to universities via the Esri educational institution license. At UJI, this server already hosted production-ready (properly prepared) GIS data and services regarding the university campus, collected and built for the Smart Campus system [35]. This server

provided a common access point to geographical data for the simulation and the SPS. The relevant hosted GIS services for the simulation and the SPS were: (1) parking spots data, (2) building and campus entrance points data, and (3) car and pedestrian routing services on a previously digitized street network. These services are consumed as REST web services. Under this uncoupled schema, any changes to parking spots, buildings, or routes are immediately available to the SPS, any related application, and the simulator.

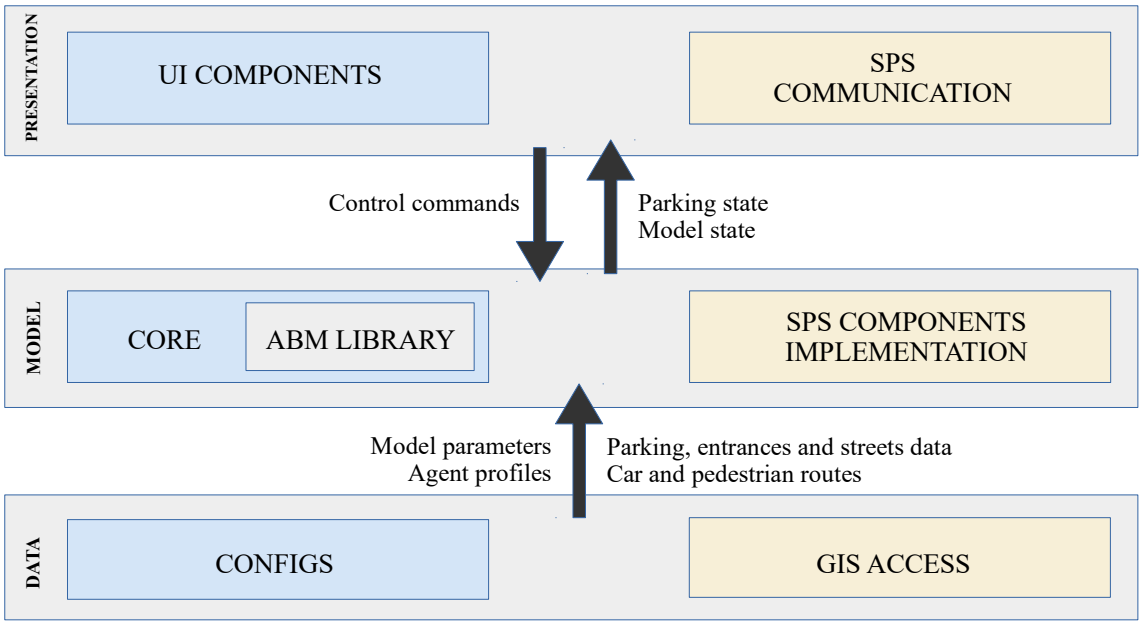

Figure 1. Simulator's layered organization. SPS = smart parking systems; GIS = Geographic Information Systems; ABM = Agent-Based Modeling.

4. Parking Simulator Details

The simulator defines an agent-based model representing the parking occupancy created by drivers within the campus. Its implementation took into account the integration discussed in Section 2.

4.1. The Model

The model represents a 'virtual week' period. For each day of the week, agents arrive to the environment, follow their activity plan, and leave. An agent attempts to park as near to its destination point as possible. Destinations are particular entrance doors of buildings. In studies, such as Geng and Cassandras [5], destinations from the same building are aggregated and considered as one. In our model, given the dispositions of building entrance doors and parking spots, different doors (potentially far apart) of the same building were considered as distinct destinations. The criterion for measuring parking-to-destination proximity is walking distance as measured using actual pedestrian ways (sidewalks and crosswalks).

4.1.1. Agent Profiles

The agent activity plans (agent profiles) define the typical cases of drivers. As an example, consider a student needing to be at a specific classroom at 09:00. At 08:45, she arrives by car to the campus through the campus entrance of her choosing. She then parks near the building door she considers is the best for her destination. After a while, when the class finishes, she walks to her car and drives to the sports complex, which is far from where her car was previously parked. She parks near a door of that complex. After completing her sporting activities she drives out of the campus.

Agents that belong to the same profile have the same type of destinations and similar arrival and departure times. A random, normally distributed variation is allowed around those times. The profiles were created considering the actual number of people from a community profile that commonly visits a building. Agent profiles consider the expected number of people actually driving to the campus for each of those groups. Activity plans are important for transport simulation [36] and the agent profiles addressed in this work resemble those from Horni et al. [26] and Dieussaert et al. [24]. The numbers of people defining each profile and those describing facilities' usage were obtained from the university administration services and community surveys, which is further explained in Section 5. We consider that agent profiles built in this way is a plausible alternative to more common approaches (like car counting at parking spots and at campus entrances) as it requires considerably less effort and infrastructure.

4.1.2. Search Behavior

Some agents ('Guided') rely on the SPS for finding an available parking spot, while other agents ('Explorer') decide for themselves where to park. Studies like Geng and Cassandras [5] have also used these two behaviors, seeking to quantify the benefits from using an SPS. When an agent is created, its type is randomly defined under the restrictions established by a model parameter that controls the proportion between the two types of agents. This model parameter can be dynamically adjusted. Driving behavior and parking search are complex processes [37–39] and several studies have applied realistic behaviors [3,13,19,27], even considering recent trends like driver-less vehicles [40] or specific contexts like a city center [41] and university campuses with specific policies and notable parking supply shortages [42]. We chose two simple parking search behaviors for our model because obtaining an approximate parking occupancy, rather than the most realistic one, was enough for our simulator goal.

Explorer

The 'Explorer' search behavior takes inspiration from Dieussaert et al. [24], Benenson et al. [28], Levy et al. [29], Martens et al. [43], with some simplifications and additions. For example, variations in car speed or maximum search time are not considered, while variable agents' maximum walking distances and two measures for evaluating local parking availability are considered. An 'Explorer' agent first tries to park as close as possible to its destination. If it fails, the agent then tries to park in the first available parking spot it can find. Algorithm 1 presents pseudo-code which briefly describes the Explorer parking search behavior.

Agents move following the shortest network path to their destinations. An agent starts searching for parking when it is within a maximum walking distance to its destination. An agent's maximum walking distance is a random value ($maxWalkDist$) within a range. The agents can detect only parking spaces within a visibility distance, which is defined by a model parameter. While searching, an agent records the proportion of free-to-total parking spots it detects. The agent decides to park in an available spot when (1) the current proportion falls below a critical ratio ($criticalRatio$); or (2) the difference between the proportion from the previous step and the current proportion is greater than a critical value ($criticalReduc$). Both ($criticalRatio$) and ($criticalReduc$) are model parameters.

If an agent has completed the route to its destination without being able to park, it tries to park in the first available parking spot it detects searching first around the target building and then around other buildings. If it does not find available parking places around any building, the agent returns to a campus' exit and leaves.

Algorithm 1: Explorer agents decision rules

Input: $destination, maxWalkDist, criticalRatio, criticalReduc$

```
found ← false
compute shortest path to destination from current point
while not at the end of path and found = false do
  if distance to destination ≤ maxWalkDist then
    pastratio ← ratio
    ratio ← local available parking spots / local total parking spots
    diff ← ratio − pastratio
    if ratio > 0 and (ratio < criticalRatio or diff > criticalReduc) then
      choose closest reachable parking spot and move to it
      found ← true
    end if
  end if
  compute next point in path to move to
end while
if found = true then
  park()
else
  while more destinations to explore and found = false do
    destination ← next entrance of same building or another building
    compute shortest path to destination from current point
    while not at the end of path and found = false do
      get local available parking spots
      if there is available spots then
        choose closest reachable parking spot and move to it
        found ← true
      end if
      compute next point in path to move to
    end while
  end while
  if found = true then
    park()
  else
    leave the campus
  end if
end if
```

Guided

An agent with a 'Guided' behavior requests a parking spot reservation from the SPS's reservation component, which chooses the best available spot for its destination. It then follows the optimal route to the spot and occupies it. The reserved spot will not be offered to any other agent until the occupying agent explicitly releases it. As the reservation process is logical, not physical, before a 'Guided' agent arrives to its reserved parking spot, an 'Explorer' agent might find that spot and occupy it. When the 'Guided' agent notices that situation, it asks for a new parking spot and heads for it.

4.2. The Simulator

The chosen ABM library for model formalization and implementation was MASON [44], which facilitated the process of creating an integrated simulation. We additionally used the GeoMASON extension [45], which incorporates support for vector and raster geospatial data. The final implementation followed the software design shown in Figure 1.

The *Data* layer loads the simulation's configuration (including agent profiles) from files, reads the GIS data—which is principally campus cartography—from the GIS Server, and reads parking place status from the SPS. The GIS data for parking spots and entrances are loaded at simulation start, but the routing services are used on-demand. The parking place status information are also loaded at simulation start, thus enabling us to simulate the parking occupancy from a known starting point or from an SPS-provided parking occupancy.

Figure 2 provides an overall, simplified view of the *Model* layer design, which includes model notifications, agents' behavior, model configuration and smart parking artifacts implementation. Any component interested in receiving notifications of model changes must subscribe to the appropriate updater. Distinct agent behaviors are achieved through the basic parking agent specialization. The SPS's reservation component, though it is known and used by the simulator, is implemented independently from the simulation core, thus allowing for easy substitution. By implementing model controllers it is possible to set up the necessary relations, e.g., updaters, according to the target application platform.

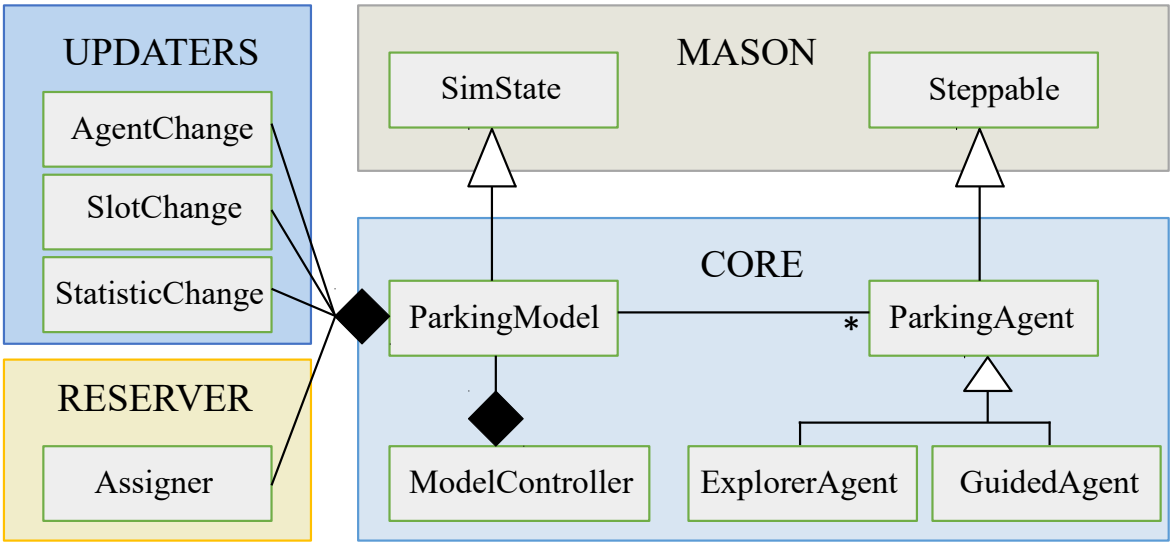

Figure 2. Main classes of the Model layer and their relation.

The *Presentation* layer has components that are notified when the model changes. These components inform the SPS about those changes (Smart Parking Communication) or show them (UI Components) in applications that wrap the simulator. The Smart Parking Communication component delivers occupancy state changes. Three implementations of the simulator's UI Components have resulted in three distinct applications: (1) A desktop version that uses MASON visualization utilities (Figure 3) and shows the parking availability and the agent moving across the campus, (2) a console version for simplified and faster runs, and (3) a web-hosted version. Having three different interfaces for different purposes highlights the flexibility of the approach used for building the simulator. A video showing a simulation run is available online (Demonstration Video: https://youtu.be/gZj21WtOmio).

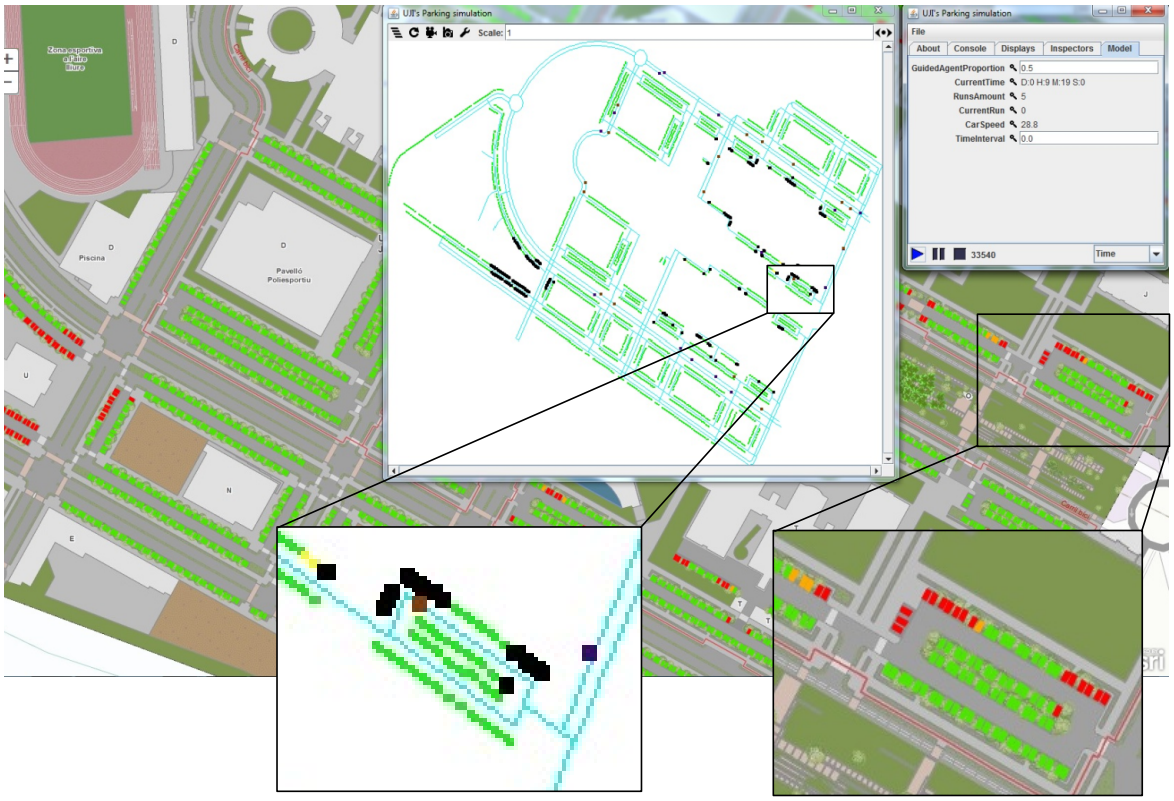

Figure 3. One of the simulator applications running in the foreground, while in the background a web map shows the parking spots' current status as stored in the SPS. In the simulator, black squares represent parked agents. In the web map, red squares represent occupied parking spots.

4.3. Parking Reservation Component

The parking reservation component serves reservation requests in the order they are received. Several studies have proposed advanced reservation algorithms that take into account, e.g., current driver travel time, parking pricing, and reservation updates when the parking availability changes [2,21,32]. As the reservation algorithm was not among this study's main goals, the reservation component uses a simple heuristic that only takes into account the parking spots availability and the walking distance from the spots to the specified destinations. In experimental measurements, network communication lags for car and pedestrian routing requests had medians of 0.43 and 0.08 s, respectively. Car route requests are only made when a car starts moving, thus they do not add a significant burden if made on-demand. An agent's parking reservation request requires determination of the available parking spot that is the closest (according to walking distance) to an agent's destination. To avoid on-demand pedestrian routing requests associated with parking reservation requests, the parking reservation component calculates beforehand, then sorts and stores the distances from every parking spot to every destination. This approach is feasible because the number of destinations (buildings' doors) and parking spots is 225 and 3809, respectively. Therefore, the time for finding the closest available spot only depends on the total number of parking spots.

4.4. Final Development Considerations

The simulator development presented in the previous sections also requires additional minor design and development decisions (All code is shared in a public repository: https://goo.gl/GmWyt1), such as development platform or execution environment. Furthermore, although the usage of resulting simulator software artifacts is straightforward, some additional steps are required if an SPS development is initiated from those artifacts.

The choice of the development platform (and programming language) depends mainly on preferences from the development team and additional design considerations (e.g., related to the SPS design). As can be inferred from Kravari and Bassiliades [17], Crooks et al. [22], it is possible to accommodate model specification (using an ABM library) to the chosen development platform. The usage of an ABM library for model implementation implies coding driver behaviors following the library specifications. For example, using MASON (which is Java-based), driving behavior is coded in methods from a class. An agent is created as an object from that class, and a method from it is executed by a discrete event scheduler. In the case of models considering very complex driving behaviors, the required coding effort may be significant. For those models, a benefit trade-off between re-usability and coding effort should be considered.

The development platform may also influence the way remote services are consumed. The REST architectural style is widely used and its support spans most development platforms. The communication with those services is eased if SDKs exist for the targeted GIS server. For example, the software company Esri provides several runtime SDKs for client applications to access services published by Esri ArcGIS Enterprise, as well as for map visualization and geometrical/geodesic operations. Other SDKs alternative sources could be Mapbox [46] or Geotools [47]. For models that cover large and congested areas (which simultaneously include several thousands of drivers), a large number of network requests are issued, which could potentially overload the GIS server especially if that server already supports multiple users for other purposes. In such situations, a different design (routing operations computed in the simulation) may be preferable.

For the integration of the simulator core components into an application, already mentioned in Section 4.2 for the web or desktop versions, most of the effort is devoted to creating a proper presentation/usage of simulation output and controlling the simulation execution. For example, the web version wraps the simulator core components with REST web services. A thin web client uses them to obtain the simulation state (occupancy) and to display it on a map, and to control the simulation (starting/stopping the simulation and parameters configuration).

Once the simulator is built and tested, the following elements are ready for reuse in an SPS:

1. GIS data and services hosted in a GIS server,
2. software artifacts for remote GIS data read/write operations,
3. reservation component, and
4. visualization components created for simulation testing.

The combination of the enumerated elements with an occupancy detection system (e.g., magnetic sensors and its communication components) and a user (mobile) application can produce a relatively simple, yet functional SPS. As an example, the occupancy or availability state storage, the computation of vehicles routing indications and the maps for visualization are provided by elements in (1). Elements from (2) can be reused to create a web application that provides a gateway for occupancy state discovery and update. The update operations are used by the occupancy detection system. The discovery operations are used by the client application, which also uses elements from (1), and by a web dashboard to monitor the SPS state, which can also reuse elements from (4). The reservation component (3) is vital for the SPS and can be readily used in the SPS as it is isolated from the simulation.

5. Case Study: Exploration of SPS Expected Usage

Several experiments were performed to demonstrate the simulator's potential use and to explore likely benefits of SPS usage. The model parameters' values are presented in Table 1.

Table 1. Model parameters values.

maxWalkDist(m)	[50 to 100]
criticalRatio	0.25
criticalReduc	0.15
visDistance(m)	40
carSpeed(km/h)	30

Model parameters *maxWalkDist*, *criticalRatio*, and *criticalReduc* were already explained in Section 3. Parameter *visDistance* indicates the distance for which 'Explorer' agents consider their detected parking occupancy for deciding whether to park or not. Parameter *carSpeed* sets agents' movement velocity. Agent profile values are not presented here because of their large number of details. The simulator includes other relevant parameters that are not model parameters, for example, the GIS data and routing services, and whether to initialize the simulation with the current SPS occupancy state.

To define destinations, times, and amounts of distinct profiles (groups) of people coming to the campus by car, a set of root behaviors was created based on actual university data for a typical academic month. The root behaviors act as templates used to automatically create agent profiles, which in turn are used as templates for creating agents during a simulation run. The root behaviors used in the experiments considered the main distinct groups of the university members. They defined tasks in which the first drivers arrived to the campus around 08:00 and the last ones leaved the campus around 20:00. The root behaviors included a main task, which was going to a building, and subsequent optional tasks. Optional tasks were added to agent profiles randomly, complying with the expected facilities usage. The optional tasks were going to a sport facility, to the library, or to a distant building, if the optional task's destination was not already the main task's destination. We obtained the typical quantities of each group accessing each facility and their typical staying times, although for profile creation a group-dependent random variation was applied to staying times. Through driver surveys, we also estimated the likelihood for each group to use a car inside the campus. By creating agent profiles automatically from root profiles, we achieved a rich set of over 300 profiles, which is important in transport simulation [36].

Each experiment consisted of 15 simulation runs with the same parameters. Each experiment was run for a specific proportion p between 'Explorer' and 'Guided'. It also used a profile set created using the root behaviors and considering a specific proportion c of people actually using their car on campus. The maximum amount (N) of people expected to come to the campus on a typical day is about 15,000. Each root behavior (people profile) b accounts for some part N_b of that amount of people. For an experiment run using 0.1 as value for c, the number of agents created for root behavior b is $0.1N_b$, and the expected total amount of agents is (about) 1500. Parameters p and c allow exploration of situations with different levels of SPS usage and parking demand, respectively. The measured total driving distance of each agent considered all stretches of its multi-destination journey. With $D(p,c)$ denoting the set of all traveled distance measurements for a particular experiment, then:

$$\begin{aligned} D(p,c) &= D_G(p,c) \cup D_E(p,c) \\ p &\in P = \{20\%, 50\%, 80\%\} \\ c &\in C = \{0.1, 0.2, ..., 1.0\}, \end{aligned} \tag{1}$$

where $D_G(p,c)$ and $D_E(p,c)$ denote subsets that contain measurements only from 'Guided' agents or from 'Explorer' agents, respectively. We denote the mean values of previous sets as $\overline{D(p,c)}$, $\overline{D_G(p,c)}$, and $\overline{D_E(p,c)}$.

Figure 4 presents the case for $\overline{D(p,c)}$. It shows that the SPS usage should reduce the parking searching time. There is a drop in mean value for 'Guided' agents from proportions 0.1 to 0.2. This may

be due to the fact that with proportion 0.1, only a few agents, or no agents at all, have as destinations buildings with a low number of people and located near campus entrances.

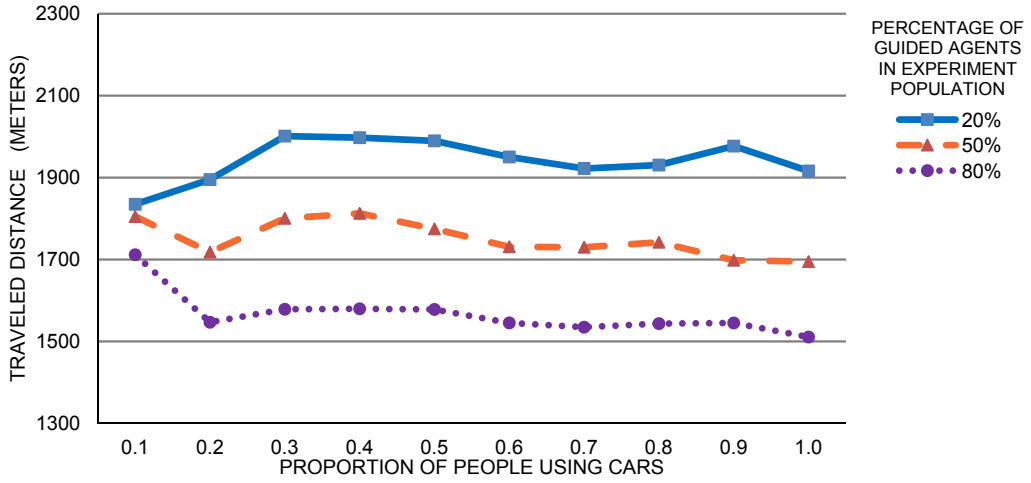

Figure 4. Variation of the mean traveled distance.

The mean traveled distance remained stable in spite of the growing number of drivers. The parking reservation component reserves parking spots that are uniformly distanced from a building target door, causing their mean distance to remain stable for 'Guided' agents. For 'Explorer' agents, we have identified two likely reasons: (1) Parking demand is spread over the day and across several building doors, and (2) when some drivers arrive to campus (most likely after midday), others have already left. Furthermore, parking demand and supply are not heterogeneous across the campus; in a similar way they differ in other scenarios [27], leading to divergences from the expected increase in the parking search time when the parking demand increases. The relation between parking search time and parking demand is not trivial [48].

Figure 5 presents another way to use the mean traveled distance for meaningful comparisons. Each data point $d(p, c)$ of the series is calculated as:

$$d(p, c) = \overline{D_E(p, c)} - \overline{D_G(p, c)}. \quad (2)$$

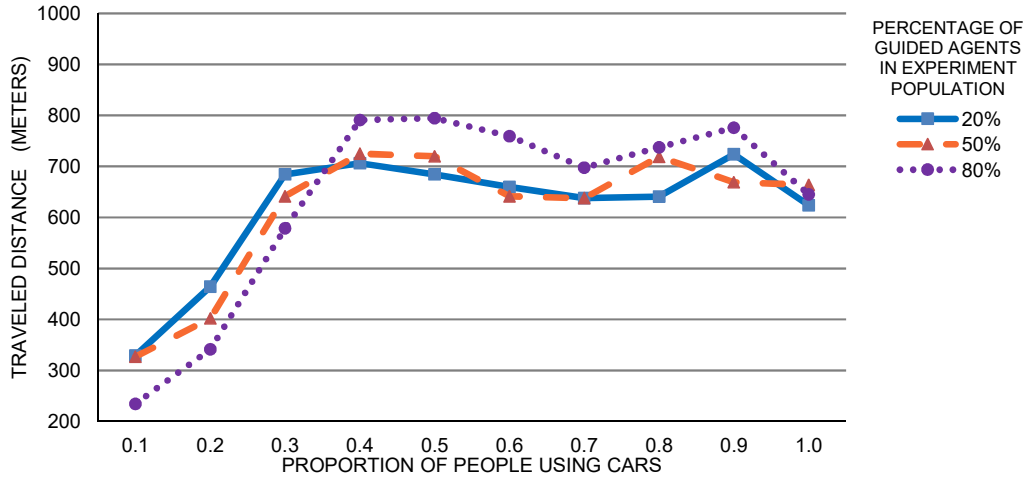

Figure 5. Variation of the mean traveled distance, as expressed by the difference between the mean distance of 'Explorer' and 'Guided' populations.

Figure 5 indicates that the 'Explorer' agents have longer driving distances. The difference stabilizes and stays around 700 m, which may hint at the minimum expected difference in the mean traveled distance in the campus scenario. An unsuccessful parking search across the parking spaces that surround a campus' building would add a value to the accumulated search distance of an agent that, on average, is above 600 m.

When considering all measurements from all experiments, the mean value was 1737 m and the standard deviation was 1174, approximately. The two metrics explored so far in this subsection ($\overline{D_G(p,c)} and \overline{D_E(p,c)}$ and $d(p,c)$) are affected by the distribution of buildings (and their doors), parking spots, and campus' entrances. To avoid that issue, instead of using the mean value, we considered using percentiles values. First, two sets are defined as follows:

$$T_E(p) = \bigcup_{c \in C} \{D_E(p,c), p \in P\} \tag{3}$$
$$T_G(p) = \bigcup_{c \in C} \{D_G(p,c), p \in P\}.$$

Let $n_E(i,p)$ and $n_G(i,p)$ denote the *ith* percentiles of $T_E(p)$ and $T_G(p)$, respectively. The data points of Figure 6 are those percentiles values, considering three levels of usage of the SPS. The chart allows comparisons of distance categories, and it suggests that 'Explorer' agents are strongly affected by high parking demand situations, regardless of the usage level of the SPS. In addition, to compare different levels of usage of the SPS within the same chart, we defined new data points as follows:

$$d(i,p) = n_E(i,p) - n_G(i,p), p \in P. \tag{4}$$

Figure 7 shows the data points $d(i,p)$. The metric value of each proportion $p \in P$ for a percentile are very similar to each other for percentiles below 40, but it significantly differs for the percentiles at about 80. The difference for the highest percentiles is higher for 80% of SPS usage, which is also noticeable in Figure 6. The reason for the higher difference in high SPS usage situations is that 'Guided' agents 'discover' and thus take the remaining spots faster than 'Explorer' agents.

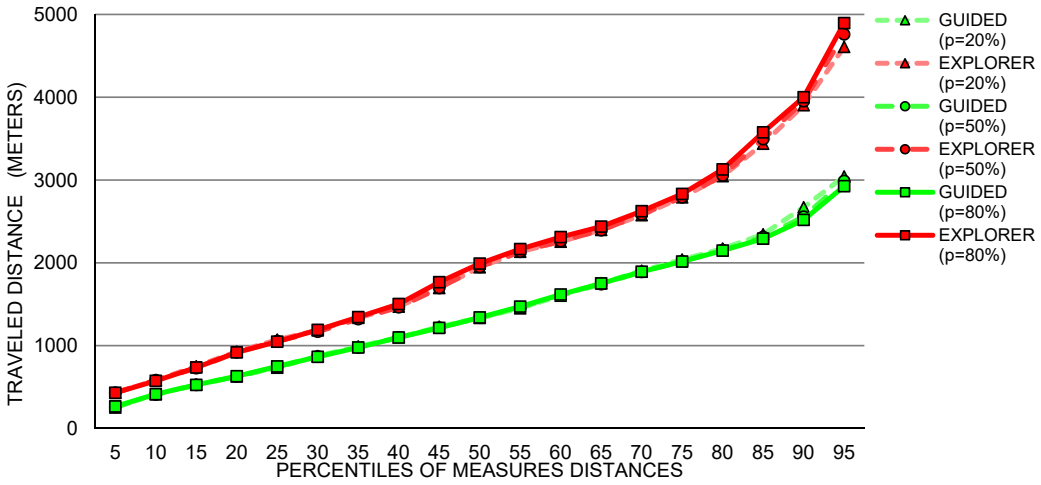

Figure 6. Difference in traveled distance between 'Explorer' and 'Guided' agents, as seen using percentiles and considering p = 20%, p = 50%, and p = 80%.

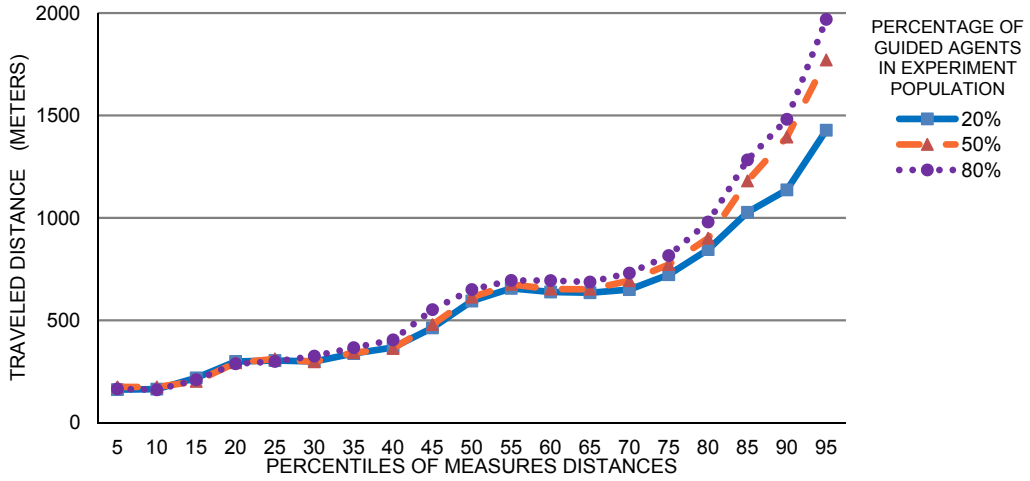

Figure 7. Difference in traveled distance, as expressed by the difference of same percentile values of the 'Explorer' and 'Guided' populations.

Figure 8 shows that the occurrence of the reservation guarantee problem grows as the competition for parking spaces increases. It is most notable when the 'Guided' to 'Explorer' proportion value is 50%. The guarantee problem is more notable for proportion of 75% than 25% because in the former case there are more agents whose reserved spaces are susceptible to be 'stolen'. Drivers declining to use the real SPS may be an issue for those who use it. When a driver has its reserved spot 'stolen' by another driver, the former needs to keep driving to locate an available spot. Depending on the parking demand, such additional search may take a long time. Furthermore, regardless of the amount of time devoted to the additional search, the reservation violation creates discomfort for the driver, which may lead to a decline in SPS usage. The 'reservation' term in an SPS that does not make physical reservation enforcement could be misleading for a driver, and 'suggestion' term should be used instead. When a parking spot is 'stolen' and given the occupancy detection capability of the SPS, a driver that was suggested to take that spot should be alerted and provided with an alternative spot and a new route indication to it. To avoid alternatives far away from the original suggestion, the latter could be determined taken into account the availability of other parking spots close to it, an idea that resembles heuristics applied in PGI systems [2].

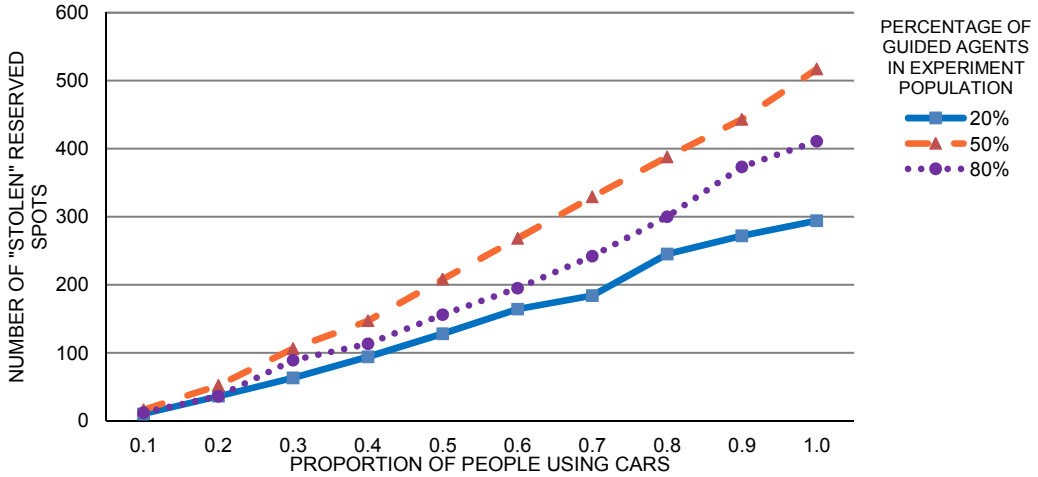

Figure 8. The reservation guarantee problem, explored for several levels of parking demand and agent types proportions.

Single Building Analysis

To study a specific high parking demand situation, e.g., in the case of a notorious event hosted in a campus facility, we ran simulations for which agents tried to park near a randomly chosen entrance of a particular building during a 'virtual' day period. We made them arrive at the campus at similar times. The parking demand significantly exceeded the parking supply around the building, and the agents had to park around other buildings. As presented in Figure 9, in the case of the 'Explorer' agents, the worst result corresponds to the proportion of 'Guided' agents $p = 20\%$, which is a result not only from competition, but also from the fact that the 'Explorer' agents that try to park near the same building's entrance follow similar routes while searching for parking, which makes them go through the least favorable paths. In the case of the 'Guided' agents, the worst results correspond to the proportion $p = 50\%$, which corresponds to when they are the most affected by the reservation guarantee problem.

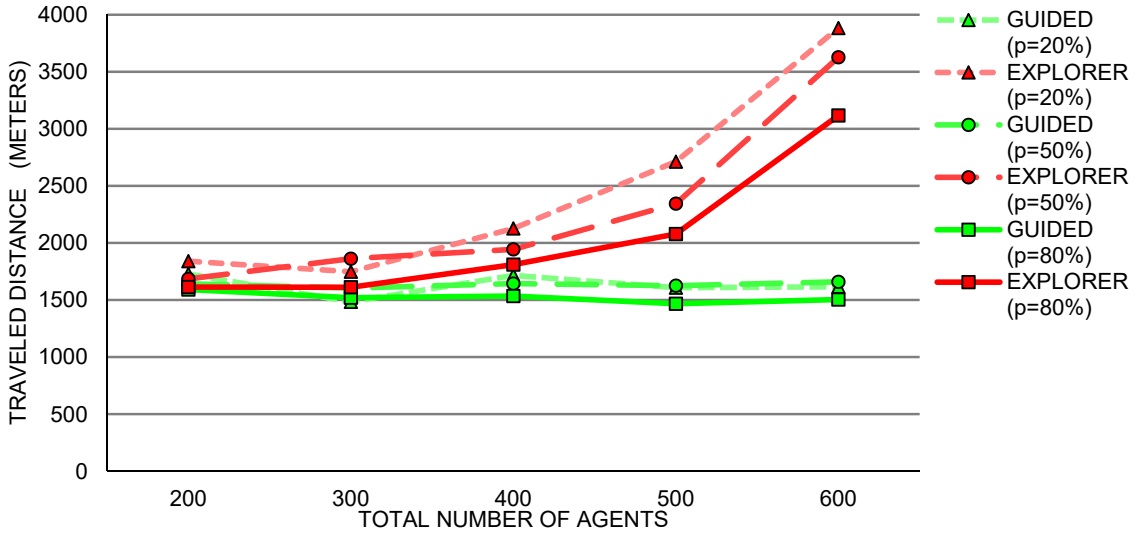

Figure 9. Traveled distance when considering only one campus building.

The correspondence between larger traveled distance and higher parking demand for the 'Explorer' agents is in line with the results presented in previous studies [27]. Likewise, the steady behavior for 'Guided' agents corresponds to already reported benefits of parking reservation systems [2]. However, the comparison between the three studied levels of SPS usage for 'Explorer' agents reveals insights different to those obtained when considering the simulations campus-wide during an entire week. The difference between the two study cases is an indication that the traveled distance does not only depend on the demand and level of usage of the SPS, but also on the scenario. The campus traveled distance analysis results from this work should be applied with caution to other distinct scenarios. The campus has a large amount of parking spots, while in a different scenario, e.g., a city center, the number of parking spots may be rather small. Depending on the scenario, the analysis might have to be performed on small extents. The reservation guarantee problem is, however, inherent to on-street parking, and it should be expected to be the most significant when the numbers of people using and refusing to use an SPS are similar.

A simple driver model and a simple parking reservation heuristic were used in this work to show that an SPS development team could be able to assume the simulation development without significant effort. Having agents with parking search behaviors more advanced than those used in this work would decrease the traveled distance numbers in the campus scenario, e.g., using learning to avoid places that are usually crowded. However, we consider that more general results, like the 'Guided' agents having lower traveled distances than the 'Explorer' agents, and the reservation problem being

the most significant when the number of the two agent types are similar, should remain true even when a new a parking search behavior is considered.

6. Conclusions

This paper presented practical considerations from the development of a parking occupancy simulator. The simulator helped in assessing suitability of an SPS suitability for a university campus, and its design and development contributed to reduce the SPS development efforts. The simulator shares software components, GIS data, and services with the SPS. The paper commented on design decisions made regarding the SPS and the parking occupancy simulator, which may be used in similar contexts to minimize development efforts. The application of the proposed methodology on the simulator development will produce software components readily usable in an SPS, as well as GIS data and services readily available online when published on a GIS sever. The software created for this work is available in a public repository, for further technical details inspection and to encourage reproducibility. The paper also presented experimental evaluation of SPS usage benefits using the simulator. The experiments differentiated profiles of agents that use the SPS and those who decline to use it. Analysis of experimental results showed how to use the total driving distance as a metric for evaluating the SPS benefits from a driver point of view. The experimental results also allowed us to explore the effects of having a parking reservation that is logical but not physical, clearly showing that the drivers who decline to use the SPS may "steal" a significant number of already reserved parking spot, which in our experiments reached numbers higher than 100 for a parking demands above the 40 percent of the maximum demand. The guarantee problem becomes more significant as the level of usage of the SPS increases, topping out at 50%, which may hamper drivers' acceptability towards an SPS.

The parking choices used in this work are simple, as they were not among its main goals. Future improvement directions could include the incorporation of more realistic choices, driving behavior, and the ability to learn from experience, which may be combined with an improved reservation algorithm in order to provide agents with updates upon changes in the occupancy state, e.g., when other alternative spots become available. Additionally, a comprehensive model validation with a study case for a different scenario and parking occupancy measurements taken in the field is planned as future work. The design and development guides and experimental analysis presented in this paper show the convenience of including simulations when considering the application of an SPS to a scenario, while diminishing possible dissuasive aspects like model complexity or simulation development efforts, that SPS development teams may consider.

Author Contributions: Conceptualization: Germán M. Mendoza-Silva and Raul Montoliu; Methodology: Germán M. Mendoza-Silva, Michael Gould and Joaquín Torres-Sospedra; Investigation: Germán M. Mendoza-Silva; Software, Germán M. Mendoza-Silva; Supervision: Raul Montoliu andMichael Gould; Resources, Joaquín Torres-Sospedra and Joaquín Huerta; Writing—original draft, Germán M. Mendoza-Silva and Michael Gould; Writing—review & editing, Michael Gould, Joaquín Torres-Sospedra and Joaquín Huerta

Funding: We thank funding from the Spanish' Ministerio de Economía y Competitividad under the project 'SmartWays' (Convocatoria Retos-Colaboración, RTC-2014-1466-4). Germán M. Mendoza-Silva gratefully acknowledges funding from grant PREDOC/2016/55 by Universitat Jaume I.

Acknowledgments: We thank our university's Management Department for providing campus usage and community information.

Conflicts of Interest: No potential conflict of interest was reported by the authors. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Inci, E. A review of the economics of parking. *Econ. Transp.* **2015**, *4*, 50–63. [CrossRef]
2. Kotb, A.O.; Shen, Y.C.; Huang, Y. Smart Parking Guidance, Monitoring and Reservations: A Review. *IEEE Intell. Transp. Syst. Mag.* **2017**, *9*, 6–16. [CrossRef]

3. Małecki, K. A computer simulation of traffic flow with on-street parking and drivers' behaviour based on cellular automata and a multi-agent system. *J. Comput. Sci.* **2018**, *28*, 32–42. [CrossRef]
4. Waraich, R.A.; Axhausen, K.W. Agent-based parking choice model. *Transp. Res. Rec.* **2012**, *2319*, 39–46. [CrossRef]
5. Geng, Y.; Cassandras, C.G. New 'smart parking' system based on resource allocation and reservations. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1129–1139. [CrossRef]
6. Shin, J.H.; Jun, H.B. A study on smart parking guidance algorithm. *Transp. Res. Part C Emerg. Technol.* **2014**, *44*, 299–317. [CrossRef]
7. Bazzan, A.L.C.; Klügl, F. A review on agent-based technology for traffic and transportation. *Knowl. Eng. Rev.* **2014**, *29*, 375–403. [CrossRef]
8. Maggi, E.; Vallino, E. Understanding urban mobility and the impact of public policies: The role of the agent-based models. *Res. Transp. Econ.* **2016**, *55*, 50–59. [CrossRef]
9. Crooks, A.T.; Castle, C.J.E. The integration of agent-based modelling and geographical information for geospatial simulation. In *Agent-Based Models of Geographical Systems*; Springer: Dordrecht, The Netherlands, 2012; pp. 219–251.
10. Dangermond, J. Geospatial Technology and the Future of the City, 2015. Available online: http://www.esri.com/esri-news/arcnews/winter1415articles/geospatial-technology-and-the-future-of-the-city (accessed on 23 April 2019).
11. Frost&Sullivan. Smart Parking to Enable Intelligent Mobility in Global Mega Cities, 2015. Available online: http://ww2.frost.com/news/press-releases/smart-parking-enable-intelligent-mobility-global-mega-cities/ (accessed on 23 April 2019).
12. Rhodes, C.; Blewitt, W.; Sharp, C.; Ushaw, G.; Morgan, G. Smart Routing: A Novel Application of Collaborative Path-Finding to Smart Parking Systems. In Proceedings of the 2014 IEEE 16th Conference on Business Informatics, Geneva, Switzerland, 4–17 July 2014; pp. 119–126.
13. Levy, N.; Render, M.; Benenson, I. Spatially explicit modeling of parking search as a tool for urban parking facilities and policy assessment. *Transp. Policy* **2015**, *39*, 9–20. [CrossRef]
14. Mei, Z.; Feng, C.; Ding, W.; Zhang, L.; Wang, D. Better lucky than rich? Comparative analysis of parking reservation and parking charge. *Transp. Policy* **2019**, *75*, 47–56. [CrossRef]
15. Leclercq, L.; Sénécat, A.; Mariotte, G. Dynamic macroscopic simulation of on-street parking search: A trip-based approach. *Transp. Res. Part B Methodol.* **2017**, *101*, 268–282. [CrossRef]
16. Leephakpreeda, T. Car-parking guidance with fuzzy knowledge-based decision making. *Build. Environ.* **2007**, *42*, 803–809. [CrossRef]
17. Kravari, K.; Bassiliades, N. A survey of agent platforms. *J. Artif. Soc. Soc. Simul.* **2015**, *18*, 11. [CrossRef]
18. Boudali, I.; Ouada, M.B. Smart Parking Reservation System Based on Distributed Multicriteria Approach. *Appl. Artif. Intell.* **2017**, *31*, 518–537. [CrossRef]
19. Ni, X.Y.; Sun, D.J. Agent-Based Modelling and Simulation to Assess the Impact of Parking Reservation System. *J. Adv. Transp.* **2017**, *2017*, 2576094. [CrossRef]
20. Chen, Z.; Spana, S.; Yin, Y.; Du, Y. An Advanced Parking Navigation System for Downtown Parking. *Netw. Spat. Econ.* **2019**. doi:10.1007/s11067-019-9443-4. [CrossRef]
21. Lin, T.; Rivano, H.; Le Mouël, F. A survey of smart parking solutions. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3229–3253. [CrossRef]
22. Crooks, A.; Malleson, N.; Manley, E.; Heppenstall, A. *Agent-Based Modelling and Geographical Information Systems: A Practical Primer*; SAGE Publications Limited: London, UK, 2018.
23. Taillandier, P.; Gaudou, B.; Grignard, A.; Huynh, Q.N.; Marilleau, N.; Caillou, P.; Philippon, D.; Drogoul, A. Building, composing and experimenting complex spatial models with the GAMA platform. *GeoInformatica* **2018**. doi:10.1007/s10707-018-00339-6. [CrossRef]
24. Dieussaert, K.; Aerts, K.; Steenberghen, T.; Maerivoet, S.; Spitaels, K. SUSTAPARK: An agent-based model for simulating parking search. In Proceedings of the AGILE International Conference on Geographic Information Science, Hannover, Germany, 2–5 June 2009.
25. Lee, K.S.; Eom, J.K.; seop Moon, D. Applications of TRANSIMS in Transportation: A Literature Review. *Procedia Comput. Sci.* **2014**, *32*, 769–773. [CrossRef]
26. Horni, A.; Nagel, K.; Axhausen, K.W. *The Multi-Agent Transport Simulation MATSim*; Ubiquity Press: London, UK, 2016.

27. Fulman, N.; Benenson, I. Agent-Based Modeling for Transportation Planning: A Method for Estimating Parking Search Time Based on Demand and Supply. *arXiv* **2018**, arXiv:1806.10874.
28. Benenson, I.; Martens, K.; Birfir, S. PARKAGENT: An agent-based model of parking in the city. *Comput. Environ. Urban Syst.* **2008**, *32*, 431–439. [CrossRef]
29. Levy, N.; Martens, K.; Benenson, I. Exploring cruising using agent-based and analytical models of parking. *Transp. A Transp. Sci.* **2013**, *9*, 773–797. [CrossRef]
30. Surpris, G.; Liu, D.; Vincenzi, D. How Much Can a Smart Parking System Save You? *Ergon. Des. Q. Hum. Factors Appl.* **2014**, *22*, 15–20. [CrossRef]
31. Wang, H.; He, W. A Reservation-based Smart Parking System. In Proceedings of the 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Shanghai, China, 10–15 April 2011; pp. 690–695.
32. Tasseron, G.; Martens, K. Urban parking space reservation through bottom-up information provision: An agent-based analysis. *Comput. Environ. Urban Syst.* **2017**, *64*, 30–41. [CrossRef]
33. Cao, J.; Menendez, M. System dynamics of urban traffic based on its parking-related-states. *Transp. Res. Part B Methodol.* **2015**, *81*, 718–736. [CrossRef]
34. Fowler, M. *Patterns of Enterprise Application Architecture*; Addison-Wesley Longman Publishing Co., Inc.: Reading, MA, USA, 2002.
35. Torres-Sospedra, J.; Avariento, J.; Rambla, D.; Montoliu, R.; Casteleyn, S.; Benedito-Bordonau, M.; Gould, M.; Huerta, J. Enhancing integrated indoor/outdoor mobility in a smart campus. *Int. J. Geogr. Inf. Sci.* **2015**, *29*, 1955–1968. [CrossRef]
36. Baqueri, S.F.A.; Adnan, M.; Kochan, B.; Bellemans, T. Activity-based model for medium-sized cities considering external activity–travel: Enhancing FEATHERS framework. *Future Gener. Comput. Syst.* **2019**, *96*, 51–63. [CrossRef]
37. Chaniotakis, E.; Pel, A.J. Drivers' parking location choice under uncertain parking availability and search times: A stated preference experiment. *Transp. Res. Part A Policy Pract.* **2015**, *82*, 228–239. [CrossRef]
38. Zhao, C.; Li, S.; Wang, W.; Li, X.; Du, Y. Advanced Parking Space Management Strategy Design: An Agent-Based Simulation Optimization Approach. *Transp. Res. Rec.* **2018**. [CrossRef]
39. Antolín, G.; Ibeas, Á.; Alonso, B.; dell'Olio, L. Modelling parking behaviour considering users heterogeneities. *Transp. Policy* **2018**, *67*, 23–30. [CrossRef]
40. Bischoff, J.; Maciejewski, M.; Schlenther, T.; Nagel, K. Autonomous vehicles and their impact on parking search. *IEEE Intell. Transp. Syst. Mag.* **2018**. [CrossRef]
41. Khaliq, A.; van der Waerden, P.; Janssens, D.; Wets, G. A Conceptual Framework for Forecasting Car Driver's On-Street Parking Decisions. *Transp. Res. Procedia* **2019**, *37*, 131–138. [CrossRef]
42. Meng, F.; Du, Y.; Li, Y.C.; Wong, S.C. Modeling heterogeneous parking choice behavior on university campuses. *Transp. Plan. Technol.* **2018**, *41*, 154–169. [CrossRef]
43. Martens, K.; Benenson, I.; Levy, N. The dilemma of on-street parking policy: Exploring cruising for parking using an agent-based model. In *Geospatial Analysis and Modelling of Urban Structure and Dynamics*; Springer: Berlin, Germany, 2010; pp. 121–138.
44. Luke, S. Multiagent simulation and the MASON library, 2015. Available online: https://cs.gmu.edu/~eclab/projects/mason/manual.pdf (accessed on 23 April 2019).
45. Coletti, M. The GeoMason Cookbook, 2013. Available online: https://cs.gmu.edu/~eclab/projects/mason/extensions/geomason/geomason.pdf (accessed on 23 April 2019).
46. Kastanakis, B. *Mapbox Cookbook*; Packt Publishing Ltd.: Birmingham, UK, 2016.
47. GeoTools: The Open Source Java GIS Toolkit, 2019. Available online: https://geotools.org/ (accessed on 23 April 2019).
48. Arnott, R.; Williams, P. Cruising for parking around a circle. *Transp. Res. Part B Methodol.* **2017**, *104*, 357–375. [CrossRef]