Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

DISCRETE EVENT SIMULATION OF SMART PARKING CONFLICT MANAGEMENT

Antoine Dominici
Laurent Capocchi
Emmanuelle De Gentili
Jean-François Santucci

University of Corsica
SPE UMR CNRS 6134, Campus Grimaldi
20250 Corte (France)

Abstract

Smart parking is a framework aimed to optimize the occupancy of parking spots based on specifications that include the behavior of drivers. One of the challenges in this area concerns the determination of a reliable model able to resolve cumulative parking conflicts that appear when many drivers look for parking in a dynamic environment system where user behavior is paramount. This abstract presents a discrete-event modeling and simulation approach dedicated to propose conflict management strategies based on the estimated travel time to reach desired places around a specific area.

1 Introduction

More and more cities are choosing to use Smart Parking software tools to solve their parking problems. Indeed, because of the inflation of land prices in city centers, adding new parking lots addresses a lot of problems and offers cities very few solutions. This is why the solution of providing information to users has become so important because it allows, with much less financial means, to ensure that the various car parks already in place can be used to their full potential (Lin, Rivano, and Le Mouël 2017). The smart parking approach presented in this paper is a complex discrete-event system whose components can be described by finite state automata reacting to internal or external events. The DEVS (Discrete-Event system Specification) formalism (Zeigler et al. 2018) makes it possible to model this system and to simulate it both in real time and in simulated time.

2 Proposed Approach

During our previous research (Dominici et al. 2020), a system combining discrete-event simulation and artificial intelligence to determine the time at which a place will be released from its user has been developed. To do this we have classified the different classes according to the estimated time before their release. In this paper, based on our previous work, we want to create a system to direct a driver looking for a place according to the release times of the different places available to him. We must also take into account the competition between drivers wanting to park so that they do not interfere with each other and therefore do not increase the time to find a space due to conflicts.

To prove that such an approach is possible and interesting to achieve with a discrete-event system, we propose to simulate the evolution of class-based sensors previously constructed in our previous research (Dominici et al. 2020). Then we add a system allowing users to simulate finding a place while being in conflict with other users, all this while applying different conflict management policies in order to determine which would be the most suitable for a real situation.

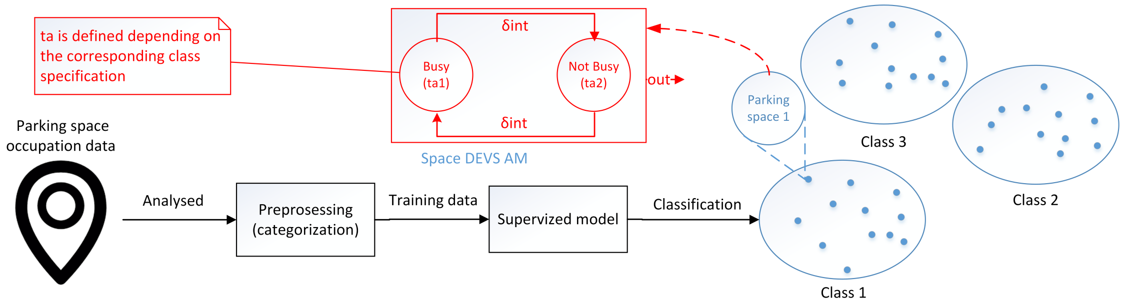

Figure 1: Classification sensor system with the new atomic model ”space”.

3 Modeling and Simulation of Smart Parking Conflicts

Figure 1 shows an overview of the approach used in our previous paper (Dominici et al. 2020) to classify parking spaces as well as an extension illustrated in red that will be presented in this paper. A “Space” atomic model can be constructed with states whose lifetime depending on the previous defined classes and representing a sensor placed on the ground detecting the presence of a car.

Each ”Space” atomic models is associated with a class before to start the simulation. This parking slot will change state at regular intervals depending on its class which is modeled using the DEVS time advance function. In our current application, a driver has the ability to search for one (or a set of) place(s) (inside an area) in order to maximize the chances of finding one of them available. In order to simulate the users, we have to create an environment in which a driver can move thanks to a model called ”Travel”. We also need to place the sensors on this virtual environment. Once this is done, each of the instantiated users will therefore move according to one or another strategy to one place. However these can come into conflict by coveting the same place. We have created an atomic model ”Access Conflict Management” which has (according to different policies) to manage potential conflicts occurring during the simulation.

We have therefore done a significant amount of simulation of the evolution of parking spaces for a dynamic and random environment. The simulations were performed with different policies. We noticed that the policy can be less and less effective depending on a environment which doesn’t fit with it. A minimum distance policy will then be more interesting when used in the short term, while a policy taking into account the places chosen and avoiding redundancy and inter-blocking is much more interesting in the long term.

Our results allowed us to see that an application of the simulation in the previous case presented advantages at all points. However the applied policies being too simplistic it does not allow to solve all the problem related to the parking in the cities. In the future, we will therefore have to rely on more advanced methods such as reinforcement learning in order to obtain much more effective policy. We will also try to apply these methods to real cases in the Smart-Parking application which will soon be available in the city.

REFERENCES

Dominici, A., L. Capocchi, E. De Gentili, and J.-F. Santucci. 2020. “Towards a combination of discrete-event simulation with machine learning for smart-parking”. In *Proceedings of the 2020 Summer Simulation Conference*, 1–12.

Lin, T., H. Rivano, and F. Le Mouël. 2017. “A survey of smart parking solutions”. *IEEE Transactions on Intelligent Transportation Systems* 18(12):3229–3253.

Zeigler, B. P., A. Muzy, and E. Kofman. 2018. *Theory of modeling and simulation: discrete event & iterative system computational foundations*. Academic press.