3-31-2021

CASSI: Designing a Simulation Environment for Vehicle Relocation in Carsharing

Christoph Prinz

University of Goettingen, christoph.prinz@uni-goettingen.de

Mathias Willnat

University of Goettingen, mathias.willnat@uni-goettingen.de

Alfred Benedikt Brendel

University of Goettingen and Technische Universität Dresden, Alfred_benedikt.brendel@tu-dresden.de

Sascha Lichtenberg

Technische Universität Dresden, sascha.lichtenberg@tu-dresden.de

Lutz Kolbe

Georg-August-University of Göttingen, lutz.kolbe@wiwi.uni-goettingen.de

Follow this and additional works at: <https://aisel.aisnet.org/thci>

Recommended Citation

Prinz, C., Willnat, M., Brendel, A. B., Lichtenberg, S., & Kolbe, L. (2021). CASSI: Designing a Simulation Environment for Vehicle Relocation in Carsharing. *AIS Transactions on Human-Computer Interaction*, 13(1), 12-37. <https://doi.org/10.17705/1thci.00140>
DOI: 10.17705/1thci.00140

This material is brought to you by the AIS Journals at AIS Electronic Library (AISeL). It has been accepted for inclusion in AIS Transactions on Human-Computer Interaction by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

3-2021

CASSI: Designing a Simulation Environment for Vehicle Relocation in Carsharing

Christoph Prinz

University of Göttingen, christoph.prinz@uni-goettingen.de

Mathias Willnat

University of Göttingen, mathias.willnat@uni-goettingen.de

Alfred Benedikt Brendel

Technische Universität Dresden, alfred_benedikt.brendel@tu-dresden.de

Sascha Lichtenberg

Technische Universität Dresden, sascha.lichtenberg@tu-dresden.de

Lutz Kolbe

University of Göttingen, lutz.kolbe@wiwi.uni-goettingen.de

Follow this and additional works at: <http://aisel.aisnet.org/thci/>

Recommended Citation

Prinz, C., Willnat, M., Brendel, A. B., Lichtenberg, S., & Kolbe, L. M. (2021). CASSI: Design of a simulation environment for vehicle relocation in carsharing. *AIS Transactions on Human-Computer Interaction*, 13(1), pp. 12-37.

DOI: 10.17705/1thci.00140

Available at <http://aisel.aisnet.org/thci/vol13/iss1/2>

CASSI: Designing a Simulation Environment for Vehicle Relocation in Carsharing

Christoph Prinz

University of Göttingen
christoph.prinz@uni-goettingen.de

Mathias Willnat

University of Göttingen
mathias.willnat@uni-goettingen.de

Alfred Benedikt Brendel

Technische Universität Dresden
alfred_benedikt.brendel@tu-dresden.de

Sascha Lichtenberg

Technische Universität Dresden
sascha.lichtenberg@tu-dresden.de

Lutz Kolbe

University of Göttingen
lutz.kolbe@wiwi.uni-goettingen.de

Abstract:

Simulations offer an efficient solution to represent operational services and track the impact of changing systematic factors and business constraints. Carsharing services provide users with mobility services on demand. Although research has introduced strategies to optimize efforts to set up and operate such a system, they lack reusable and flexible simulation environments. For instance, carsharing research applies simulations to better understand and solve the problem of balancing vehicle supply and demand, which operators need to solve to prevent operational inefficiencies and ensure customer satisfaction. Hence, one cannot feasibly test new balancing mechanisms directly in a real-world environment. As for now, researchers have implemented simulations from scratch, which results in high development efforts and a limited ability to compare results. In this paper, we address this gap by designing a versatile carsharing simulation tool that researchers can easily use and adapt. The tool simplifies the process of modeling a carsharing system and developing operation strategies. Furthermore, we propose various system performance measures to increase the developed solutions' comparability.

Keywords: Carsharing, Discrete Event Simulation, Vehicle Relocation, Design Science Research.

Alan Hevner was the accepting senior editor for this paper.

1 Introduction

Embedding data analytics and process simulation into research and organizations has become more critical to drive value creation (Lavalle, Lesser, Shockley, Hopkins, & Kruschwitz, 2011). Simulations in particular provide a cost-efficient solution to comprehensively represent operational services and to track the impact that systemic factors and business constraints have on system behavior when such factors and constraints change (Alfian, Rhee, Ijaz, Syafrudin, & Fitriyani, 2017). In this context, carsharing shows that having detailed knowledge about system behavior, which allows an organization to derive implications due to changes in business operations, represents a significant competitive advantage and a fundamental requirement to create successful business models (Clemente, Fanti, Mangini, & Ukovich, 2013b; Jorge & Correia, 2013; Nijland, van Meerkerk, & Joen, 2015).

Carsharing services provide individuals with cars from a fleet on a pay-per-use basis and constitute low-emission, smart, flexible, and dynamic mobility services to complement current public transportation infrastructure (Shaheen, Cohen, & Chung, 2008; Firnkorn & Müller, 2011; Martin & Shaheen, 2011; Baptista, Melo, & Rolim, 2014; Becker, Ciari, & Axhausen, 2018). On the one hand, these services must be convenient to attract a large user base in order to help solve future mobility problems; on the other hand, they must generate profit to help the mobility industry transform (Schiller, Scheidl, & Pottebaum, 2017). Against this background, one major problem that carsharing services face concerns balancing vehicle supply and demand. Asynchronous demands lead to an imbalance of vehicles in the operation area, which reduces accessibility and, thereby, attractiveness for customers and profitability for the industry. In order to provide solutions to this problem, research has begun to develop methods to determine and schedule vehicle relocations in complex simulation systems (Brendel, Brauer, & Hildebrandt, 2016; Herrmann, Schulte, & Voß, 2014; Jorge, Correia, & Barnhart, 2014).

Researchers have typically developed simulations to solve the relocation problem from scratch and adapted them to specific research context and showcases (Čertický, Jacob, Pibil, & Moler, 2014). Hence, researchers need extensive programming skills to either reuse existing simulations by adapting them or developing a simulation tailored to their carsharing system configuration. Overall, this requirement constitutes a high entrance barrier for interested scholars and also leads to high costs and working effort, which slows down research progress on how to solve this highly relevant challenge. Furthermore, researchers cannot easily compare operation strategies' impact and practicality because they have simulated them in vastly different ways and because no uniform performance measures for carsharing systems exist. In order to address issues that affect current simulation solutions, we create a versatile carsharing simulation tool with uniform performance measures that one can easily use and adapt. In particular, we address the following research question (RQ):

RQ: How should one design an accessible carsharing simulation platform to allow researchers to develop and evaluate carsharing relocation strategies?

This paper proceeds as follows: in Section, we review the current literature on car sharing simulations. In Section 3, we describe the research approach and procedures we followed. In Section 4, we present our results and generalize them to create a design theory. In Section 5, we discuss the study's limitations and possible starting points for future research. Finally, in Section 6, we conclude the paper.

2 Related Work

Carsharing systems typically operate in three forms: 1) station-based two-way carsharing, 2) station-based one-way carsharing, and 3) free-floating carsharing (Brendel et al., 2016). Station-based two-way carsharing makes cars available at designated stations. Whenever a customer picks a car from a particular station for a short-time rental, the customer needs to return it to the same station (Balac, Becker, Ciari, & Axhausen, 2018; Di Febbraro, Sacco, & Saeednia, 2018). Station-based one-way carsharing extends this approach enabling customers to return rented vehicles to any available station in the system. In free-floating carsharing, customers can pick up any available vehicles from a fleet distributed in an operation area (Shaheen, Sperling, & Wagner, 1997; Di Febbraro, Sacco, & Saeednia, 2013; Jorge & Correia, 2013; Boyaci, Zografos, & Geroliminis, 2014; Degirmenci & Breitner, 2014). The rental process completes when the customer parks the car inside a defined area and features pay-per-use pricing based on time and/or distance travelled (Hildebrandt, Hanelt, Piccinini, Kolbe, & Niero-Bisch, 2015; Martin & Shaheen, 2011).

Shaheen et al. (1997) and Zoepf and Keith (2016) conclude that users evaluate a carsharing service's attractiveness mainly based on vehicle ability at reservation time in a short distance. Free-floating carsharing services provide this flexibility, but asynchronous vehicle demand and supply leads to imbalance in the system. In practice, vehicles tend to cumulate in hot spots, which leads to a lack of vehicles in cold spots and, therefore, decreases the service's attractiveness (Boysen, Briskorn, & Schwerdfeger, 2019; Weikl & Bogenberger, 2013). Furthermore, introducing electric vehicles into carsharing systems add more constraints such as the need to optimize charging strategies (Brendel, Lichtenberg, Brauer, Nastjuk, & Kolbe, 2018b).

2.1 Strategies to Simulate System Operations

In general, operational researchers have widely used simulation modeling to approximate real-world systems' behavior and address decision-making complexity in systems (Jahangirian, Eldabi, Naseer, Stergioulas, & Young, 2010). In this context, researchers have applied three main simulation techniques: 1) discrete event simulation (DES), 2) system dynamics (SD), and 3) agent-based simulation (ABS) (Maidstone, 2012).

First, DES simulates a system's behavior based on a sequence of discrete events (Fishman, 1978). Entities in the system follow processes and change their state while time passes (e.g., a carsharing transaction could comprise a reservation request, vehicle pick-up, and vehicle drop-off). A sequence of events could contain multiple user requests that the simulation logic satisfies as soon as they have been scheduled.

Second, SD simulates a system's behavior based on the idea that all objects inside a system interact with each other, which means that a change in one variable affects the other variables over time. SD simulations define the flows between entities in a system and views problems from a macroscopic perspective (Sharp & Price, 1984). For example, customer demand influences the number of parked vehicles at a carsharing station. A DS simulation defines the spatiotemporal flow between stations and builds the system's foundational behavior.

Third, ABS simulates a system's behavior by replicating its objects' microscopic behavior. The system comprises autonomous agents that follow rules to achieve their objects. Furthermore, they can interact with each other and the environment (Axhausen, Nagel, & Horni, 2016). For example, electric vehicles inside a free-floating carsharing system constitute agents that can satisfy demand when they have sufficient battery power and other users have not blocked them. When users pick up these cars for a trip, the cars maintain their state (e.g., position, battery level) and contribute to the system behavior from the bottom up.

Combining two or more of these simulation approaches has grown in popularity in operation research since most real-world problems are complex and depend on many different constraints and characteristics, which one single method can rarely address adequately (Brailsford, Eldabi, Kunc, Mustafee, & Osorio, 2019). To find an optimal system operation solution, simulations help one compare alternating strategies' efficiency and effectiveness, which means that they do not lead automatically to an optimal solution. The operation logic of SD simulations can be formulated as a set of parameterized equations. One can derive the optimal parameter set for such mixed-integer programming problems using CPLEX solvers, heuristics, or implementations such as the branch-and-cut approach. However, understanding, modeling, and applying such equations requires deep stochastic knowledge, which makes the entry into carsharing relocation research more difficult for scholars. To address scholars' requirements on accessibility, one should consider DES- and ABS-based simulation processes in designing a simulation framework supporting the development and evaluation of relocation strategies. Still, the support of mixed integer programming or stochastic programming should be offered to more advanced users.

2.2 Information Systems Instantiated in Carsharing Relocation Research

Research has developed information systems (IS) that help optimize carsharing models' system parameters (e.g., fleet size, station position) (Rickenberg, Gebhardt, & Breitner, 2013) and help carsharing services operate (e.g., relocations, adaptive pricing) (Brendel, Brennecke, & Kolbe, 2018a). According to the literature reviews that Cepolina, Farina, and Pratelli (2014), Illgen and Höck (2019), and Jorge and Correia (2013) conducted, relocation research commonly focus on improving a carsharing system's

performance measures by finding decision variables that enhance the system's capability to serve customers (e.g., leading to more accepted rental requests) and, subsequently, generate profits.

In this context, researchers have tried to determine an ideal initial system setup (Barth & Todd, 1999; Brandstätter, Kahr, & Leitner, 2017; Brendel, Zapadka, Brennecke, & Kolbe, 2018c; El Fassi, Awasthi, & Viviani, 2012; Sonneber, Kuehne, & Breitner,) to forecast customer demand (Balac et al., 2018; Boyaci et al., 2015; Ciari, Bock, & Balmer, 2014; Daraio, Cagliero, Chiusano, Garza, & Giordano, 2020; Li, Liao, Timmermans, Huang, & Zhou, 2018; Wang, Cheu, & Lee, 2010) or to implement and evaluate efficient relocation strategies (Ait-Ouahmed, Josselin, & Zhou, 2017; Alfian, Rhee, & Yoon, 2014; Balac et al., 2018; Barth, Todd, & Xue, 2004; Brendel et al., 2018a; Bruglieri, Colorni, & Luè, 2014; Clemente, Fanti, Iacobellis, Ukovich, 2013a; Gambella, Malaguti, Masini, & Vigo, 2018; Jorge et al., 2014; Kek, Cheuo, & Chor, 2006; Nourinejad, Zhu, Bahrami, & Roorda, 2015; Repoux, Boyaci, Geroliminis, Boyaci, & Geroliminis, 2015).

In reviewing the literature, Illgen and Höck (2019) found that relocation research has addressed operational and strategic goals using optimizations (16/35), simulations (13/35), or hybrid solutions (6/35). Researchers have performed optimizations by applying mixed integer programming, such as with CPLEX solvers (Ait-Ouahmed et al., 2017; Bruglieri et al., 2014; Kek, Cheu, Meng, & Fung, 2009; Nourinejad & Roorda, 2014; Sonneberg et al., 2015; Weikl & Bogenberger, 2015). However, the underlying mathematical models' high complexity and the lack of guided system-definition and -evaluation processes can prevent scholars and practitioners from conducting operational research.

Furthermore, in synthesizing the literature, Illgen and Höck (2019) and Jorge and Correia (2013) found that, in the carsharing relocation research domain, 21 projects have relied on simulations. Most projects applied a DES approach (11/21). Accordingly, we can see DES as the standard in the domain (Alfian et al., 2014; Barth & Todd 1999; Brendel et al., 2018a; Clemente et al., 2013a; El Fassi et al., 2012; Herrmann et al., 2014; Kek et al., 2006). However, researchers primarily developed these simulations from scratch and, thus, adapted them highly to meet specific requirements. In contrast, other researchers have built their solutions on existing simulations, such as the ABS framework MATSim (Balac et al., 2018; Laarabi & Bruno, 2017); the DES tool Rockwell Automation ARENA (Clemente et al., 2013a; El Fassi et al., 2012); and AnyLogic (Jorge et al., 2014), which supports ABS, DES, and SD. Existing solutions require less programming skills compared to from-scratch solutions and, therefore, represent a good entry point for users with few programming skills. The more specific the research problem, the more difficult and time consuming researchers will find adapting such frameworks to solve it. In some cases, one might not even be able to exactly match the simulation logic with the research object.

One can conclude that researchers have addressed most carsharing relocation problems by implementing DES into command-line applications. Such applications do not consider requirements such as usability and practicality in particular, which pertain to business environments or researchers without programming background. Such a foundation makes the entry in carsharing research harder because it requires time-consuming introductory training to understand and modify underlying code.

To the best of our knowledge, no tool allows one to implement multiple operation strategies that go beyond changing execution parameters or underlying data models. Existing simulations in literature lack modularity and do not allow users to recombine multiple operation strategies, demand profiles, and system setups. The fact that researchers have used existing solutions only to satisfy a single case study further reflects this low flexibility and customizability.

3 Research Approach

With our research approach, we address issues with current simulation solutions by following Hevner, March, Park, and Ram's (2004) and Hevner's (2007) frameworks in combination. Design science research (DSR) constitutes an effective and efficient problem-solving paradigm that supports researchers in producing innovative ideas, practices, technical capabilities, and products for analyzing, designing, implementing, managing, and using information systems (Hevner et al., 2004).

We depict the DSR setting and its interrelated cycles in Figure 1. The relevance cycle connects the design cycle's activities with the artifact's intended environment. This connection enables researchers to assemble real-world requirements to describe and later solve subsequent real-world problems. Furthermore, one introduces artifacts to the environment during a relevance cycle. The rigor cycle relates the design activities to the existing body of knowledge. Thus, one can integrate existing knowledge into

design activities, and research results can later extend the knowledge base. At the core of the DSR process rests the design cycle, which represents researchers' implementation and evaluation activities.

Figure 1. Design Science Research Setting

We summarize the research activities we conducted in Table 1 and describe them in Sections 3.1 to 3.3. According to Iivari (2015), we follow DSR Strategy 2 by creating our IT artifact CASSI which is rooted in practice to solve the problem of inefficient carsharing relocation research. Based on our design decisions, we formulate a design theory (see Section 4.5) that generalizes our findings into a solution concept for microsimulations.

Table 1. Summary of Research Iterations

Iteration	Relevance cycle	Rigor cycle	Design cycle
1	Conduct expert interviews with researchers and carsharing providers	Literature review about IS for carsharing and discrete event simulations	Develop a carsharing framework concept
2	Hold brainstorming workshop with researchers	Literature review about KPIs for carsharing systems	Implement "CASSI" artifact including KPIs
3	Perform field test with researchers and students	Publication writing, Design Theory	

3.1 Iteration 1: Understanding the Problem Domain

In the relevance cycle, we performed expert interviews with researchers that have worked on carsharing operation strategies and a carsharing provider. They explained the challenges they faced when implementing algorithms and comparing simulation results. From the interviews, we found that researchers often implement carsharing simulations from scratch as a foundation for developing operation strategies. As such, these simulations pertain highly to a specific research context; thus, they do not contain enough flexibility for researchers to reuse them in other scenarios. Simulating a certain operation algorithm under different constraints (e.g., with electric vehicles, in other operation areas, or with multiple combined relocation strategies) constitutes a practice requirement that available solutions do not satisfy. Without modular approaches, researchers cannot reuse the simulation logic they develop, which decreases development efficiency and slows down research in the carsharing domain. Furthermore, one needs programming skills and to understand code/data models to use, tune, and develop carsharing operation strategies. That requirement makes it hard for non-IT-specialists such as students or industry members to produce insights and benefit from findings. Based on these insights, we formulated requirements for tool development.

To gain relevant knowledge about carsharing and carsharing simulations, we analyzed the literature reviews that Cepolina et al. (2014), Illgen and Höck (2019), and Jorge and Correia (2013) conducted. As an extension, we queried ABI/INFORM Complete (via ProQuest), the Association for Computing Machinery Digital Library, the Association for Information Systems eLibrary, Business Source Complete (via EBSCO Host), JSTOR, Wiley, ScienceDirect, Springer Link, and Google Scholar with the following keyword string: "carsharing AND (simulation OR model OR optimization) AND (tool OR framework)".

After screening titles, abstracts, and provided documentation, we selected relevant publications based on three criteria: whether they discussed a carsharing setup, 2) whether they discussed operation strategies (especially vehicle relocation), and 3) whether they used carsharing system simulations. We accepted only publications that cover the instantiation of utilized information systems. After identifying relevant

publications, we determined how researchers have designed current simulations in carsharing relocation research and what requirements such simulations required.

In the subsequent design cycle, we leveraged the knowledge we gathered to conceptualize an initial simulation tool that supports scientists in developing and refining novel algorithms that address optimization problems in carsharing environments.

3.2 Iteration 2: Designing a Comprehensive Simulation Solution

To ensure that our simulation tool worked in practice, we instantiated it as a Web application called “CASSI” (CArSharing Simulator) and performed an internal expert workshop with three researchers in the mobility domain. We explained how the simulation runtime and extensibility interface worked and asked them to reflect on how they would implement their research scenario using our artifact. They evaluated the development process’s effectiveness and simplicity compared to a from-scratch implementation. The researchers agreed that the generic simulation model with extension points had enough versatility such that users could use it to develop problem-tailored simulations and that it would prevent users from writing the same code multiple times and, therefore, accelerate the research process. From the workshop, we also found that CASSI could allow researchers to compare results since it can automatically determine key performance indicators for simulations. In a practice-oriented valuation, researchers would need the ability to compare results to iteratively find the optimal setup and operation strategy. Furthermore, we received requests for CASSI to support the structured process to determine an optimal hyperparameter configuration for the simulation logic by implementing an automated grid search over a given set of parameter ranges.

In the rigor cycle, we gathered knowledge about how research determines a carsharing system’s performance. We systematically reviewed the literature using the following query: “(evaluation OR performance measure OR key performance indicator) AND carsharing AND (model OR tool OR framework)”.

Based on our findings, we refined CASSI’s design and improved it to be a carsharing simulation tool, which allows scholars, researchers, and carsharing providers to develop and evaluate carsharing systems and carsharing operation strategies.

3.3 Iteration 3: Performing Field Test and Writing Publication

To ensure that CASSI worked in practice and that it offered an easy entry into carsharing research, we performed a workshop with eight students who majored in computer science and business informatics, two research associates at our university, and one member from the mobility industry. The participants represent a cross-section of users with different programming skill levels and knowledge regarding carsharing. We introduced CASSI by explaining how it supports the typical research steps system definition, strategy development, simulation run, and performance measurement. Second, we asked them to use the artifact to define a carsharing system in a town they chose and to execute the simulation without a custom strategy as a base case. Third, we tasked the participants with implementing a grid approach to balance the vehicles in the system. In the grid approach, one defines a grid over the operation area and ensures that every field grid contains at least one car. When a field lacks any cars, one should arrange to relocate cars from the field with the most cars in order to optimize the system’s operation. In the task, participants had to also make the grid size configurable over CASSI’s template system so they could identify a reasonable grid size and create a line plot that contrasted the acceptance ratio for different grid sizes. Lastly, the participants filled out a questionnaire to validate the degree to which the presented CASSI instantiation satisfied requirements. We also provided open text fields for further proposals.

Participants agreed that that CASSI offers an easy entry into carsharing relocation research and that they preferred using CASSI over a from-scratch implementation. In particular, we found that CASSI had enough versatility to allow one to model carsharing systems from practice and that development template and documentation constituted good entry points to implement custom relocation strategies. We also found that participants regarded the evaluation mechanisms and sharing functionality as useful features. We addressed suggestions from participants about improving CASSI’s underlying data models in the final artifact.

We completed the last iteration by documenting our research and development process. Furthermore, we generalized our findings as a design theory (Gregor & Jones, 2007).

4 Results

In recent years, new carsharing business models have contributed to the various challenges that existing carsharing infrastructures face in balancing vehicle supply and demand. Correspondingly, we need to develop, adapt, and evaluate novel relocation methods for different carsharing formats (Remane et al., 2016). By combining the insights from the relevance cycles, we derived the requirement set for our solution instantiation. Thus, we iteratively gathered the final requirement set, which we summarize in Table 2.

Table 2. Final Requirement Set for a Carsharing Simulation Tool

	Requirement	Description
R01	Accessibility	Responsive Web application that provides access to the simulation engine's functionality via the frontend. Users should be able to execute the basic simulation without programming skills.
R02	Flexible carsharing system definition	Graphical interface that allows users to flexibly model carsharing systems, which includes assets (e.g., fleet, operation-areas), demand, and accounting figures. The simulation engine should support free-floating and station-based carsharing scenarios.
R03	Programmable simulation	Interface that allows users to upload their code to simulate, optimize, and evaluate the carsharing system in a way that they can tailor to relocation research projects.
R04	Support strategy development process	Offer users step-by-step support in developing and testing user-written operation strategies
R05	Comprehensive performance evaluation	Make it easy to evaluate the influence of decision parameters and to compare different strategies. Ensure that users can loosely couple simulation setups and operation strategies and that they can evaluate performance across projects.
R06	Automate parameter tuning	Provide mechanisms that help users to systematically find optimal parameters from a set of possible configuration values
R07	Share results	Enable users to share results in a way that ensures that others can interactively explore them.
R08	Reference data	Provide reference data for the carsharing system and demand and example implementations for common operation strategies

Synthesizing multiple publications on carsharing simulations, we identified the following core activities in the simulation process (Illgen & Hoeck, 2019):

- 1) System definition: modeling the properties and assets of the carsharing system
- 2) Strategy development: implementing an operation algorithm and defining runtime parameters and decision variables
- 3) Simulation run: executing the strategy in the simulation
- 4) Performance measurement: evaluating the system performance

Based on that structure, we conceptualized a carsharing simulation tool "CASSI" (see Figure 2), which assists with all four steps. We implemented it with Python Django, and it addressed stakeholders' and developers' user role. Addressing R01, R02, R05 and R06, we designed CASSI in a way that allows stakeholders (e.g., practitioners, business-oriented scholars, etc.) to define a system and then to run and evaluate operation strategies without any programming skills. Furthermore, we addressed R03 and R04 with functionality that targets a developer's (e.g., scientists, computer-science-oriented scholars, etc.) role. In doing so, we also provided a structured development template that allows scientists and scholars to develop a setup and operation strategy using Python code. We designed CASSI's backend in a way that allows users to inject those strategies into the core simulation model.

4.1 System Definition

The system setup requires versatility in terms of system design (e.g., support for free-floating and station-based systems) and data importing (e.g., for importing real-world carsharing systems) (R02). CASSI allows users to define custom asset types as part of their system (e.g., multiple car and infrastructure types) (see Appendix A, Figure A1). Addressing heterogeneous fleets, the asset type car supports properties such as engine type and refueling speed while the asset type infrastructure provides charging or refueling spots. Users can instantiate all asset types by adding them to an interactive map (see Figure 3) (R01). The map represents a simulation project's initial state and persists with all of its assets in CASSI's backend.

Figure 2. Architecture of CASSI Simulation Tool

In order to simplify the definition process, we implemented three ways for users to add cars, infrastructure, and operation areas to the system (R01, R02) (see Appendix A, Figure A2):

- 1) Manual mode: users can place items by clicking the map and can delimit operation areas by tagging its boundaries.
- 2) Random mode: CASSI places a selectable number of items randomly on the map by using the Google Maps API in order to ensure that items such as vehicles and charging stations will be placed on open streets.
- 3) Upload mode: users can upload a CSV file that contains the items' coordinates.

To define operation areas, user can also upload coordinates or draw them interactively by clicking on the map (see Appendix A, Figure A3).

Figure 3. CASSI's User Interface to Setup a Carsharing System

Addressing the modularity requirement in R05, users can use the system definition in any simulation context to use the same initial system for different operation strategies. As such, developers can easily apply different operation scenarios to the same system definition.

4.2 Strategy Development

Users who want to implement custom setup and operation strategies with corresponding decision variables (R03, R04) or who want to run custom analytics on the simulation's object model will find this step relevant. Users can skip this step if they focus on analyzing a model's performance in different system setups or demand scenarios since they can define multiple initial setups (see Section 4.1) or provide modified demand information (see Section 4.3).

Addressing the need to simulate custom operation strategies that carsharing researchers face (R03, R04), CASSI provides a development template with predefined extension points that allows developers to analyze the current system state and to influence the simulation's run time. We prepared a step-by-step guide with examples about how to access and change data that users commonly use in the simulation (R01, R04).

The generic simulation, a discrete event simulation, implements the base simulation case as a Python class simulation that applies the demand profile to the carsharing system and evaluates whether the system can satisfy user requests or not. The user has to upload the demand profile to CASSI as a rental history in which the following tuple r represents each rental (R08):

$$r = (\text{origin, destination, } \tau_s, \tau_t, \text{distance, power consumption}) \quad (1)$$

Here, origin describes the pick-up coordinates, destination defines the drop-off coordinates, τ_s represents the start time, τ_t represents the travel time, distance measures the kilometers driven during the rental, and power consumption signifies the energy needed. The history comprises events, whereas the simulation engine evaluates whether the system in its current state can handle the request. The simulation engine satisfies user demand under the constraint that users will walk 500 meters to an available carsharing vehicle (Herrmann et al., 2014). In case the system can process requested configurations, CASSI's generic simulation automatically updates the system state according to the input tuple r .

Figure 4. Actions and Extension Points in CASSI's Simulation Runtime

We illustrate the execution logic, interfaces, and extensibility points that allow developers to customize their research projects in Figure 4. Developers can also define custom input parameters, which allow users to influence strategy behavior (e.g., by applying a different walking distance constraint). When configuring the simulation, CASSI will automatically render the forms for input parameters into its frontend (see Figure 5). In this way, our design simplifies the process that users must follow to modify simulation logic since the Web frontend can do it for them (R01). Making a strategy's behavior adaptable by offering such parameters builds the foundation for the process that users can follow to find an optimal configuration parameter set (R06).

To implement a custom operation strategy, developers can extend the generic simulation class by using Python's inheritance mechanism. In the Web application, they can download the development template and execute it locally. We explain the simulation process's actions and the roles that the provided functional steps play next (R03, R04).

Initialization constitutes the first action in the simulation tool. The generic simulation initializes the required data structures and makes them available as public member variables. The data structure includes the underlying system definition (e.g., operation areas, initial car distribution, car types, infrastructure locations) according to the selected project. It also includes custom parameters that users provide. Users can use extension points before and after the initialization to inject the custom simulation's code into the core simulation logic to, for example, initialize required algorithms such as machine learning models or execute custom preparation tasks. When the initialization process completes, CASSI's backend will notify the Web application to update the progress.

The screenshot shows the 'Car-Sharing Simulator' web interface. The top navigation bar includes 'Projects', 'Göttingen YourCar', 'Strategies', and 'Simulation'. The user is logged in as 'prinz'. The main heading is 'SERA'. Below it, there are two main sections:

- 1. Load Booking Data**: A text input field contains 'Buchungen 2016 für CARSIM.txt (17814 lines detected)' with a 'Browse' button to its right. Below this is a toggle switch for 'Test Mode: Just execute first 100 bookings', which is currently turned 'On'.
- 2. Setup Simulation 'SERA'**: This section contains a 'Simulation Parameter' box. Inside, there are two tabs: 'Parameter Mode' (selected) and 'Grid Search Mode (Advanced)'. Below the tabs are three input fields:
 - location_epsilon**: [0.0035, 0.007]
 - c1**: [0.01, 1, 10, 100, 1000, 10000, 100000, 1000000]
 - acceptance_threshold**: [1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987]

At the bottom of the interface is a large blue button labeled 'Start Simulation' with a play icon.

Figure 5. CASSI's User Interface to Configure and Run a Simulation

The simulation loop constitutes the second layer in the simulation tool. It frames multiple simulations based on the same data model but different execution parameters. Users might input the execution parameters, or they may result from a solver. The custom simulation can use the pre-extensibility point to, for example, reset data structures or maintain different execution parameters. Users can use the post-extensibility point to, for example, process and pack performance measures.

The rental loop constitutes the heart of the simulation. The generic simulation fires discrete events that represent demand based on the uploaded demand profile. Whenever the system can satisfy user demand, it will update the vehicle's position and fuel state/battery level. In the rental loop, the simulation collects the data it needs to calculate the key performance indicators (KPIs) (e.g., acceptance ratio). In order to balance the supply, the custom simulation can use the rental's pre- and post-extensibility points to inject commands. Developers commonly inject rules for triggering a relocation decision into the custom strategy, which the simulation then executes based on the current system. Furthermore, one can update the object model to provide custom analytics in the evaluation steps. Once the simulation processes demand, it returns the results to the simulation loop. As soon as the simulation finishes, it will provide the system's object model to the frontend as a dictionary. This model includes cars' position and state, performance evaluation figures, and any data a developer provides in a custom implementation.

After developing a custom strategy, users can upload their modified simulation logic. CASSI executes a unit test and reports the results, and, if it detects an error, specifies the problem's cause and position. Users can apply uploaded strategies to any system definition and share them among other CASSI users. Thus, the system allows, for example, students with little programming experience to investigate the influence that decision variables, demand profiles, and operation strategies have on a carsharing environment.

4.3 Simulation Run

After users have selected a system definition and an operation strategy based on the development template that we describe in Section 4.2, they can configure the simulation. CASSI requires users to provide demand data and to fill out the input parameter since it uses such information to create the simulation (see Figure 5).

The form for the input parameters loads dynamically depending on the form definition in the strategy template. In the example we provide (see Figure 5), the user defined the parameter `location_epsilon`, `c1`, and `acceptance_threshold`, which influenced the developed operation strategy. The simulation permutes and executes the given parameter values in various simulation runs in order to find the best performing configuration set. The backend runs the simulation and returns the system state and implemented performance indicators as a dictionary. Addressing R05, CASSI'S interface can display the dictionary's values as raw data, as scatter and line plots, and as a heat map (see Figure 6). The application stores the way in which users configure the evaluation view across multiple executions, which allows users to compare the system performance when input parameters or strategies change. CASSI also supports autocomplete, which allows users to access the simulations data model easily (R01). Users can further restart the simulation with changed input parameters or demand data.

Figure 6. CASSI's User Interface for Visualizing the Simulation Results as a Line Plot (Left) and Heat Map (Right)

Furthermore, we implemented a semi-automatic grid search that users can set up as part of the simulation settings. The search allows users to systematically compare execution configurations (R06). CASSI will automatically execute every parameter value set (see arrays in Figure 5) in an independent simulation loop. The simulation engine stores performance measures separately for each iteration, which makes it easy for users to compare how different parameter configurations perform. CASSI remembers the output configuration and will automatically update how it visualizes the performance measures once the new simulation completes (R06).

4.4 Performance Measures

Researchers do not use uniform performance indicators to evaluate how well operation strategies perform. In particular, finding an optimal strategy or parameter configuration requires uniform and practice-oriented performance measures. Instead, researchers often implement individual performance measures that target profit, service quality, or relocation effort. To solve this evaluation confusion and to make research in CASSI and in other publications more comparable, we selected the most common key performance indicators (KPIs) that CASSI will automatically determine in the simulation run (R05). We summarize the selected KPIs and the formula to determine their values in Table 3.

When users enable the grid search mode in the optimization process, CASSI highlights the best hyperparameter configuration (R06) depending on a selected performance measure. The evaluation view allows users to compare the system performance across the simulation iterations and also displays the performance improvement compared to the worst performing parameter set (see Figure 7). Helping to make research more comprehensible, we enable users to share the results and performance indicators of their carsharing simulation experiments. CASSI can also provide a public read-only Web link to the result view (R07).

Table 3. KPIs to Evaluate the Performance of Carsharing Systems

KPI	Formula	Demonstrated by
Service quality		
Acceptance ratio	$\frac{\sum \text{completed reservations}}{\sum \text{total reservations}}$	(Alfian et al. 2017; Nourinejad et al., 2015)
Average fleet usage	$\frac{\sum \text{vehicles operated}}{\sum \text{vehicles possible operated}}$	(Alfian, Rhee, Kang, & Yoon, 2015; El Fassi et al., 2012)
Relocations		
Relocations ratio	$\frac{\sum \text{relocations}}{\sum \text{accepted rentals}}$	(Brendel et al., 2018a; Wagner, Willing, Brandt, & Neumann, 2015)
Number of relocations	$\sum \text{relocations}$	(Brendel et al., 2018a; Kek et al., 2009)
Profit		
Utilization ratio	$\frac{\sum \text{hours vehicles used}}{\sum \text{hours vehicles available}}$	(Alfian et al., 2015; Nourinejad & Roorda, 2014)
Profit after relocations	$\sum \text{revenue} - \sum \text{relocations} * \text{cost}_{\text{reloc}}$	(Clemente et al., 2013b; Nourinejad & Roorda, 2014)

Figure 7. CASSI's User Interface to Evaluate how a Simulation Run Performs

4.5 Design Theory

In this section, we present our nascent design theory to generalize our research results on how one should design a carsharing simulation tool. In iterating on our DSR project, we performed rigor cycles to build a knowledge base. Throughout that process, we specified a nascent design theory's components following Gregor and Jones (2007). Furthermore, we generalized the requirements and design decision that we present in previous sections to meta-requirements and design principles. We suggest that other researchers apply our findings to the more generic microsimulation domain (e.g., in logistics, traffic, geosystems, processes, financial transactions). In Table 4, we describe our design theory's components and link our design principles to the requirements that our artifact addresses. One can categorize design theory's knowledge contribution type by determining the problem's and solution's maturity. Our research theory addresses problems in the mature application and research domains that examine carsharing and carsharing relocation strategies. However, existing carsharing simulation solutions and relocation strategies lack effectiveness and maturity. Therefore, one can consider our knowledge contribution type an improvement that "create[s] better solutions in the form of more efficient and effective products, processes, services, technologies, or ideas" (Gregor & Hevner, 2013, p. 346).

Table 4. CASSI Design Theory

Component	Description
Purpose and scope	CASSI supports researchers in developing strategies for setting up and operating carsharing systems. It helps make carsharing relocation research more accessible for scholars and practitioners. Meta-requirements: system state visualization, flexible system definition, simulation programming interface, guidance, structured optimization process, uniform performance measurement, data interchangeability, real-world data
Constructs	Fleet, operation-areas, infrastructure, map, simulation template, development guide, decision variables, strategy test, optimization process, events (demand), discrete event simulation, relocation, plots, heatmap, KPIs
Principle of form and function	DP01: the development process follows structured steps: system definition, strategy development, simulation run, and performance measurement (R02, R03, R04, R05, R06) DP02: carsharing system definition comprises infrastructure, fleet, and operation areas; the artifact visualizes the definition (R01, R02) DP03: historical demand serves as the input for the discrete event simulation (R04, R08) DP04: predefined extension points, step-by-step development guide, and validation methods simplify programming (R03, R04, R08) DP05: user and machine interfaces to export and import data and code enable versatility (R01, R02, R04, R07) DP06: a grid search mode for simulation parameters structures how users enumerate simulations and helps them find an optimal solution (R06) DP07: easy-to-understand results help users evaluate the impact of parameter and system changes (R01, R05, R07) DP08: uniform performance measures enable comprehensive evaluation (R05, R07)
Artifact mutability	Users can freely define the carsharing system (FF, SB), assets, the fleet, and their properties. Users can upload code to influence the simulation process at predefined extension points. They can also model dependencies and adapt the execution with flexible decision variables. Users can use flexible data sources for the output figures and heat maps.
Testable propositions	The tool helps stakeholders, researchers, and scholars to design and evaluate relocation algorithms for carsharing and uniformly presents simulation results. Thus, stakeholders, researchers, and scholars can save time and effort compared to if they adopted a from-scratch implementation.
Justificatory knowledge	Carsharing literature, discrete event simulation, object-oriented programming, geocoding, data analysis, continuous improvement process, hyperparameter optimization

First, our design theory addresses the system's purpose and scope. We instantiated a software artifact that offers a simulation framework that helps users optimally set up and operate carsharing systems. In the relevance cycles, we derived requirements that we abstracted into meta-requirements for the more generic microsimulation class. This class represents analytical tools that allow one to model system properties and behavior in order to evaluate of the effects that changed system properties or introduced

process improvements have before one implements them in the real world. In particular, the design principles we propose reduce the entry barrier for scholars and practitioners into operation research.

Second, we specified the entities of interest in our theory (constructs): the subjects of interest, processes and algorithms, and the artifact's components. Some constructs pertain specifically to the carsharing context (e.g., fleet, operation areas, infrastructure, map, relocation), while others can apply to more generic settings (e.g., simulation template, development guide, decision variables, strategy testing, optimization process, events, discrete event simulation, plots, heatmap, KPIs).

Third, we described the design principles (form and function principles) that we derived to fulfill the requirements and meta-requirements. Finding an appropriate system setup and operation algorithm contains four steps: 1) system definition, 2) strategy development, 3) simulation run, and 4) performance measure (DP01). Each step in the process provides user- and machine-friendly ways to influence its execution and to process the outcome (DP05). The system definition comprises infrastructure, fleet, and operation areas (DP02), and users can completely adapt it via the frontend or XML files. Users can easily understand simulation results since the artifact presents them in adaptable plots and heat maps. The artifact also allows users to process simulation results externally since it can export them in JavaScript Object Notation (JSON) format (DP07). Uniform performance measures for the simulation ensure users can compare research results (DP08). Using real-world data as a simulation basis increases findings' practicality, meaning, and impact (DP03). At its core, the framework provides pre-defined extension points where users can upload custom execution logic (DP04). Step-by-step guides, example strategies, a strategy template, and a code-validation interface supports users in the development process. To help users find the best performing solution, we implemented a grid search in a user-definable parameter set (DP06). In the end, in designing simulation strategies, one must ensure that non-IT experts can easily use and understand them and professionals and researchers can sufficiently modify them to meet their requirements. CASSI addresses ease of use with its Web interface that users can use to set up systems and run generic or shared relocation algorithms without the need to adapt program code. On the other hand, CASSI allows professional users to implement the Python interface that CASSI uses and to develop and optimize relocation strategies that they can pertain to particular project contexts.

Fourth, our design theory explains how one can design an artifact to ensure that one can easily adapt or extend it, which ensures high mutability. We achieved this adaptability by implementing a modular system with extension points where users can inject custom code. Languages such as Python allow flexible data and object models and the dictionary structure enables type-independent data exchange. Providing user interfaces to adapt decision variables, input data, and reporting figures allow users to adapt the system to specific contexts.

Fifth, we state how users can ensure that a solution solves a specific problem (testable propositions). The proposed microsimulation should help stakeholders, researchers, and scholars to design and evaluate relocation algorithms for carsharing, and it should uniformly present simulation results. Thus, stakeholders, researchers, and scholars must save time and effort compared to if they adopted a from-scratch implementation.

Finally, we list sources that cover the underlying system, discrete-event simulations, programming, and applied strategies to explain why and how our system fulfills the testable propositions.

5 Discussion

In the research project that we present in this paper, we developed a new simulation tool that supports researchers in implementing and evaluating novel and existing relocation algorithms.

During the second iteration, we found from the expert workshop showed that users could implement the SERA relocation algorithm, which Brendel et al. (2018a) introduced, using CASSI's development template. We successfully rebuilt the SERA relocation algorithm and validated the original development's results. During the third iteration, we found from the workshop with students that users without programming skills could define a carsharing setup in CASSI and execute a simulation based on a shared relocation strategy. Furthermore, with assistance from the provided documentation, scholars with programming skills successfully programmed a simple, threshold-based relocation strategy.

One cannot sufficiently compare CASSI's improvement in development efficiency against other tools that related work has presented since the corresponding artifacts remain mostly private. In both workshops

that we conducted, we confirmed that our proposed artifact answers the need for an accessible carsharing simulator that helps researchers and practitioners develop and evaluate carsharing operation strategies.

5.1 Theoretical Contributions

With this study, we contribute to the IS knowledge base via exhaustively analyzing state-of-the-art carsharing simulation solutions and relocation algorithms. We synthesize requirements from carsharing simulations to point out existing microsimulation solutions' benefits and shortcomings. We structure the process to develop operation algorithms and introduce novel and, whenever appropriate, established approaches to support each underlying step with information technology. We define universal data structures and suggest architecture for a framework to customize execution logic.

Moreover, we synthesize strategies to measure carsharing systems' performance from publications in relocation research. We suggest KPIs to the research community that will enable researchers to uniformly evaluate a carsharing system's service quality, profit, and relocation performance. We show that following interactive system definition process and presenting results in plots or heat maps makes it easier for users to understand system behavior. To generalize our findings to microsimulations, we iteratively developed a design theory for our simulation framework. Hence, we contribute to explaining how one can design a structured optimization process.

5.2 Practical Contributions

The artifact we developed, CASSI, allows users to develop algorithms to optimize carsharing systems and their operation more efficiently than with other tools. Users save time and effort in comparison to from-scratch developments because CASSI provides a ready-to-use generic simulation framework and dynamic data structures. Still, users can apply a developed algorithm to other carsharing systems. Furthermore, we prove that a uniform and highly supported development process does not necessarily eliminate solution variability. CASSI allows scholars and non-IT-experts to easily access relocation research, which should help amplify activity in the domain. Thanks to the increased efficiency, we expect researchers to develop more carsharing operation strategies. By providing uniform KPIs to help stakeholders evaluate how developed algorithms perform in practice, we simplify efforts to identify promising approaches and direct the focus to constructive future research. Furthermore, the mechanism to share simulation results with CASSI marks a step towards comprehensive and transparent research that might also inspire other research domains. Finally, our nascent design theory gives researchers guidance to develop other microsimulation solutions. Such guidance can lead to more practical contributions and artifacts that help students and researchers to contribute to their research domain.

5.3 Limitations

Our instantiated simulation artifact has several limitations. First, the architecture as a Web application offers a user-friendly and state-of-the-art approach but also causes problems. Allowing users to upload code into the backend represents a security risk, though we did not quantify the extent. Furthermore, CASSI executes strategies without the frontend, which means the execution context becomes "unclaimed" when users leave or reload the webpage. Our design lacks an opportunity for users to stop those unclaimed simulation processes, which can significantly slow down the backend server's performance.

Second, data availability limited our evaluation process. We were only able to test our artifact with data from a single free-floating carsharing operator. Therefore, we did not perform real data tests for station-based systems or system with electric cars. Furthermore, the data about the rental history that we used to model demand did not contain rejected or unsatisfied requests and, therefore, did not represent demand precisely.

Third, our presented performance measures make results in CASSI comparable, but they need to become established in the research domain to enable the comparison with other implementations. The measures highly depend on assumptions on which the simulation relies. For example, customers' assumed willingness to walk when picking up a car influences the acceptance ratio in the system. Since the research domain does not unify those assumptions, one can ensure performance measures' meaningfulness only by considering system constraints and assumptions.

5.4 Future Research

Future research should first validate whether one can apply CASSI's process for optimizing carsharing to station-based carsharing and systems with more complex constraints, such as charging requirements for electric cars. Also, researchers can improve our artifact's versatility by adding support for other shared-economy systems such as bikesharing or ridesharing.

Second, we suggest conducting future research on how to design the strategy development entirely as part of CASSI's frontend. A possible solution includes an online code editor that features code completion, syntax validation, and built-in documentation. Such a solution would mean that users would no longer need to develop their strategies on their local machines and make CASSI a standalone solution. As a contribution to carsharing relocation research, CASSI users should implement established and novel relocation strategies in order to create more benchmark solutions and to provide code examples in a shared matter.

Third, we suggest that researchers examine how users could generate and provide valuable input data. Our proposed artifact requires users to upload rental requests (possibly from a real-world system). However, acquiring such data often requires the cooperation with a carsharing provider, and such data does not represent the actual demand since it does not include unsatisfied user requests. Research should address strategies to generate artificial datasets or to apply custom demand patterns or distributions to existing datasets.

Finally, researchers should and improve test CASSI in further practice projects and case studies by finding empirical evidence that it offers easy access to carsharing research and by validating that users can convert its simulation results into practice benefits. In particular, researchers could give different students or practitioners a task to implement a certain simulation scenario in which some use CASSI and others build their own simulations based on other frameworks or from scratch to generate insights into CASSI's benefits and shortcomings. Consequently, researchers should instantiate and evaluate the design principles we propose in other problem settings in microsimulation domain.

6 Conclusion

In this paper, we discuss how we applied the DSR paradigm to design and develop a carsharing simulation tool called CASSI. We developed a versatile simulator that structures, supports, and unifies the research process of optimizing carsharing systems and their operation and validated its practicality in a field test with researchers and scholars. We synthesized requirements from carsharing simulations to point out existing microsimulation solutions' benefits and shortcomings. We structured the process of developing project tailored relocation solutions and introduced novel and, whenever appropriate, established approaches to support each underlying step (system setup, strategy development, simulation run, and evaluation) with information technology. We defined universal data structures and suggested architecture for a tool to customize execution logic. However, we concluded that the availability of real-world system data significantly improves a simulation's validity. We suggest KPIs that will enable the research community to uniformly evaluate a carsharing system's service quality, profit, and relocation performance. We showed that using an interactive system definition process and presenting results in plots or heat maps makes it easier for users to understand system behavior.

References

- Ait-Ouahmed, A., Josselin, D., & Zhou, F. (2017). Relocation optimization of electric cars in one-way car-sharing systems: Modeling, exact solving and heuristics algorithms. *International Journal of Geographical Information Science*, 32(2), 367-398.
- Alfian, G., Rhee, J., Ijaz, M., Syafrudin, M., & Fitriyani, N. (2017). Performance Analysis of a Forecasting Relocation Model for One-Way Carsharing. *Applied Sciences*, 7(6), 1-22.
- Alfian, G., Rhee, J., Kang, Y. S., & Yoon, B. (2015). Performance comparison of reservation based and instant access one-way car sharing service through discrete event simulation. *Sustainability*, 7(9), 12465-12489.
- Alfian, G., Rhee, J., & Yoon, B. (2014). A relocation simulation model for one-way carsharing service. In *Proceedings of the IEEE International Conference on Industrial Technology*.
- Axhausen, K. W., Nagel, K., & Horni, A. (2016). *The multi-agent transport simulation MATSim*. London, UK: Ubiquity Press.
- Balac, M., Becker, H., Ciari, F., & Axhausen, K. (2018). Modeling competing free-floating carsharing operators: A case study for Zurich, Switzerland. *Transportation Research Part C*, 98, 101-117.
- Baptista, P., Melo, S., & Rolim, C. (2014). Energy, environmental and mobility impacts of car-sharing systems: Empirical results from Lisbon, Portugal. *Procedia—Social and Behavioral Sciences*, 111, 28-37.
- Barth, M., Todd, M., & Xue, L. (2004). User-based vehicle relocation techniques for multiple-station shared-use vehicle systems. In *Proceedings of the Transportation Research Board 80th Annual Meeting*.
- Barth, M., & Todd, M. (1999). Simulation model performance analysis of a multiple station shared vehicle system. *Transportation Research Part C: Emerging Technologies*, 7(4), 237-259.
- Becker, H., Ciari, F., & Axhausen, K. W. (2018). Measuring the car ownership impact of free-floating car-sharing—a case study in Basel, Switzerland. *Transportation Research Part D: Transport and Environment*, 65, 51-62.
- Boyaci, B., Zografos, K. G., & Geroliminis, N. (2015). An optimization framework for the development of efficient one-way car-sharing systems. *European Journal of Operational Research*, 240(3), 718-733.
- Boysen, N., Briskorn, D., & Schwerdfeger, S. (2019). Matching supply and demand in a sharing economy: Classification, computational complexity, and application. *European Journal of Operational Research*, 278(2), 578-595.
- Brailsford, S. C., Eldabi, T., Kunc, M., Mustafee, N., & Osorio, A. F. (2019). Hybrid simulation modelling in operational research: A state-of-the-art review. *European Journal of Operational Research*, 278(3), 721-737.
- Brandstätter, G., Kahr, M., & Leitner, M. (2017). Determining optimal locations for charging stations of electric car-sharing systems under stochastic demand. *Transportation Research Part B: Methodological*, 104, 17-35.
- Brendel, A. B., Brauer, B., & Hildebrandt, B. (2016). Toward user-based relocation information systems in station-based one-way car sharing. In *Proceedings of the 22nd Americas Conference on Information Systems*.
- Brendel, A. B., Brennecke, J. T., & Kolbe, L. M. (2018a). Applying econophysics in the context of carsharing—development of a vehicle relocation algorithm and decision support system. In *Proceedings of the International Conference on Information Systems*.
- Brendel, A. B., Lichtenberg, S., Brauer, B., Nastjuk, I., & Kolbe, L. M. (2018b). Improving electric vehicle utilization in carsharing: A framework and simulation of an e-carsharing vehicle utilization management system. *Transportation Research Part D: Transport and Environment*, 64, 230-245.

- Brendel, A. B., Zapadka, P., Brennecke, J. T., & Kolbe, L. M. (2018c). A decision support system for computation of carsharing pricing areas and its influence on vehicle distribution. In *Proceedings of the International Conference on Information Systems*.
- Bruglieri, M., Colorni, A., & Luè, A. (2014). The vehicle relocation problem for the one-way electric vehicle sharing: An Application to the milan case. *Procedia—Social and Behavioral Sciences*, 111, 18-27.
- Cepolina, E. M., Farina, A., & Pratelli, A. (2014). Car-sharing relocation strategies: A state of the art. In A. Pratelli (Ed.), *Public mobility systems*. London, UK: WIT Press.
- Čertický, M., Jakob, M., Píbil, R., & Moler, Z. (2014). Agent-based simulation testbed for on-demand mobility services. In *Procedia Computer Science*, 32, 808-815.
- Ciari, F., Bock, B., & Balmer, M. (2014). Modeling station-based and free-floating carsharing demand. *Transportation Research Record: Journal of the Transportation Research Board*, 2416, 37-47.
- Clemente, M., Fanti, M. P., Iacobellis, G., & Ukovich, W. (2013a). A discrete-event simulation approach for the management of a car sharing service. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*.
- Clemente, M., Fanti, M. P., Mangini, A. M., & Ukovich, W. (2013b). The vehicle relocation problem in car sharing systems: Modeling and simulation in a Petri net framework. In J. M. Colom & J. Desel (Eds.), *Application and theory of petri nets and concurrency* (LNCS vol. 7927 pp. 250-269). Berlin: Springer.
- Daraio, E., Cagliero, L., Chiusano, S., Garza, P., & Giordano, D. (2020). Predicting car availability in free floating car sharing systems: Leveraging machine learning in challenging contexts. *Electronics*, 9(8), 1-22.
- Degirmenci, K., & Breitner, H. (2014). Carsharing: A literature review and a perspective for information systems research. In *Proceedings of the Multi-Conference on Information Systems*.
- Di Febbraro, A., Sacco, N., & Saeednia, M. (2018). One-way car-sharing profit maximization by means of user-based vehicle relocation. *IEEE Transactions on Intelligent Transportation Systems*, 20(2), 1-14.
- Di Febbraro, A., Sacco, N., & Saeednia, M. (2013). One-way carsharing. *Transportation Research Record: Journal of the Transportation Research Board*, 2319, 113-120.
- El Fassi, A., Awasthi, A., & Viviani, M. (2012). Evaluation of carsharing network's growth strategies through discrete event simulation. *Expert Systems with Applications*, 39(8), 6692-6705.
- Firnkorn, J., & Müller, M. (2011). What will be the environmental effects of new free-floating car-sharing systems? The case of car2go in Ulm. *Ecological Economics*, 70(8), 1519-1528.
- Fishman, G. S. (1978). *Principles of discrete event simulation*. New York, NY: John Wiley and Sons.
- Gambella, C., Malaguti, E., Masini, F., & Vigo, D. (2018). Optimizing relocation operations in electric car-sharing. *Omega*, 81, 234-245.
- Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, 37(2), 337-355.
- Gregor, S., & Jones, D. (2007). The anatomy of a design theory. *Journal of the Association for Information Systems*, 8(5), 312-335.
- Herrmann, S., Schulte, F., & Voß, S. (2014). Increasing acceptance of free-floating car sharing systems using smart relocation strategies : A survey based study of car2go Hamburg. In R. G. González-Ramírez, F. Schulte, S., Voß, & J. A. Ceroni Díaz (Eds.), *Computational logistics* (LNCS vol. 8760, pp. 151-162). Berlin: Springer.
- Hevner, A. (2007). A three cycle view of design science research. *Scandinavian Journal of Information Systems*, 19(2), 87-92.
- Hevner, A., March, S., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75-105.

- Hildebrandt, B., Hanelt, A., Piccinini, E., Kolbe, L. M., & Niero-Bisch, T. (2015). The value of IS in business model innovation for sustainable mobility services—the case of carsharing. In *Proceedings of the International Conference on Wirtschaftsinformatik*.
- Iivari, J. (2015). Distinguishing and contrasting two strategies for design science research. *European Journal of Information Systems*, 24(1), 107-115.
- Illgen, S., & Höck, M. (2019). Literature review of the vehicle relocation problem in one-way car sharing networks. *Transportation Research Part B: Methodological*, 120, 193-204.
- Jahangirian, M., Eldabi, T., Naseer, A., Stergioulas, L. K., & Young, T. (2010). Simulation in manufacturing and business: A review. *European Journal of Operational Research*, 203(1), 1-13.
- Jorge, D., & Correia, G. (2013). Carsharing systems demand estimation and defined operations: A literature review. *European Journal of Transport and Infrastructure Research*, 13(3), 201-220.
- Jorge, D., Correia, G. H. A., & Barnhart, C. (2014). Comparing optimal relocation operations with simulated relocation policies in one-way carsharing systems. *IEEE Transactions on Intelligent Transportation Systems*, 15(4), 1667-1675.
- Kek, A., Cheu, R., & Chor, M. (2006). Relocation simulation model for multiple-station shared-use vehicle systems. *Transportation Research Record: Journal of the Transportation Research Board*, 1986(1), 81-88.
- Kek, A., Cheu, R. L., Meng, Q., & Fung, C. H. (2009). A decision support system for vehicle relocation operations in carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, 45(1), 149-158.
- Laarabi, M. H., & Bruno, R. (2017). A generic software framework for carsharing modelling based on a large-scale multi-agent traffic simulation platform. In M. R. Namazi-Rad, L. Padgham, P. Perez, K. Nagel, & A. Bazzan (Eds.), *Agent based modelling of urban systems* (LNCS vol. 10051). Berlin: Springer.
- Lavalle, S., Lesser, E., Shockley, R., Hopkins, M. S., & Kruschwitz, N. (2011). Big data, analytics and the path from insights to value. *MITSloan Management Review*, 52(2), 21-32.
- Li, Q., Liao, F., Timmermans, H. J. P., Huang, H., & Zhou, J. (2018). Incorporating free-floating car-sharing into an activity-based dynamic user equilibrium model: A demand-side model. *Transportation Research Part B: Methodological*, 107, 102-123.
- Maidstone, R. (2012). Discrete event simulation, system dynamics and agent based simulation: Discussion and comparison. *System*, 1-6.
- Martin, E., & Shaheen, S. (2011). The impact of carsharing on household vehicle ownership. *Access Magazine*, 1(38), 22-27.
- Nijland, H., van Meerkerk, J., & Hoen, A. (2015). *Impact of car sharing on mobility and CO2 emissions*. PBL Netherlands Environmental Assessment Agency. Retrieved from https://www.pbl.nl/sites/default/files/downloads/PBL_2015_Note_Impact_of_car_sharing_1842.pdf
- Nourinejad, M., & Roorda, M. J. (2014). A dynamic carsharing decision support system. *Transportation Research Part E: Logistics and Transportation Review*, 66, 36-50.
- Nourinejad, M., Zhu, S., Bahrami, S., & Roorda, M. J. (2015). Vehicle relocation and staff rebalancing in one-way carsharing systems. *Transportation Research Part E*, 81, 98-113.
- Remane, G., Nickerson, R. C., Hanelt, A., Tesch, J. F., Nickerson, R. C., Tesch, J. F., & Kolbe, L. M. (2016). A Taxonomy of Carsharing Business Models. In *Proceedings of the International Conference on Information Systems*.
- Repoux, M., Boyaci, B., Geroliminis, N., Boyacı, B., & Geroliminis, N. (2015). Simulation and optimization of one-way car-sharing systems with variant relocation policies. In *Proceedings of the Transportation Research Board 94th Annual Meeting*.
- Rickenberg, T. A. A., Gebhardt, A., & Breitner, M. H. (2013). A decision support system for the optimization of electric car sharing stations. In *Proceedings of the European Conference on Information Systems*.

- Schiller, T., Scheidl, J., & Pottebaum, T. (2017). Car sharing in Europe: Business models, national variations and upcoming disruptions. *Deloitte*. Retrieved from <https://www2.deloitte.com/content/dam/Deloitte/de/Documents/consumer-industrial-products/CIP-Automotive-Car-Sharing-in-Europe.pdf>
- Shaheen, S. A., Cohen, A. P., & Chung, M. S. (2008). North American carsharing: Ten-year retrospective. *Transportation Research Record: Journal of the Transportation Research Board*, 2110(1), 35-44.
- Shaheen, S., Sperling, D., & Wagner, C. (1997). Carsharing in Europe and North America: Past, present, and future. *Transportation Research Part D: Transport and Environment*, 2(2), 107-123.
- Sharp, J. A., & Price, D. H. R. (1984). System dynamics and operational research: An appraisal. *European Journal of Operational Research*, 16(1), 1-12.
- Sonneberg, M.-O., Kuehne, K., & Breitner, M. H. (2015). A decision support system for the optimization of car sharing stations. In *Proceedings of the International Conference on Information Systems*.
- Wagner, S., Willing, C., Brandt, T., & Neumann, D. (2015). Data analytics for location-based services: enabling user-based relocation of carsharing vehicles. In *Proceedings of the International Conference on Information Systems*.
- Wang, H., Cheu, R., & Lee, D. H. (2010). Dynamic relocating vehicle resources using a microscopic traffic simulation model for carsharing services. In *Proceedings of the 3rd International Joint Conference on Computational Sciences and Optimization*.
- Weigl, S., & Bogenberger, K. (2013). Relocation strategies and algorithms for free-floating car sharing systems. *IEEE Intelligent Transportation Systems Magazine*, 5(4), 100-111.
- Weigl, S., & Bogenberger, K. (2015). A practice-ready relocation model for free-floating carsharing systems with electric vehicles—mesoscopic approach and field trial results. *Transportation Research Part C: Emerging Technologies*, 57, 206-223.
- Zoepf, S. M., & Keith, D. R. (2016). User decision-making and technology choices in the U.S. carsharing market. *Transport Policy*, 51, 150-157.

Appendix: CASSI's System Definition Interfaces

The figure shows two side-by-side forms for creating new types in the CASSI system.

Create new car type (Left):

- Name*:** A text input field.
- Fuel type*:** A dropdown menu with a downward arrow.
- Time to refuel seconds*:** A text input field.
- Icon filename*:** A text input field containing "cars/car_base.png".
- Buttons:** A blue "Save" button and a "Back to Project" button.

Create new infrastructure type (Right):

- Name*:** A text input field.
- Car capacity*:** A text input field.
- Provided fuel*:** A dropdown menu with a downward arrow.
- Icon filename*:** A text input field containing "infrastructure/location.png".
- Buttons:** A blue "Save" button and a "Back to Project" button.

Figure A1. CASSI's Interface to Define New Car Types (Left) and New Infrastructure Types (Right)

The figure shows three screenshots of the "Select strategy to add 'Seat Mii'" dialog box, illustrating different strategies for adding assets.

Top Left (Random Strategy):

- Strategy:** Random
- Instructions:** "Please select the amount of cars you want to add to the map. The cars will be distributed randomly into the displayed area of the map."
- Number of Cars:** 20
- Checkbox:** ☐ Replace existing cars of this type
- Buttons:** "Add Cars" and "Close"

Top Right (Read File Strategy):

- Strategy:** Read File
- Instructions:** "Please upload a csv file that contains the initial positions of the cars. Expects a comma-seperated file (Longitude, Latitude), without header-line."
- Upload car positions:** A text input field with a "Browse" button.
- Checkbox:** ☐ Replace existing cars of this type
- Buttons:** "Add Cars" and "Close"

Bottom (Manual Strategy):

- Strategy:** Manual
- Instructions:** "You can add cars to the map by just clicking into the map view."
- Checkbox:** ☐ Replace existing cars of this type
- Buttons:** "Add Cars" and "Close"

Figure A2. Strategies to Add Assets such as Vehicle into the Initial Setup of a CASSI Project

Figure A3. CASSI's User Interface to Define Operation Areas

About the Authors

Christoph Prinz is a research associate and doctoral candidate at the Chair of Information Management at the University of Göttingen. Holding a degree in computer science, his research focuses on leveraging the potential of data in shared mobility systems. As part of the Smart Mobility Research Group, he performs design oriented research in the context of sustainable mobility, crowd working and machine learning. His research has been published in proceedings of established conferences like *European Conference on Information Systems* and *Internationale Tagung der Wirtschaftsinformatik*.

Mathias Willnat is a research associate and doctoral candidate at the Chair of Information Management at the University of Göttingen. Holding degrees in both economics and transportation economics, Mathias operated as a statistical analyst, developer and operations manager in the transportation sector in the past. As part of the Smart Mobility Research Group, he focuses his studies on patterns and implications of novel transportation concepts. His research has been published at the *International Conference on Information Systems*, *Americas Conference on Information Systems* and *European Conference on Information Systems*.

Alfred Benedikt Brendel is the chairholder for business informatics, esp. intelligent systems and services, at Technische Universität Dresden, Germany, and holds a doctoral degree from the University of Göttingen. His research focuses on Design Science Research to develop novel design principles and theories in the areas of digital health, smart mobility, and crowd working. His work has been published in leading IS conferences, such as the *International Conference on Information Systems* and *European Conference on Information Systems*, and renowned journals, including *Business & Information Systems Engineering* and *Information Systems Frontiers*.

Sascha Lichtenberg is a doctoral candidate at Technische Universität Dresden, Germany chair for business informatics, especially intelligent systems and services. In his position as head of the digital work research group, he is researching feasibility of persuasive systems with a focus on gamification and conversational agent to increase motivation and performance in various contexts. Sascha published multiple articles in proceedings of established conferences, such as the *International Conference on Information Systems*, and well-regarded journals, like the *Transportation Research Part D: Transport and Environment*.

Lutz Kolbe is a professor of Information Systems and leads the Chair of Information Management at the University of Göttingen. His research focuses on the management of information and information technology as a crucial factor for sustainable business success. He manages the research and project activities of three groups covering the areas of digital transformation, digital health, and smart mobility. His publications have appeared in several IS journals and conferences, such as the *European Journal of Information Systems*, the *Journal of the Association for Information Systems*, and the *Information Systems Journal*.

Copyright © 2021 by the Association for Information Systems. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than the Association for Information Systems must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or fee. Request permission to publish from: AIS Administrative Office, P.O. Box 2712 Atlanta, GA, 30301-2712 Attn: Reprints via e-mail from publications@aisnet.org.

Editor-in-Chief

<https://aisel.aisnet.org/thci/>

Fiona Nah, Missouri University of Science and Technology, USA

Advisory Board

Izak Benbasat, University of British Columbia, Canada
John M. Carroll, Penn State University, USA
Phillip Ein-Dor, Tel-Aviv University, Israel
Dennis F. Galletta, University of Pittsburgh, USA
Shirley Gregor, National Australian University, Australia
Elena Karahanna, University of Georgia, USA
Paul Benjamin Lowry, Virginia Tech, USA

Jenny Preece, University of Maryland, USA
Gavriel Salvendy, University of Central Florida, USA
Ben Shneiderman, University of Maryland, USA
Joe Valacich, University of Arizona, USA
Jane Webster, Queen's University, Canada
K.K. Wei, Singapore Institute of Management, Singapore
Ping Zhang, Syracuse University, USA

Senior Editor Board

Torkil Clemmensen, Copenhagen Business School, Denmark
Fred Davis, Texas Tech University, USA
Gert-Jan de Vreede, University of South Florida, USA
Soussan Djasasbi, Worcester Polytechnic Institute, USA
Traci Hess, University of Massachusetts Amherst, USA
Shuk Ying (Susanna) Ho, Australian National University, Australia
Matthew Jensen, University of Oklahoma, USA
Atreyi Kankanhalli, National University of Singapore, Singapore
Jinwoo Kim, Yonsei University, Korea
Eleanor Loiacono, College of William & Mary, USA
Anne Massey, University of Massachusetts Amherst, USA
Gregory D. Moody, University of Nevada Las Vegas, USA

Lorne Olfman, Claremont Graduate University, USA
Stacie Petter, Baylor University, USA
Choon Ling Sia, City University of Hong Kong, Hong Kong SAR
Heshan Sun, University of Oklahoma, USA
Kar Yan Tam, Hong Kong U. of Science & Technology, Hong Kong SAR
Chee-Wee Tan, Copenhagen Business School, Denmark
Dov Te'eni, Tel-Aviv University, Israel
Jason Thatcher, Temple University, USA
Noam Tractinsky, Ben-Gurion University of the Negev, Israel
Viswanath Venkatesh, University of Arkansas, USA
Mun Yi, Korea Advanced Institute of Science & Technology, Korea
Dongsong Zhang, University of North Carolina Charlotte, USA

Editorial Board

Miguel Aguirre-Urreta, Florida International University, USA
Michel Avital, Copenhagen Business School, Denmark
Gaurav Bansal, University of Wisconsin-Green Bay, USA
Ricardo Buettner, Aalen University, Germany
Langtao Chen, Missouri University of Science and Technology, USA
Christy M.K. Cheung, Hong Kong Baptist University, Hong Kong SAR
Tsai-Hsin Chu, National Chiayi University, Taiwan
Cecil Chua, Missouri University of Science and Technology, USA
Constantinos Coursaris, HEC Montreal, Canada
Michael Davern, University of Melbourne, Australia
Carina de Villiers, University of Pretoria, South Africa
Gurpreet Dhillon, University of North Carolina at Greensboro, USA
Alexandra Durcikova, University of Oklahoma, USA
Andreas Eckhardt, University of Innsbruck, Austria
Brenda Eschenbrenner, University of Nebraska at Kearney, USA
Xiaowen Fang, DePaul University, USA
James Gaskin, Brigham Young University, USA
Matt Germonprez, University of Nebraska at Omaha, USA
Jennifer Gerow, Virginia Military Institute, USA
Suparna Goswami, Technische U.München, Germany
Camille Grange, HEC Montreal, Canada
Juho Harami, Tampere University, Finland
Khaled Hassanein, McMaster University, Canada
Milena Head, McMaster University, Canada
Netta Iivari, Oulu University, Finland
Zhenhui Jack Jiang, University of Hong Kong, Hong Kong SAR
Richard Johnson, Washington State University, USA
Weiling Ke, Southern University of Science and Technology, China

Sherrie Komiak, Memorial U. of Newfoundland, Canada
Yi-Cheng Ku, Fu Chen Catholic University, Taiwan
Na Li, Baker College, USA
Yuan Li, University of Tennessee, USA
Ji-Ye Mao, Renmin University, China
Scott McCoy, College of William and Mary, USA
Tom Meservy, Brigham Young University, USA
Stefan Morana, Saarland University, Germany
Robert F. Otondo, Mississippi State University, USA
Lingyun Qiu, Peking University, China
Sheizaf Rafaeli, University of Haifa, Israel
Rene Riedl, Johannes Kepler University Linz, Austria
Lionel Robert, University of Michigan, USA
Khawaja Saeed, Wichita State University, USA
Shu Schiller, Wright State University, USA
Christoph Schneider, IESE Business School, Spain
Theresa Shaft, University of Oklahoma, USA
Stefan Smolnik, University of Hagen, Germany
Jeff Stanton, Syracuse University, USA
Chee-Wee Tan, Copenhagen Business School, Denmark
Horst Treiblmaier, Modul University Vienna, Austria
Ozgur Turetken, Ryerson University, Canada
Wietske van Osch, HEC Montreal, Canada
Weiquan Wang, City University of Hong Kong, Hong Kong SAR
Dezhi Wu, University of South Carolina, USA
Fahri Yetim, FOM U. of Appl. Sci., Germany
Cheng Zhang, Fudan University, China
Meiyun Zuo, Renmin University, China

Managing Editor

Gregory D. Moody, University of Nevada Las Vegas, USA