Keywords: Agent-based simulation, traffic simulation, cognitive abilities

Abstract
Currently, there are a large number of traffic simulation frameworks available. Usually, the principle of these simulation frameworks is to move vehicles from some location A to some location B as the result of different equations of motion or fluid dynamics. As it is, reality is much more complex because what actually happens between A and B is highly determined by the preferences and attitudes of the driver himself and not just by any mathematical formalism. In this work, we introduce an approach which considers a driver’s mind within large scale traffic simulations. For this purpose we describe a BDI based conceptualization of a driver and extend common simulation topologies with service oriented concepts.

1. INTRODUCTION
Powerful computational models as well as quick development in computer technologies have gained the field of traffic simulation a rapid upswing over the last years. Currently, there are a large number of traffic simulators available. To provide some structure to this wide field, traffic simulators are usually classified according to the level of detail with which they represent the system to be studied [10]. Commonly one distinguishes between a microscopic-, a mesoscopic- or a macroscopic model.

1.1. Motivation
Despite the wide range of available simulation frameworks, most products share the fact, that the vehicle simulation is done in a deterministic fashion. Usually, the simulated vehicles are moved from a location A to a location B as a result of different equations of motion or fluid dynamics. As it is, reality is much more complex, because what actually happens on the road is not only determined by physics of motion, but also by the perception and attitudes of the drivers. Yet, as a matter of fact, most state-of-the-art simulation frameworks do not provide concepts to reflect the driver’s attitudes and preferences within a computer-aided simulation. Only few approaches try to bridge this gap [3], [6], [14], [17]. Usually, these approaches apply agent oriented concepts and realize individual behavior by implementing perception, cognition, reasoning and goal-oriented strategies. Although good results have been achieved by this technique, the outcome of these approaches is limited to varying driving styles of the simulated vehicles, such as aggressive or cautious. Yet, a more comprehensive, “strategic” consideration is mostly missing. A driver with a high affinity for public transport for instance might change his means of transportation when confronted with a traffic jam near a metro station and available parking. This aspect does not affect the driving process per se, but influences the traffic situation a fortiori. In the case of electric mobility, the necessity for strategic considerations is even more apparent. Here, drivers have to deal with the range restriction of electric vehicles and develop strategies to accomplish their regular errands, despite these restrictions. We have discussed this problem already [12] and also formulated a first concept for drivers with planning abilities [11].

In this paper we refine our so far simulation model [11] (see Section 2.) which allows for the specification of preferences and attitudes of drivers on a strategic level. Following a user-centric approach, the possibilities of movement are not limited to moving vehicles but are extended to walking and other transportation facilities. Further services that are being provided in the context of transportation, such as car parks and fuel stations, can be added to the simulation model. We start by explaining the model we have specified for the driver and point out the additional requirements for the topology model which are necessary to make this approach work. Subsequently, we exemplarily model different traits of acceptance towards a park-and-ride service combination and demonstrate the functionality of our approach by presenting the simulation results of this example (see Section 3.). We proceed by comparing our approach to related work and show how our work differs from common approaches (see Section 4.). Finally, we discuss the practical experiences we have made so far with our approach and wrap up with a conclusion (see Section 5.).

2. APPROACH
We have to address two topics. We start by defining a model for the environment which is able to influence the behavior of a driver by certain stimuli. Then we define the behavioral model for the driver, which comprehends the stimuli of the environment and is able to generate the driver’s action.
2.1. The Service City

As a matter of fact, traffic simulations move cars on some infrastructural topology. The difference between our approach and approaches which do not consider human aspects is that a driver is able to perceive and interact with this topology by making use of certain *Infrastructural Features* which may either support the driver in achieving his goals or influence his strategy in achieving them. Before we proceed we provide a definition for the term: *Infrastructural Feature*.

*An Infrastructural Feature can be everything which is able to fulfill a desire (or parts of it) of a person at a certain location of an infrastructure.*

Admittedly, this is rather high level. An example will help to show the practical implications. Consider public transports for instance. Public transports provide a service at many places of an infrastructure and supports a person’s desire to reach a certain location. Another example are car parks. Located at some location they provide service for any driver who wants to park his vehicle. According to our definition, *Infrastructural Features* are not necessarily related to traffic, but can also be interpreted as: Shop, ATM machine, restaurant, takeaway, telephone booth and many more, since all of them are part of a municipal infrastructure and capable to influence a driver’s behavior, depending on his mental state and his preferences.

Based on the our definition of *Infrastructural Features*, it is nearly impossible to provide a complete model for any larger city; this is not our intention. Our objective is to provide a uniform way for the specification of these features in order allow for easy, custom definitions. Above, we mentioned that features are able to fulfill desires, or in other words, they provide a certain service for anyone using the feature. In fact, the resemblance to the service metaphor is so close, that we apply this principle and formally define an *Infrastructural Feature* as a service. We start with the requirements. Services define preconditions to ensure a successful execution. *Infrastructural Features* imply a similar requirement. Consider a fuel station for instance. In order to use a fuel station, one’s car has to run with a fuel type which is sold at the fuel station. For a driver *d*, and a fuel service *f*, this condition can be defined as follows:

\[
\exists {\text{fuelType}} \in f.getFuelTypes() : \quad d.getCar().getFuelType() = \text{fuelType}
\]

At this point, we establish, that the definition of an *Infrastructural Feature* as a service comprises the specification of its preconditions. Alongside the preconditions, the usage of each *Infrastructural Feature* serves a purpose. In the case of our exemplary fuel service, this purpose can be defined as follows:

\[
d.getCar().fuelLevel() = d.getCar().maxCapacity()
\]

In the service metaphor, purposes are termed “effects”. In compliance with the service metaphor, we will from now on refer to this purpose as “effect”. Next, we have to consider the location issue. *Infrastructural Features* are located at a certain location. Hence, we defined our services to feature a location attribute. We also provide a scope which we can later use to determine if a driver is able to perceive the service. In real life, this scope can be interpreted as optical range, but also as knowledge of the service. Finally, we define a duration method for the service, which is used by the simulation engine to compute the required time for a service execution.

To sum up, an *Infrastructural Feature* consists of its preconditions, its effects, information about the execution duration as well as a geographic location and a scope. The usage of a service has to fulfill the following statements:

\[
\forall s \in \text{Services}, \forall d \in \text{Drivers} : \\
\quad \forall p \in s.getPreconditions() : \\
\quad \quad d.canUse(s) \rightarrow p \quad (1)
\]

\[
\forall s \in \text{Services}, \forall d \in \text{Drivers} : \\
\quad \forall e \in s.getEffects() : \\
\quad \quad s.execute(d) \rightarrow e \quad (2)
\]

We can now define a psychological behavior model for the drivers which is able to consider the stimuli given by an infrastructure as described above. We do this in accordance with the *Belief-Desire-Intention (BDI)* paradigm [16].

2.2. The BDI Driver

Following the definition of Wooldridge and Jennings [18], a driver is an agent that is able to act autonomous, reactive, pro-active and socially competent to changes in its environment. We apply an agent oriented view and follow a popular model for the conceptualization of human behavior: *The BDI model* [16]. This approach provides us with a specification for our implementation and a validation of the agent’s behavior. We can implement critical processes in terms of several distinct modules, each one realizing a particular “phase” of the agent’s overall behavior. The operation principle and behavior phases of our BDI agent are illustrated in Figure 1.

Altogether, the agent’s execution cycle comprises four phases. Triggered by his perception (1), the agent starts with the *Belief Revision*, in which he updates (3) his belief base

\[1\] More complex, cross-linked services such as a metro system, may also feature several locations. We will provide more details in Section 3.
with his current perception (2a) and his so far beliefs (2b). With his updated new belief base (4a) and his current intentions (4b), the agent updates (5) his current set of goals in the Generate Options phase. In combination with the agent’s plans and his current intentions, the new set of goals constitutes the input (6a, 6b, 6c) for the Filter phase, which generates (7) a new set of intentions. Finally, the new set of intentions is used (8) to determine the agent’s Actuation, by which he influences (9) his environment. In more detail, the different phases look as follows:

**Prerequisites:** In order to make this approach work, we define a set of required basic capabilities (or, using the BDI terminology “plans”) for each agent. Each driver has to provide a walk and a drive plan, which the agent uses to either walk or drive from a location A to a location B. In addition, we need a whole set of further information on the usage of both plans, hence we bundle each plan into respectively one, so called Plan Object, which we additionally furnish with information on the plan’s preconditions, its effects and a function which returns the duration for an intended trip. We will not go into detail regarding the implementation of the walk and drive capabilities, since the state of the art provides many solutions for this topic (for a comprehensive survey refer to Section 4.). In short, we are using A* routing for our application and calculate the required duration by accessing the max-speed fields of our simulation topology, which is based on the Open Streetmap [15] (OSM) framework. Regarding the preconditions, we demand that the agent is in possession of a car in order to have driving capability. In comparison, the walk capability requires the agent to be on foot. We define the effects of both plans to move an agent from his current location to the desired target within the period of time which is returned by the duration function of the respective Plan Object.

2

**Option Generation:** In this phase, the agent determines if he is able to make use of any of the perceived services, by evaluating the service’s preconditions. Preconditions can be defined such that a third party reasoner can be used to check them, or one can implement the check within the system. A failed precondition check will not change the state of the agent, but in case of a successful evaluation, the desire to make use of the service will be stored in the form of a goal within the goal base of the agent. In this goal base we differentiate between one superior goal, which expresses an agent’s main objective to reach a certain location and several (sub-)goals which emerge dynamically as an agent’s desire to make use of an infrastructural service. The agent is compelled to his superior goal, exclusively. Other goals can be considered as alternative options in reaching his ultimate target. Whether or not an alternative option is chosen is determined by the agent’s attitudes and preferences. This decision is done within the subsequent Filter phase.

**Filter:** In this phase, the agent retrieves his goals from the goal base and tries to find ways to achieve them. As mentioned above, we distinguish between two types of goals here. One goal is superior to any other goals and expresses the agent’s main objective to reach a certain location and several (sub-)goals which emerge dynamically as an agent’s desire to make use of an infrastructural service. The agent is compelled to his superior goal, exclusively. Other goals can be considered as alternative options in reaching his ultimate target. Whether or not an alternative option is chosen is determined by the agent’s attitudes and preferences. This decision is done within the subsequent Filter phase.
happens just after a superior goal has been placed within an agent’s goal-base is straightforward. By considering his Plan Objects, the agent determines if he uses either his drive- or his walk capability. Since the first option requires less time for reaching the goal, an execution of the drive method is likely to be chosen. When a second goal emerges, the original strategy gets competition. The filter function will now compute a strategy to make use of the service. Usually, this strategy starts with a drive to the location of the service. Subsequently, the filter function assumes an execution of the service, and computes a strategy to achieve the superior goal (just as it did initially), based on the state which is defined by the service’s effects. Again, this is quite a simple case, since it is of course possible that an agent is located within the scope of more than one service. In case of more available services, each possible composition of services is evaluated and temporarily stored in combination with the measured cost for each strategy. This measurement process actually represents the agent’s preference profile, because here it is decided which criteria an agent selects from many proposed strategies. Our approach allows for a creative implementation of this process, since the programmer is able to access the duration methods of the services and Plan Objects involved in the strategy, and can therefore define a time based selection, or custom acceptances for each strategy element. For our example (see Section 3.), we applied a merged approach and artificially lengthened the required time for a particular service according to a percentage value, which we defined to express the agent’s acceptance of this service. Once each possible strategy has been computed, the Filter phase will come to end and place the best strategy into the agent’s intention repository.

**Actuation:** In this phase, the computed strategy is executed. A strategy can only comprise the method execution of a Plan Object, or the execution of services. While we defined the implementation for the walk and drive methods earlier, we did not mention the implementation of a service’s execution method so far. Next to preconditions, effects and a duration method, a service specification has to implement the actual execution method. The result of this method is to alter the internal values and parameters of the service, the driver and the environment and to communicate any changes to the simulation engine. The simulation engine realizes these changes after the execution time of the service and the agent can start to perceive its environment, again.

### 3. BDI DRIVERS IN A SERVICE CITY

In order to clarify the concept, we proceed with an example which we also use to evaluate our work. We select two common Infrastructural Features: A car park and a metro station. We start this section by explaining the implementation of both services and proceed by explaining the required adjustments for the driver agents. Subsequently, we explain our simulation setup in detail and wrap up by illustrating the collected results.

**The Metro Station Service:** As described in the previous section, the integration of new services comprises the specification of the service’s preconditions. In order to make use of the metro service, we have to make sure that the driver is currently not located in a car.

\[
driver.getCar() = \emptyset
\]

Although we are dealing with a complex and cross-linked service, we apply a rather simple implementation for the service’s effects. Later we place three instances of the metro service into the simulation topology. While the different instances are located at different positions, each service moves the executing driver to the same target. To this end, we create a metro network with three entrance points, but only one exit point. The only effect of the service is to move a driver to the target position \( p \).

\[
driver.getPosition() = p
\]

Due to the different access points of the metro service, we define the service’s duration method to return the required time for the metro ride from the location of the service instance to the universal target location.

**The Car-Park Service:** For the car park service, we assume that each driver who wants to make use of the service is riding a vehicle and that the service provides enough vacant capacity, which we initially set on 2,000 parking lots for each service instance.

\[
driver.getCar() \neq \emptyset
\]

\[
parkingService.getCurrentCapacity() > 0
\]

For the effects, we define that the driver is no longer be possession of his car and that the capacity before the service execution \( c \) is decreased by one.

\[
driver.getCar() = \emptyset
\]

\[
parkingService.getCurrentCapacity() = c - 1
\]

Finally, we implement the duration method of the car park service to return one minute.
The Drivers: For the drivers, we manipulate the filter module of our agents to mimic different acceptances towards the metro service. We then perform an evaluation of the proposed strategies by accumulating the regular durations for any walk, drive and parking service usage and artificially lengthen the required time for the metro service inversely proportional to the driver’s acceptance towards the service. We define the acceptance by means of an attribute of the driver. The higher a driver’s acceptance for the service, the lower the chance for artificially increased costs and the higher the probability for a service usage.

3.1. The Example

We start by integrating the designed services into a simulation topology. For each parking service, we defined a total capacity of 2,000 parking lots. We further integrated three metro services into the topology, each one at a different location but altogether with the same target. We specified a total amount of 10,000 cars for each simulation run and defined an area of potential start locations and an area of potential target locations. With the selected arrangement for these start and target pools we made sure the cars pass the influence of our services. The entire simulation setup is illustrated in Figure 2.

![Figure 2: The simulation scenario, including three metro services (white), thirteen parking services (green), as well as markers for start and target areas (dark green and red) of the cars.](image)

In total, we performed four simulations, respectively one for a 20%, 40%, 60% and 80% acceptance towards a potential use of the metro service. For each simulation, the drivers are located at random places within the start area and in possession of a car. Their calendar contains one event with an instruction to proceed to a random location within the target area. Once the simulation engine detects a scheduled event, a goal is generated and placed within the goal base of the agent. The agent now starts his journey. Since no other option is available (no service is perceived and the walk capability requires the driver to be on foot), the agent computes and executes a strategy involving his drive capability. Once a driver enters the visibility scope of a service, further concurrent strategies will be proposed. In case of entering the scope of a parking lot, the driver computes a strategy to park his car and walk to the target location. Usually this option will fail because we apply a strategy selection which is based on the required time to the target and walking strategies tend to be highly expensive. In case of entering the scope of a metro station, any evaluation on making use of the service will fail, since we demand the driver to be on foot. Only when the car is within the scope of both services, the Filter module will be able to compute a valid strategy, involving a ride to the parking service, its execution, a walk to the metro service, its execution and a walk to the target location. Depending on the driver’s acceptance of the metro service, this optional strategy either replaces the original one or is rejected. We illustrate the results of the four simulation runs in Figure 3.

![Figure 3: Results of the four simulations, showing the parking service’s utilization in percentage values.](image)

The acceptance for the metro service increases from the top to the bottom, so that the topmost illustration features an acceptance of 20% and the lowermost illustration features an acceptance of 80%. Each illustration shows the capacity utilization of respectively one parking service by means of colored circles. Red circles represent utilizations beyond 90%, yellow circles represent utilizations beyond 50% and green circles represent utilizations below 50%. One can clearly see that different user profiles tend to influence the overall traffic situation differently. Where a low service acceptance results in a high utilization of the parking services within the target area, an increasing acceptance causes a migration of the
utilization peak, until –in case of an acceptance of 60% and more– it is not possible to make use of the first metro station, because its parking capabilities are exhausted. According to these results, we can observe that different user profiles influence traffic situations differently. The consideration of these parameters is thus able to increase the quality of simulation results.

4. RELATED WORK

Prior to our development, we performed a comprehensive survey on existing approaches in order to distinguish our work from others and to gain ideas for our own implementation. We started by examining popular and mostly commercial products. In particular, we tried to identify behavioral models for the simulated drivers which comprehend influences of the surrounding environment. We came to the conclusion, that commercial software provides no such feature and extended our survey to the academic domain. Here, we were able to identify many approaches with behavioral models for the simulated drivers. We recognized, that most works either implicitly or explicitly apply an agent-oriented view on the simulated vehicles. Yet, most conceptualizations were focused on the driving process per se and miss the more comprehensive consideration of how the superior goal of reaching a certain target location is actually realized. The vehicles of the examined approaches have been implemented to mimic different driving styles, such as aggressive or cautious. We were not able to identify individual strategic behavior, which derives alternative options and which is triggered by environmental influences. In the following, we present a selection of related approaches. Next to some popular commercial implementations we will introduce works with academic background which influenced our own work the most.

4.1. Commercial Traffic Simulations

Aimsun [2] is an integrated transport modeling software which is used to improve road infrastructure, reduce emissions, cut congestion and design urban environments for vehicles and pedestrians. Next to static and dynamic traffic assignment and mesoscopic simulations, the software supports further features, such as a creation of vehicle actuated signals, an introduction of public transport priority schemes, a definition of traffic management strategies with triggers and actions and many more. Cube [4] is a comprehensive, macroscopic transportation analysis system. The software offers a set of modules. While the base module constitutes the system interface, there are also extensions for multimodal micro simulation and for the forecasting of urban, regional, and long distance passenger travel demand. VISSIM [7] is a commercial simulation program for multi-modal traffic flow modeling. With a high level of detail it accurately simulates urban and highway traffic, including pedestrians, cyclists and motorized vehicles. In comparison to most available simulation software, VISSIM allows for modeling of the parking search process. OmniTRANS [13] is a commercial, multi-modal and multi-temporal simulation software which is used to analyze the interactions of transport modes in an urban context. The applied algorithms are based on mode chains and allow for the modeling of complex schemes such as park-and-ride, kiss-and-ride and bike-and-ride. Further, it is possible to define the geometric layout and signal plans of intersections. Examinations of multiple time configurations are supported as well.

4.2. Academic Background

The SUMO [9] simulation framework is an open-source, microscopic traffic simulation package, which has been developed as testbed for research matters. The software has been designed to handle large road networks and can be easily extended. A set of extension modules are already provided (i.e. online interaction, trace export or additional GUIs), while custom implementations are facilitated as well. Since SUMO provides a programming interface, we initially considered SUMO as simulation engine for our own implementation. At last, we decided against this idea since we feared the port-based communication which is required by the interface to be a potential bottleneck. However, for the future we will consider an evaluation of SUMO for our purpose. Both, Ehlert et al. [6] and Paruchuri et al. [14] describe similar approaches for modeling unorganized traffic in a traffic simulation. Paruchuri et al. further present a way to use micro and macro goals for the realization of different driving styles, such as aggressive, normal and cautious. Psychological traits, such as “confidence” and “rush” are implemented as well. The applied goal model inspired our own implementation and convinced us to use the BDI principle for our work. The SCANeR® II simulator [3] is an agent based traffic simulator and has been developed by the French automobile manufacturer Renault for ergonomics and advanced engineering studies, for research in road traffic, for human factor studies and for driver training. The software features microscopic simulation of interactively driven entities. Currently, vehicles, trucks, motorcycles, bikes, train, trams or pedestrians are supported. Each simulated object is able to either mimic “unique” or “risky” driving behavior. Since SCANeR® II is focused on the simulation of the driver, the application features a comprehensive model for the association between the driver and his means of transportation. The requirements for our application were similar and for this reason, we applied a similar implementation. Rigoli et al. [17] describe a mesoscopic traffic simulator, in which the simulated vehicles are implemented by using a behavioral multi-agent approach to model human drivers. In the simulation, the applied agents generate a qualitative description of the sensed environment, while an included reasoning module is used to generate a set
of possible driving options. The simulator features simulations on straight line highways with multiple road lanes and with no entry or exit lanes and comprises different environmental conditions, such as ice or fog. The option generation model in particular was highly interesting for our own work. In fact, we implemented a similar model which – instead of providing options for “short term” decisions – we designed to produce a set of alternative strategies.

4.3. Lineup

As mentioned in the introduction of this section, models in commercial traffic simulation software are the exception rather than the rule. Academic approaches provide according considerations but their “scope” is focused on short term decisions rather than involving strategic behavior. For this reason, the examined approaches disregard the bigger picture of what might influence a driver to rethink his strategy to reach a certain location. Yet, as we have demonstrated in the previous section, this exact behavior is a significant variable for traffic situations.

5. CONCLUSION

In this paper, we refined and implemented a recently presented concept [11] for behavioral aspects and traits of drivers in a computer-aided traffic simulation. We applied a service metaphor for the definition of infrastructural stimuli which are able to influence a driver on his journey and used an agent model for the driver, which is able to perceive the provided services of an infrastructure during his journeys. We did that in compliance with the BDI approach. To clarify our principle and for evaluation purposes, we defined an exemplary simulation setup, involving a parking- and a metro service and ran several simulations on this setup, each one with a different acceptance for the metro service. We demonstrated the functionality of our approach by presenting simulation results. Based on a survey on related work, we can say that none of the examined state-of-the-art products or approaches has the capability to consider strategic behavior, which evolves from infrastructural influences and a driver’s attitudes, although our example clarified the necessity for this kind of consideration.

Admittedly, in this paper we only gave little attention to the actual simulation engine, and instead put particular emphasis on the realization on the concept of the applied agents and the model of the infrastructure. The reason for that is that the simulation engine per se does not differ from contemporary approaches and that we see the contribution of our work elsewhere.

5.1. Experiences

In comparison to other traffic simulation frameworks, our concept does not apply a view on vehicles, but focuses on the driver himself for the simulation. This point of view allows us for flexible applications of our simulator, not least because it is easy to exchange the specific data for the vehicles. Information on the traffic situations are actually a byproduct of the simulation of actual persons. Currently, we make use of the flexibility of our approach in an industrial funded project, in which we co-operate with Volkswagen AG under the objective to evaluate Driver Assistance Systems [8].

Beside this project, we apply our approach in a second nationally founded work, in which we co-operate with another major car manufacturer and a national energy provider under the objective to examine usage patterns of charging infrastructures for electric vehicles. The modular concept once more fulfills its need and allows in this context for a high detailed implementation of a charging station service as it is used by electric vehicles. The implementation does not only consider single charging stations, but arranges them according to an original electricity network, including all grid-related constraints. Using this detailed implementation we currently try to predict the influences of e-mobility on regional energy grids.

Our approach of applying a BDI model provides a clear specification for our implementation. We can implement critical processes in terms of several distinct modules, each one realizing a particular phase of the agent’s overall behavior. Admittedly, we applied rather simple algorithms for this work. The Filter phase for example computes any possible permutation of actions and is neither effective nor guarantees success. In fact, we are discussing a more sophisticated mechanism already.

In our approach we demonstrate that the appliance of a BDI architecture for drivers in a traffic simulation allows to define influences of an infrastructure on a driver exclusively by coding these influences and not necessarily by adapting the agent. For behavioral variations however, the underlying BDI architecture facilitates adaption through its modular assembly.

5.2. Future Work

In the future we will develop different extensions to the simulation which provide a wider range of acting options to the agents, as well as more complex BDI capabilities. In the following we describe selected intentions.

BDI agents do not only have to be able to recognize external services but also other agents by perception. In doing so, the agents can negotiate with each other and check whether they share goals which are (almost) identical. This offers the possibility to co-operate and share resources like vehicles and to minimize the cost function of each agent (Assuming the cost function reflects monetary and ecological aspects, respectively). With an according feature, the simulation framework can be used for the simulation and optimization of carsharing projects. Also, the actual infrastructural utilization
should be taken into consideration when choosing a transportation service. For example, if there is a traffic jam within the route to destination the transportation by vehicle should become rated worse. This could influence the agent not using the vehicle but public transportation. Finally, we intend to develop a concept for so called Regional Conditions, which directly affect drivers which are located within. We will use this concept to integrate (regional) weather conditions, since it is our belief, that human traffic behavior (and also consumption of electric vehicles) is highly constrained by those.

6. ACKNOWLEDGMENTS

This work is partially funded by the Federal Ministry for the Environment, Nature Conservation and Nuclear Safety under the funding reference numbers 16EM0004 and 16EM0071.

REFERENCES


