sustainability

MDPI

Article

Spatio-Semantic Road Space Modeling for Vehicle–Pedestrian Simulation to Test Automated Driving Systems

Benedikt Schwab [1,2,*], Christof Beil [2,*] and Thomas H. Kolbe [2]

[1] AUDI AG, Auto-Union-Straße 1, 85045 Ingolstadt, Germany
[2] Geoinformatics, Technical University of Munich, Arcisstrasse 21, 80333 Munich, Germany; thomas.kolbe@tum.de
* Correspondence: benedikt.schwab@audi.de (B.S.); christof.beil@tum.de (C.B.); Tel.: +49-841-89-985642 (B.S.)

Received: 30 March 2020; Accepted: 28 April 2020; Published: 7 May 2020

check for updates

Abstract: Automated driving technologies offer the opportunity to substantially reduce the number of road accidents and fatalities. This requires the development of systems that can handle traffic scenarios more reliable than the human driver. The extreme number of traffic scenarios, though, causes enormous challenges in testing and proving the correct system functioning. Due to its efficiency and reproducibility, the test procedure will involve environment simulations to which the system under test is exposed. A combination of traffic, driving and Vulnerable Road User (VRU) simulation is therefore required for a holistic environment simulation. Since these simulators have different requirements and support various formats, a concept for integrated spatio-semantic road space modeling is proposed in this paper. For this purpose, the established standard OpenDRIVE, which describes road networks with their topology for submicroscopic driving simulation and HD maps, is combined with the internationally used semantic 3D city model standard CityGML. Both standards complement each other, and their combination opens the potentials of both application domains—automotive and 3D GIS. As a result, existing HD maps can now be used by model processing tools, enabling their transformation to the target formats of the respective simulators. Based on this, we demonstrate a distributed environment simulation with the submicroscopic driving simulator Virtual Test Drive and the pedestrian simulator MomenTUM at a sensitive crossing in the city of Ingolstadt. Both simulators are coupled at runtime and the architecture supports the integration of automated driving functions.

Keywords: automated driving; autonomous driving; testing; environment simulation; road space; street space; OpenDRIVE; CityGML; driving simulation; pedestrian behavior modeling

1. Introduction

The automation of the driving task offers the potential to substantially transform the mobility of the future. In particular, higher automation levels can enable the provision of mobility as a service and thereby significantly reduce costs for the customer by sharing capital as well as operating expenses of a vehicle [1,2]. The technology can further contribute to more convenient rides and provide new mobility freedoms for children, seniors and the disabled [3]. Moreover, it is expected that intelligent communication between the vehicles (V2V) and the infrastructure (V2I) will enhance the overall traffic efficiency and advance real-time traffic management [2]. A central benefit of automated driving is clearly the increased road safety and the subsequent reduction in road fatalities. However, increasing road safety is not only a mere benefit but rather a moral imperative. According to the German ethics commission, the approval of automated driving systems is only justifiable if they promise at least a reduction of harm in comparison to the human driving performance [4]. Moreover, the safety has a profound impact on the societal acceptance of the technology [5,6].

For this reason, ensuring the correct system functioning constitutes a pivotal task for the development and homologation of automated driving systems. This includes, for instance, that automated vehicles must be capable of withstanding cyber threats to preserve the passengers' integrity [2,7]. The extremely high number of environmental states to which an automated driving system can be exposed to represents a major development challenge for the validation and verification of the system [8]. Here, it must be ensured that the developed systems can adequately cope with all possibly occurring driving scenarios. This difficulty is further intensified in urban contexts when multiple VRUs are involved. As the number of traffic scenarios is exceptionally high, real test drives are neither economical nor feasible [9]. Due to its variability, reproducibility and efficiency, the simulation of the vehicle environment plays an inevitable role in the testing of automated driving systems. At the same time, system tests based on environment simulations are only effective if the models applied are valid and the gap to reality is known. The testing of the components is usually carried out in several integration steps along the V-model applying the Everything-in-the-Loop (XiL) approach. Here, the component to be tested is connected to a control loop that reproduces the interactions with other components as realistically as possible [10]. After the implemented function has been assessed using Software-in-the-Loop (SiL) tests, the next step would be to test this function on production-like hardware using Hardware-in-the-Loop (HiL) and later Vehicle-in-the-Loop (ViL) tests [11–13]. The general aim is to identify and eliminate defects as early as possible during the development process, for which the environment simulation of the component under test is a decisive prerequisite.

An environment simulation therefore requires validated models of the surrounding vehicles, VRUs and parts of the ego-vehicle. However, there is presently no testing framework that can satisfy all the requirements [14]. The linking of individual existing tools to a holistic and distributed test framework can thus represent a solution and is currently being pursued [15–18]. In the context of environment simulation not only the dynamic objects must be exchanged at runtime, but also the static environment models must be available for all individual simulators. However, the simulators pose partly different requirements to the road space models. This is particularly the case for traffic, driving and VRU simulation, which are necessary for a holistic environment simulation to test automated driving functions and thus require a consistent description of the road space.

1.1. Research Questions and Hypothesis

The objective of this work is to develop a concept for modeling spatio-semantic road spaces, which is tailored towards testing automated driving functions. The concept to be investigated shall not involve modifications to existing standards and simulators to avoid resulting incompatibilities and adaptation efforts. Therefore, the following research questions and hypothesis are formulated:

Question 1. *Which requirements are imposed by the environment simulation on the modeling of road spaces to test automated driving functions?*

Question 2. *How can road spaces be conceptually modeled to satisfy the identified requirements while avoiding the necessity of standard modifications and their associated incompatibilities?*

Hypothesis 1. *The duality of the standards OpenDRIVE and CityGML provides the modeling capabilities to meet all identified requirements.*

1.2. Organization of the Paper

The paper is divided into five sections, with Section 1 introducing the testing challenges and discussing the implications for the modeling of road spaces. Section 2 exploratively identifies the requirements formulated in Question 1 by examining the modeling of traffic, driving and VRUs individually. Based on the combined requirements a road space modeling concept is developed in Section 3 with respect to Question 2. The proposed concept combines the road network standard OpenDRIVE [19] with the semantic 3D city model standard CityGML [20,21]. To test Hypothesis 1, a distributed real-time simulation is conducted for an urban crossing as a proof of concept. Therefore, the spatio-semantic road space models are prepared for the submicroscopic driving simulator Virtual Test Drive (VTD) [10,22] and the pedestrian simulation framework MomenTUM [23,24]. The research results and the concept's potentials are discussed in Section 4. Finally, the conclusions and research limitations are drawn in Section 5.

2. Literature and Requirements Review

To test automated driving functions, the traffic scenarios are described in a machine-readable manner and the function under test is exposed to these scenarios using an environment simulation [25–27]. Furthermore, exploratory environment simulations can help to identify critical scenarios for the function under development during early development stages, when requirements are not yet fully defined [28]. Therefore, the components of a traffic scenario have been structured into five layers, which are depicted in Figure 1. The first layer describes the geometrical layout of the road including its topology and markings. Traffic infrastructural elements, such as barriers, traffic lights and traffic signs, are located at the second layer. The third layer describes temporal changes that can be caused by construction works, for example. The fourth layer essentially comprises moving objects, such as other motorized vehicles as well as VRUs, and relations to other objects or layers. The latter includes the driving maneuver "approaching another vehicle" and "approaching a stop line". Weather conditions and lighting situations are located at layer five. This layer structure is based on Schuldt's four-layer model, which was extended by a fifth [11,29]. A spatio-semantic road space model has to comprise all stationary elements of a scenario (layer 1–4) and contain all necessary topological as well as semantic information required for the simulation of the dynamic elements (layer 4 and 5).

To test automated driving systems in virtual environments, the road space models and the processing thereof must comply with a range of boundary conditions and requirements. For this purpose, the required Level of Detail (LoD) of the simulation needs to be addressed first. Traffic modeling can generally be classified into four LoDs [30]. *Macroscopic* models aggregate the individual road users by describing them as a flow with characteristics like density and velocity. Similarly, *mesoscopic* models do not differentiate between individual vehicles, but model, for example, the influence of individual behavioral parameters on traffic flow. In contrast, *microscopic* models describe the movement at the level of individual vehicles. This includes car-following models, which are parameterized by driver and vehicle characteristics. At the highest LoD, the *submicroscopic* driving simulation, perception and decision processes of the driver are modeled as well as the functioning of the vehicle's subparts. For example, steering system, powertrain, tires, brakes and sensors are modeled at this level [30]. To test automated driving systems within simulated traffic scenarios, a simulation on a submicroscopic level is required in which the agents can move continuously in space. Furthermore, HiL tests should be supported, which subsequently require a real-time environment simulation of the system under test. Since the simulation of the ego-vehicle requires a different LoD than distant oncoming traffic, there exist multi-resolution approaches that couple submicroscopic driving simulation with microscopic traffic simulation [16,17]. Concluding, the requirements for road space models are primarily derived from the fields of submicroscopic driving simulation, microscopic traffic simulation and VRU simulation as well as the interface compatibility to established test systems for automated driving functions. More general requirement overviews are provided by Schwab and Kolbe [31] and Richter et al. [32].

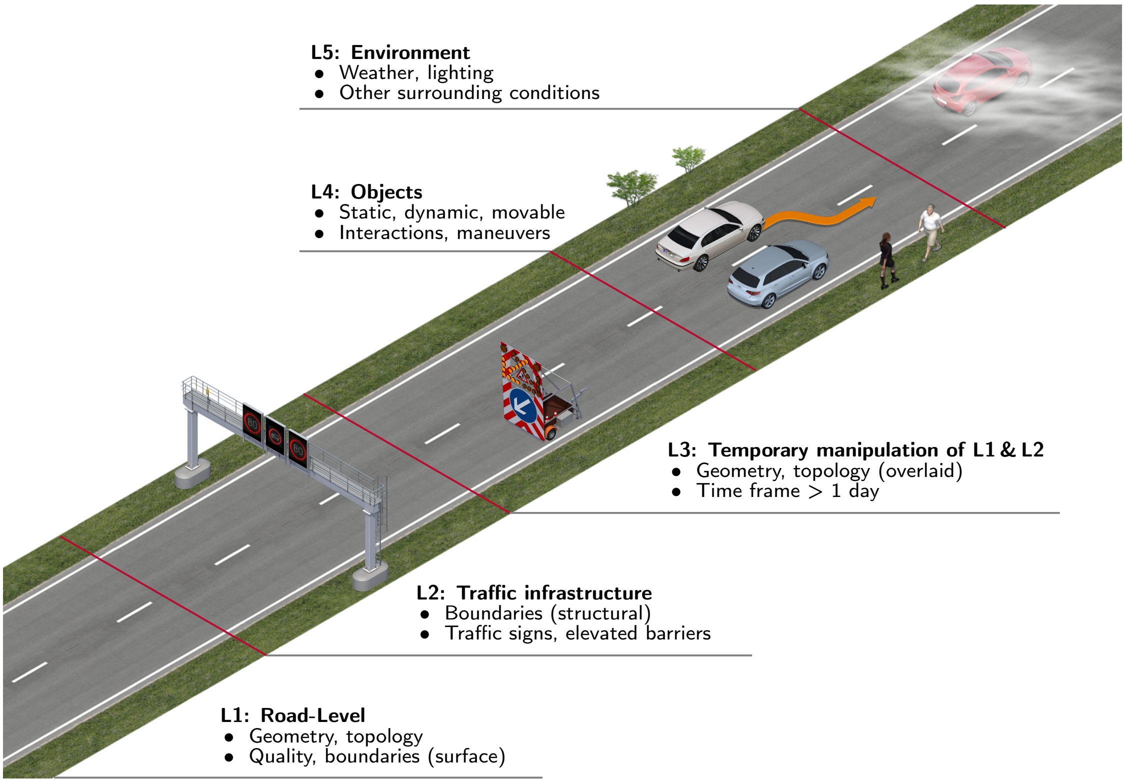

L5: Environment
• Weather, lighting
• Other surrounding conditions
L4: Objects
• Static, dynamic, movable
• Interactions, maneuvers
L3: Temporary manipulation of L1 & L2
• Geometry, topology (overlaid)
• Time frame > 1 day
L2: Traffic infrastructure
• Boundaries (structural)
• Traffic signs, elevated barriers
L1: Road-Level
• Geometry, topology
• Quality, boundaries (surface)

Figure 1. Five-layer model for a systematic description of traffic scenarios proposed by Bagschik et al. [29]. The graphical representation was inspired by [26].

2.1. Transport Modeling

The demand for traffic is driven by people pursuing activities, such as work and leisure, at different locations. To realistically simulate the surrounding traffic of the ego-vehicle, the activities carried out by inhabitants of an area must be examined over time and space. Therefore, road space models should be capable of serving as an information base to simulate intermodal traffic microscopically. The general structure for modeling transport stems from the 1960s and is known as the classic transport model [33]. While the components have been improved considerably over time, the overall structure has vastly remained the same and is visualized in Figure 2. As a starting point, the area to be simulated is divided into zones and statistical data representing socioeconomic and demographic characteristics are collected for the zones. The first stage in the classic transport model is called *trip generation* and involves the estimation of generated as well as attracted trips by each zone [33]. Although zones constitute an aggregation, it can be beneficial if the modeling is conducted at the level of individual buildings. This enables the assignment of granularly available information like building type or shop opening hours. Influencing factors for trip generation include income, car ownership, family size, value of land and residential density for personal trip productions as well as roofed space for commercial and industrial services. Factors for freight trips include number of employees, number of sales, roofed and total area of firms [33]. Hence, road space models must support the enrichment of custom variables. Additionally, it should be possible to perform geospatial analysis on them, since factors like roofed area or residential density could be derived automatically. If statistical data from different sources is available in non-coherent segmentations, the statistics can be disaggregated on an individual building basis and then reassembled into desired segmentations through geospatial queries. Statistics such as number of inhabitants or workplaces are usually provided by the city administration at district level, while large companies may also maintain sociodemographic statistics with different segmentations. For example, methods exist for the disaggregation of countrywide population data [34] but also for energy demand estimation on individual building level for entire cities [35]. Consequently, the creation, management and analysis of road space models must be efficiently processable for extensive areas including large cities with surroundings. Furthermore, it must obviously be possible to use existing datasets on road networks as well as buildings and combine them into a common model of the area.

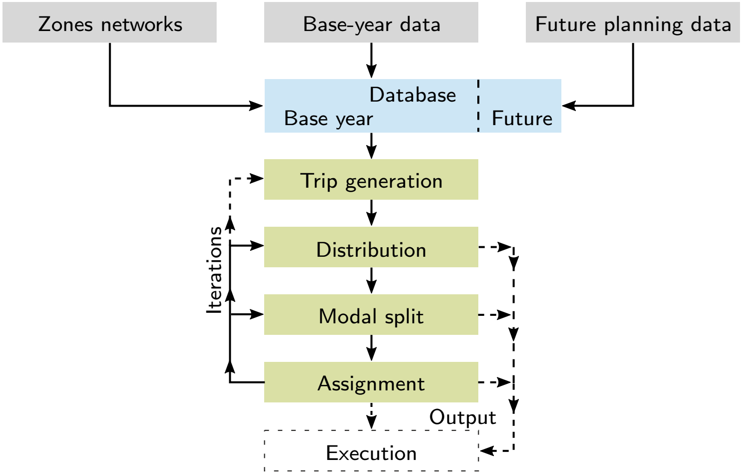


Figure 2. Classic four-stage transport model as shown in [33].

In the second stage named *trip distribution* the travel patterns between zones are modeled. For this purpose, statistics on trips are often compiled based on surveys, but there also exist data-driven approaches using mobile phone data, for instance [36]. *Modal split* constitutes the third stage and is concerned with the choice of the transport mode, such as bus, train or car. Here, discrete choice models can describe the decisions of individuals as a function of their socioeconomic characteristics [33]. In the last step *assignment*, the transport demand is assigned to the transport supply. The latter represents the cost model of the supply side and consists of a transport network with edges and their respective costs. The edge costs are influenced by distance, capacity and free-flow speed [33]. Floating Car Data (FCD) could be used here, e.g., to analyze speed-flow relationships and associate them with the road space model. To calibrate and validate the traffic simulation, real-world traffic observations are necessary. Traffic volumes can be measured by inductive loops and Internet of Things (IoT) sensors, but can also be analyzed by using the perceived surrounding traffic from FCD [2]. As traffic observations have a spatial reference, they should be representable in the road space model.

2.2. Driver and Vehicle Modeling

Since it is expected that the functionalities of automated vehicles will gradually increase over time, traffic with mixed levels of automation must be modeled. This implies that automated driving functions, the human driver and the vehicle with its driving dynamics must be simulated based on the road space model concept. Figure 3 gives a general overview of driver modeling, vehicle dynamics and their interaction with the environment.

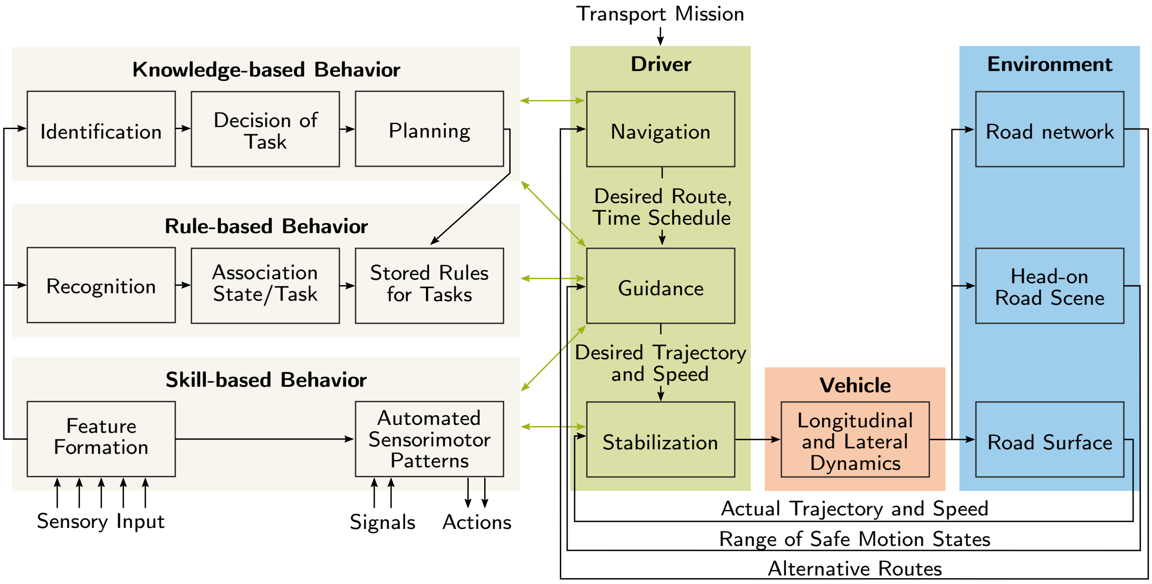


Figure 3. Overview of driver behavior modeling after Donges [37], whereas the left part constitutes the structuring of human target-oriented behavior after Rasmussen [38] and the right part represents the three-level driving task structure after Michon [39].

Michon structured the driving task into a three-level hierarchy consisting of the tasks navigation (strategical), guidance (tactical) and stabilization (operational), whereas the latter is also referred to as control. At the *navigation* level, the planning of the trip takes place, which includes the routing based on costs, risks and preferences [39]. This driving task level therefore represents the interface to the transport modeling discussed previously. Moreover, there is a feedback loop at this level enabling the driver to adjust the route in case of traffic congestions, for example. For road space modeling, this implies that the road topology must be represented and complementary tools should be available to support driver model implementations by means of existing routing algorithms. The *guidance* level involves the execution of driving maneuvers, such as turning left, following or overtaking. Driving maneuvers are usually restricted by the prevailing situation, such as surrounding traffic and obstacles [39]. Subsequently, the guidance level transfers the desired trajectory and speed to the stabilization level. During the *stabilization* task, the driver controls the position of the vehicle along the desired trajectory. To achieve this, the driver must ensure that deviations in a closed-loop control are corrected and stabilized by appropriate actions [37]. In this context, models have been proposed that address both tasks in conjunction, while other models separate between the guidance and stabilization task. Continuous models based on control system theory are applied for the simulation of these two driving tasks [37]. For example, Donges introduced a driver model in which the stabilization level is described by a compensatory closed-loop control, where the input signals consist of the lateral deviation, path curvature error as well as the heading angle error. An anticipatory open-loop control is applied at the guidance level, whereby the desired path curvature is used as the input signal. The anticipatory and compensatory control loops are additively combined and provide the steering wheel angle as an output signal of the driver model [40]. Furthermore, there also exist driver models that describe the perception process, for instance based on visual variables [41,42]. In summary, a road space model concept should therefore support the simulation of such driver models based on control system approaches.

To test specific functions or components of the automated driving system in a virtual environment, the remaining parts of the vehicle must be simulated. This includes the longitudinal, lateral and vertical dynamics of the vehicle as well as aerodynamics simulations. The road space model should hence support the simulation of the vehicle's brakes, tires, sensors. Moreover, it should be possible to simulate

Vehicle-to-Everything (V2X) communications, such as to traffic lights or other perception-enhancing concepts [43].

2.3. Pedestrian Behavior Modeling

In the context of vehicle-pedestrian-simulation, in particular models for the reconstruction of accidents have been developed [44,45]. While these are relevant for the design of the vehicle chassis, the testing of automated driving functions requires pedestrian behavior models for the description and generation of traffic scenarios. The modeling of pedestrian behavior is often separated into three interdependent levels, which were proposed by Hoogendoorn et al. [46]. Although no precise distinctions between the levels have been established in the literature and this concept has been further refined [47], the general structure helps to identify the requirements towards the capabilities of the road space models. The *strategic level* describes the activity planning and destination selection of pedestrians. This level indicates a methodical similarity to transport modeling, but the simulation of a concrete traffic scenario requires a higher spatial and semantic resolution. The activities that a pedestrian can potentially perform depend *inter alia* on the static environment. For instance, a park bench, a shop window and a building's door enable corresponding activities. At this level, psychologically inspired models of pedestrian interest in locations [48] and time-based origin-destination matrices can be applied [49]. For road space modeling this implies on the one hand that the selectable destinations should be available in a geometrically suitable representation. These are often areal representations in 2D. On the other hand, it should be possible to enrich the road space model with behavior model specific information, such as observed activity transition probabilities.

On the *tactical level*, concrete actions are modeled to reach the selected destination. This includes pedestrian decisions whether obstacles are passed to the left or to the right [46]. Pathfinding modeling is often achieved with graph-based [50] or cellular-automata-based approaches [51]. Hence, complementary tools should be available for the road space model, which support the generation of graphs as well as the division into cells. Furthermore, the weightings of the edges should be derivable based on spatio-semantic analyses.

Finally, the movement of the pedestrian along the chosen path is described on the *operational level* [46]. The acceleration of pedestrians is modeled together with the interaction to other pedestrians on this level. Here, established approaches also comprise cellular automata [52] and social force models [53]. The latter involves the superposition of forces, such as an attractive force to the selected destination and repulsive forces from obstacles. Additionally, there exist model variants that add attractive or repulsive forces to specific environment objects, such as crosswalk boundaries [54].

2.4. Modeling of Other VRUs

Besides pedestrians, there exist further road users who are particularly exposed to risks. This applies for instance to cyclists, whereby existing behavior model research is not as extensive as it is for motorized vehicle drivers or pedestrians. To model cyclist behavior at signalized intersections, Twaddle applies logistic regression at the tactical level to predict discrete choices like the response to a red signal or the left turn maneuver type [55]. The independent variables for the regression comprise geometric variables describing the intersection's layout and traffic variables including average traffic flows, for example. It should therefore be possible to represent bicycle lanes and lane topologies within the road space model. In addition, the independent variables include the existence of parking facilities, center islands and lane widths. The models are calibrated by video-based traffic observations and trajectory extractions [55]. For the cyclist modeling at the operational level, the NOMAD model is adapted by Twaddle, which was originally proposed for pedestrian behavior [56] and is similar to the social force approach. The acceleration vector is represented as norm and angle for this purpose, whereas separate models for velocity and direction are subsequently developed [55]. For research and simulation purposes the road space model should support the computation of trajectory planning approaches. As this paper focuses on vehicle-pedestrian-simulation, a further investigation has yet to be conducted for additional VRUs, such as e-scooter drivers, skateboarders or also wheelchair users.

2.5. Semantic 3D City Modeling

There are different approaches to modeling digital representations of the physical environment. Main distinctions between these types of models is the availability (or lack thereof) of semantic, geometric or topologic concepts for distinguishing or connecting separate thematic objects. Point clouds derived from mobile mapping, LiDAR or dense image matching for example often contain highly accurate geometric information on buildings, street space, vegetation or city furniture, however, do not contain semantic information on these objects individually. Automatic classification of point clouds is difficult and prone to error [57]. Meshes derived from such point clouds (and potentially colored with corresponding images) deliver visually appealing results but similar to point clouds do not contain semantic information on individual objects. The same applies to Virtual Reality (VR) models (COLLADA, X3D, FBX) commonly used for visualization purposes in computer graphics. In contrast to mentioned representations, information models (IM) are used for modeling real-world objects regarding thematic and functional aspects. Individual objects are geometrically segmented and often enriched with multiple attributes, thus containing semantic information important for many applications. Information on individual buildings, street space or other city objects in combination with attributes on function or usage open possibilities for various simulations and analyses such as pedestrian and vehicle simulation presented in this paper. Projects in the context of Computer Aided Design (CAD) or Building Information Modeling (BIM) are mostly performed in a local coordinate system with limited geographical extent and are generally created project-specific for planned or built constructions. In contrast, semantic 3D city models are typically available for entire areas, such as cities or even nations. Many of these datasets are also accessible as open data. The international Open Geospatial Consortium (OGC) standard CityGML is most commonly used for representing such semantic 3D city models, whereas further details are provided in Section 3.1.2.

2.6. Tools and Simulators

To test automated driving systems, tools must be used that enable the integration of an automated driving system and its functions. Since submicroscopic driving simulators are typically used to simulate the environment during test procedures, there exist established interfaces to the system under test. Software for this purpose include IPG CarMaker [58], VTD [22], DYNA4 [59] and CarSim [60]. Additionally, the open source driving simulator CARLA [61] should be mentioned, which is based on the Unreal Game Engine and is being used in an academic context. Consequently, a concept for road space modeling must support submicroscopic driving simulators. Furthermore, there exists a variety of solutions for microscopic traffic simulation. Among these are AIMSUN [62] and PTV Vissim [63], which is used by public authorities for traffic planning, for instance. An open source alternative constitutes the software "Simulation of Urban MObility (SUMO)", whereas the development is pursued by the German Aerospace Center [64]. For a more detailed traffic simulator comparison we refer to [65–67]. In the context of VRU simulation there exist predominantly pedestrian simulators. These include JuPedSim [68], MomenTUM [23], Menge [69], Vadere [70], PTV Viswalk [71] and crowd:it [72]. For a more detailed discussion on the different pedestrian simulators we refer to [23]. In general, modularity of the simulators is relevant during the development process, as the behavior models must be interchangeable and newly developed models must be integrable. For a road space modeling concept to be simulator-agnostic, open and well-established standards should be considered. Due to the heterogeneity of simulators and supported formats, standards should be preferred that can be converted into other formats by established software tools.

2.7. Research and Development

The development and validation of new road user models requires a certain flexibility and extensibility to evaluate and test novel approaches. This requirement should not only be met in the area of simulators, but also in the field of road space models. When researching behavior models, it should be possible to attach the class and function, but also generic variables to the objects of the road space model. The probability of pedestrians crossing at red should, for example, be assignable as attribute to the traffic light or the respective street crossing. For this purpose, it should be feasible for researchers to implement processing steps of spatio-semantic road space models using Graphical User Interface (GUI) applications. Once the information necessary for the novel behavior model has been determined in the form of additionally required classes and attributes, it should be possible to extend the data model. For the validation of behavioral models, the traffic of real road spaces is typically recorded with cameras or even with laser scans. Then the trajectories are often estimated by means of object detection algorithms and corrected by humans if necessary [55,73,74]. A surveyed and georeferenced road space model can support the estimation of the transformation matrices for cameras as well as lasers and enables the georeferencing of the reconstructed trajectories.

3. Road Space Modeling and Application-Specific Preparation

To the best of our knowledge, there currently exists no road space modeling standard that satisfies all the requirements discussed. Thus, the objective is to describe the road space by a combination of already established standards and to meet the requirements by transforming them into the respective application target formats. This approach offers the advantage that established standards are already usable within complementary tools and environment simulators do not require modifications. Furthermore, datasets are already available in established standards.

3.1. Selection of Modeling Standards

First and foremost, a concept for spatio-semantic road space models must be applicable for the environment simulation in XiL testing approaches. All submicroscopic driving simulators discussed in Section 2.6 support the import of the road network via the open standard OpenDRIVE. Moreover, the microscopic traffic simulators, SUMO and PTV Vissim, also include the functionality to import OpenDRIVE. As this standard is open and widely adopted in the vehicle-related simulation context, it constitutes a key prerequisite for modeling road spaces in a vendor-agnostic manner.

3.1.1. OpenDRIVE

The OpenDRIVE standard describes the logics of road networks and is based on the Extensible Markup Language (XML). It was initially designed by the company *Vires Simulationstechnologie GmbH* and transferred to the Association for Standardization of Automation and Measuring Systems (ASAM) in 2018. OpenDRIVE is presently being revised and extended for version 2.0 by the participating ASAM members, whereas version 1.6 from 2020 is currently valid [19]. Additionally, the OpenCRG standard is offered, which describes the road surface in high detail and thus enables *inter alia* tire and vibration simulations [75]. The OpenDRIVE standard mainly uses an inertial coordinate system according to ISO 8855 and a track coordinate system, which is defined along the reference line of a road. The reference line is described by means of concatenated geometrical elements, such as lines, clothoids, arcs and cubic polynomials in the XY plane. Similarly, the elevation profiles of roads are also modeled as a concatenation of cubic polynomials. The geometrical representation of road objects is defined in the track coordinate system and is therefore relative to the road's reference line. For example, the widths of lanes are described by polynomials, which in turn are relative to the lane reference line—a laterally shifted road reference line. Road objects, such as trees, walls and buildings, can be modeled by a limited set of parametric geometries like cylinders, cuboids and polyhedrons. The georeferencing of the road network is achieved by means of a proj4 string in the header, which enables OpenDRIVE datasets to be used as HD maps for real driving tests [19].

OpenDRIVE entails a conceptual complexity that is necessary to satisfy several requirements discussed in Section 2. This is predominantly caused by the modeling structure and the three nested coordinate systems. However, this complexity is not needed for the realization of other tasks and requirements. For example, explicit geometries for the scenery should be sufficient for the modeling of pedestrian behavior and reduce the implementation effort, since no transformation matrices and derivable curves are required. Here, the OpenDRIVE geometry model poses rather an impediment. Due to the fact that this standard is mostly automotive specific, there is only a very limited support of complementary tools and libraries in other domains. In the context of the discussed requirements OpenDRIVE lacks particularly the feature to extend its data model, the support by established transformation tools and the possibility to perform geospatial analyses on it. By combining OpenDRIVE and CityGML, however, these limitations can potentially be overcome.

3.1.2. CityGML

CityGML is an international standard of the OGC for representing and exchanging virtual city models implemented as an application schema of the Geography Markup Language (GML). The exchange format is based on XML and the currently valid version CityGML 2.0 was published in 2012 [21], while a new version of CityGML is to be finalized within the OGC CityGML Standards Working Group (SWG) soon [76].

CityGML provides concepts for modeling semantics, geometry, topology and visual appearance in multiple LoDs for several thematic components of cities and landscapes such as buildings, vegetation or transportation. The main class of the CityGML Transportation module used to represent streetspace objects is called TransportationComplex and can be thematically specialized into one of four classes called Road, Track, Railway or Square. While linear representations of TransportationComplexes are limited to LoD0 and areal representations are introduced starting from LoD1, thematically more detailed segmentations into individual TrafficAreas and AuxiliaryTrafficAreas are possible in LoD2-4. While TrafficAreas represent parts of a Transportation object that are intended to be used by traffic members such as a road surfaces or sidewalks, AuxiliaryTrafficAreas describe further elements such as kerbstones or green areas. Each Transportation object can also be specified with respective class, function or usage attributes defined in extensive code lists. These include the definition of sidewalks, crosswalks, driving lanes, parking areas or markings represented with gml:MultiSurface geometries. Spatial properties are represented by a subset of GML3's geometry model. Several extensions have been proposed to the CityGML Transportation model that will be part of CityGML 3.0 [77,78].

Closely linked to the Transportation module is the CityFurniture module used for modeling signs, traffic lights or other immovable objects such as buckets or lanterns. Vegetation objects can be represented using the Vegetation module and are either represented as volumetric PlantCovers or as single vegetation objects using the class SolitaryVegetationObject. These individual objects can be represented with arbitrary GML geometry or by an ImplicitGeometry, i.e. a prototypical geometric representation. Thematic as well as spatial aspects of buildings can be represented in multiple levels of detail, ranging from generalized outer shell models to highly accurate representations including individual windows and doors.

To allow the exchange and storage of objects and attributes not covered by any thematic module, Generic objects and attributes can be modeled. Thus, additional attributes relevant for traffic simulations such as risk metrics can easily be included. There is also a built-in mechanism for extending CityGML with additional concepts, classes or attributes not provided by default called Application Domain Extension (ADE) [79]. ADEs can be used to augment the data model by specifying either an XML schema definition file (.xsd) or via Unified Modeling Language (UML) [80]. Different thematic modules can be integrated within one 3D city model to be used for combined simulations and analyses. Detailed areal representations of city objects in combination with semantic information such as function or usage stored within CityGML features allow the creation of accurate simulation scenarios. CityGML uses GML geometries fully supported by Geographic Information System (GIS) and spatial databases and can also easily be transformed into other formats using software such as the Feature Manipulation Engine (FME) [81].

3.2. Modeling and Preparation Architecture

Due to their origin and conceptualization, OpenDRIVE and CityGML complement each other. Whereas OpenDRIVE follows a parametric and analytical modeling approach for geometries, CityGML offers resolved surface-based geometries. Both standards and the tools supporting them solve problems within their application domain—automotive and 3D-GIS. However, the duality of OpenDRIVE and CityGML can open the potentials of both domains. As it is not intended to modify the data models of the respective standards to maintain their tool support, a consistent representation of a real road space must be achieved in both standards. This duality approach and the model preparations for the environment simulators are shown in Figure 4.

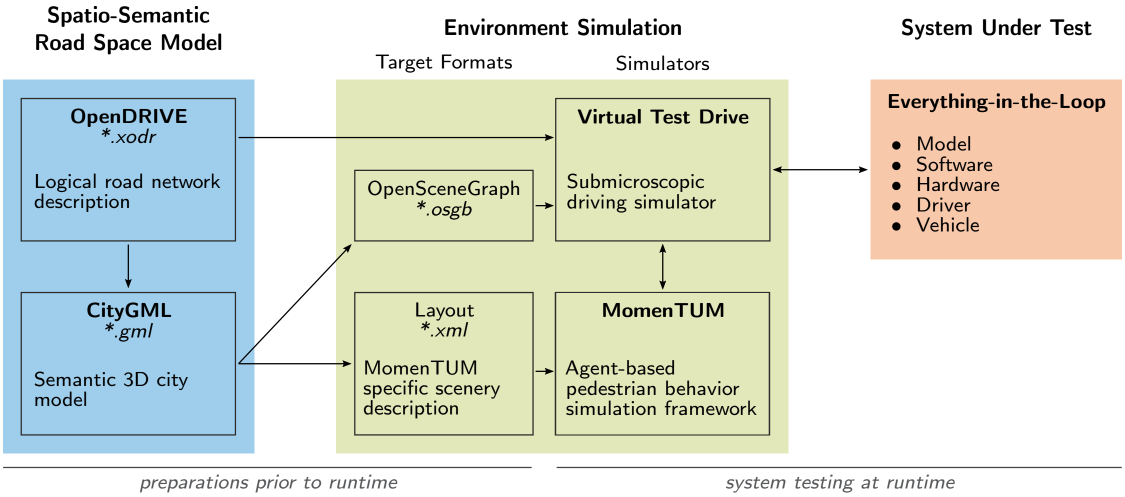

Spatio-Semantic Road Space Model
Environment Simulation
System Under Test
Target Formats
Simulators
OpenDRIVE
*.xodr
Logical road network description
CityGML
*.gml
Semantic 3D city model
OpenSceneGraph
*.osgb
Layout
*.xml
MomenTUM specific scenery description
Virtual Test Drive
Submicroscopic driving simulator
MomenTUM
Agent-based pedestrian behavior simulation framework
Everything-in-the-Loop
• Model
• Software
• Hardware
• Driver
• Vehicle
preparations prior to runtime
system testing at runtime

Figure 4. Proposed and prototypically implemented architecture: Spatio-semantic road space modeling realized through the OpenDRIVE-CityGML duality with model transformations towards the supported target formats of the respective environment simulators.

Since OpenDRIVE datasets are not only used as road network descriptions for simulation purposes, but also as HD maps for real test drives, an increasing number of highly accurate OpenDRIVE datasets of real road spaces are surveyed and prepared. The testing of perception-based localization functions leads to an increased demand for thematically rich OpenDRIVE datasets. Moreover, this trend of increasingly available OpenDRIVE datasets is expected to continue, as environment simulations must be validated against reality. Hence, as OpenDRIVE datasets are available anyway, they can serve as modeling basis. To obtain a consistent model in CityGML, a transformer is required, which is invoked prior to the simulation's runtime and discussed in Section 3.3.1.

Once the spatio-semantic road space model is constructed, it can be transformed into the supported target formats of the environment simulators used. VTD is deployed for submicroscopic driving simulations and MomenTUM is used to simulate pedestrians, whereby the simulators are coupled at runtime. MomenTUM is a pedestrian behavior simulation framework that enables the application and development of behavior models at the strategic, tactical and operational level [23]. The framework was developed at the Chair of Computational Modeling and Simulation at the Technical University of Munich and is Open Source.

3.3. Model Preparation Pipeline

The available OpenDRIVE datasets were created by the company *3D Mapping Solutions GmbH* and are based on mobile laser scannings. Their relative accuracy ranges from one to three centimeters.

3.3.1. Transforming OpenDRIVE to CityGML

To the best of our knowledge there currently exists no transformation solution that would allow resolving the geometries and semantics of OpenDRIVE and exporting them to CityGML in an expandable manner. Thus, an OpenDRIVE to CityGML transformer was implemented in the context of this project. (The project will be available for download at: https://github.com/tum-gis/rtron.) The main difficulty of the transformation is caused by the different conceptualizations of real-world objects between the two standards. As both standards are under further development, a core requirement for the implementation is therefore adaptability to change. The programming language was chosen considering the following aspects: The project should run on different operating systems and should offer a rich ecosystem of third-party libraries, particularly in the GIS domain. Furthermore, efficiency is crucial, since larger quantities of OpenDRIVE datasets are to be processed. These reflections suggest a statically typed language running on the Java Virtual Machine (JVM). Therefore, the language Kotlin was chosen, as it is interoperable with Java libraries, avoids null pointer exceptions and allows for very functional expressions [82].

Figure 5 shows the component diagram of the implemented software project. A component-based structure facilitates the separation of concerns and dependency management [83]. The lowest layer is concerned with general utility functionalities, such as methods on iterable classes within the *Standard* component. For example, mathematical functionalities are provided to other packages by wrapping third-party libraries. This enables a straightforward expandability and unit tests can be used to check whether the expected contracts are being adhered to. The second layer is concerned with the actual transformations of the models. Here, the component *Model* comprises representations of the currently valid OpenDRIVE and CityGML data models. It also contains an implementation of the road space model, which implements the actual logic including a scene-graph-like implementation to resolve the geometries to world coordinates. The *ReaderWriter* component is responsible for reading and writing the models to files. To read OpenDRIVE datasets the Java Architecture for XML Binding (JAXB) is invoked to generate Java class representations of the OpenDRIVE schema during the build process. Since the representations are dependent on the OpenDRIVE version, a mapping onto the OpenDRIVE implementation of the *Model* component is conducted. The mapping itself is implemented by using the library MapStruct [84], which automatically infers the corresponding classes and generates the mapping code at compile-time. In the case of deviations which might be caused by changes in the data model, the mapping can be altered by providing annotation-based rules or completely writing the mapping procedure for the respective class correspondence. The combination of JAXB and MapStruct architecturally enables different OpenDRIVE versions to be read in and at the same time minimizes manual adaptation efforts.

Once the OpenDRIVE dataset has been read in and is available in main memory, it can be transformed to the road space model implementation. This model implementation contains the objects with their geometries that can generate a Boundary Representation (B-Rep). For example, the logic for resolving the lane geometries is also located here. Generally, most tasks are abstracted and implemented at the utility layer to keep the transformation layer lean and maintainable. Another transformer from the road space model to CityGML is implemented by means of the library citygml4j [85]. The library provides functionality for reading, processing as well as writing CityGML datasets. B-Reps are generated and mapped onto the corresponding thematic module of CityGML. The second layer is designed to allow for adding new model transformers like exports to other map representations or transformers which are capable of manipulating OpenDRIVE datasets only.

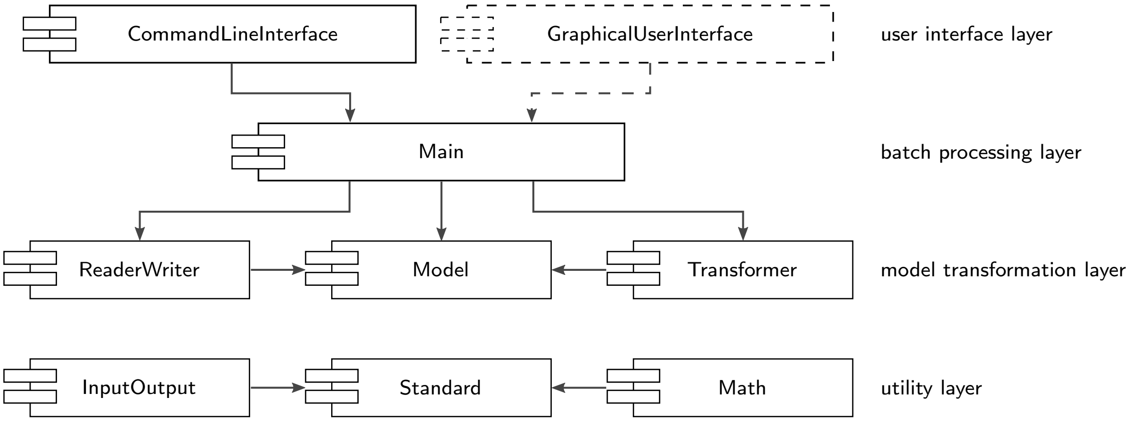


Figure 5. Component diagram of implemented OpenDRIVE to CityGML transformer with the arrows indicating dependency relationships. Components of the lowest layer provide basic functionality on which the components of the higher layers depend.

The *Main* component is responsible for coordinating the batch transformation of directories containing multiple OpenDRIVE datasets. Furthermore, the parametrization of the transformers is performed on this layer. The highest layer addresses the interface to the user. At the moment only a command line interface is implemented, but the implementation of a GUI is possible. The results of the transformation are illustrated in Figure 6. It shows the generated CityGML datasets of a crossing in Ingolstadt. Each object in CityGML contains attributes from the original OpenDRIVE dataset, such as friction, roughness, maximum allowed speed, object and lane IDs.

3.3.2. Transforming CityGML to MomenTUM's Scenery Description

The CityGML data generated from OpenDRIVE contains information on Roads, CityFurniture, SolitaryVegetationObjects and Generic objects represented with 3D geometries. Furthermore, it comprises LoD1 building models that were generated based on building layouts and height values included in the OpenDRIVE dataset. The CityGML dataset is used to create an XML document defining the geometrical layout of the simulation scenery using the data transformation software FME developed by Safe Software. Using an Extract, Transform, Load (ETL) process, FME can be used for combining and transforming (geo)data from different sources and in different formats. Geometric and semantic transformations can easily be performed using a range of predefined Transformers. The GUI of FME allows a quick and intuitive usage. Resulting data can be exported in most common data formats and be visualized using the FME Data Inspector. The FME workflow used to generate an XML MomenTUM scenery description from CityGML data is shown schematically in Figure 7. (The workbench is available for download at: https://github.com/tum-gis/momenTUM-layout-from-citygml.) First, some pre-processing of the CityGML data needs to be done to have the right geometric and semantic structure required by the pedestrian simulator MomenTUM. This includes projecting each object onto the two-dimensional plane as simulations are conducted in 2D. The original data is provided within the UTM zone 32N (EPSG:32632) coordinate system. To make coordinate values more easily manageable, systematic offsets are applied (East: −674,000, North: −5,405,000). Then, the convex hull of each object is created, and the orientation of each polygonal feature is adjusted to ensure a counterclockwise vertex winding, as this is expected by MomenTUM. All objects are aggregated to calculate a 2D bounding box of the scenery. Next, coordinates of each geometry are extracted, objects are sorted, counted and new attributes are created to be used for the assignment of CityGML objects to corresponding MomenTUM scenery elements. These elements are obstacles, areas (origin, intermediate, destination) and taggedAreas (sidewalks, crosswalks). Buildings, CityFurniture objects and Vegetation objects are directly assigned to obstacle elements. Suitable areas are defined as origins (locations where pedestrians enter the scenery), intermediates (locations pedestrians should interact with) and destinations (locations where pedestrians leave the scenery). Road and Generic objects are filtered using the CityGML function attribute to identify sidewalks, crosswalks or traffic islands (assigned to MomenTUM crosswalks). The coordinate system of the two-dimensional spatial domain is defined by a bounding box and stored in the scenery data tag, which serves as hierarchal root object to all scenery elements within the XML layout document [23,24].

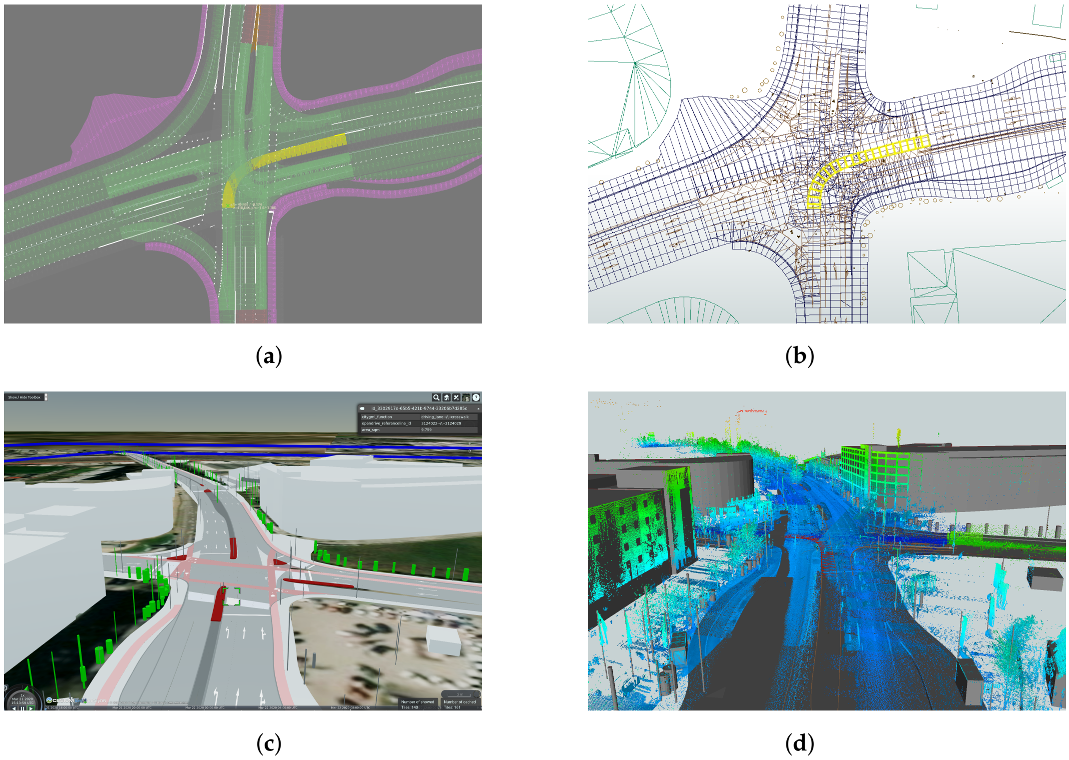


Figure 6. Consistent OpenDRIVE and CityGML representation of the Marktkaufkreuzung in Ingolstadt: A neuralgic accident black spot with intensive pedestrian and bicycle traffic. (**a**) OpenDRIVE visualized within the software *RoadEditor* by *Vires Simulationstechnologie GmbH*; (**b**) Plan view of the generated CityGML dataset within the *FME Data Inspector*; (**c**) 3D visualization of the generated CityGML dataset within browser-based *3DCityDB-Web-Map-Client* [86]; (**d**) 3D visualization of the generated CityGML dataset overlaid by georeferenced point clouds.

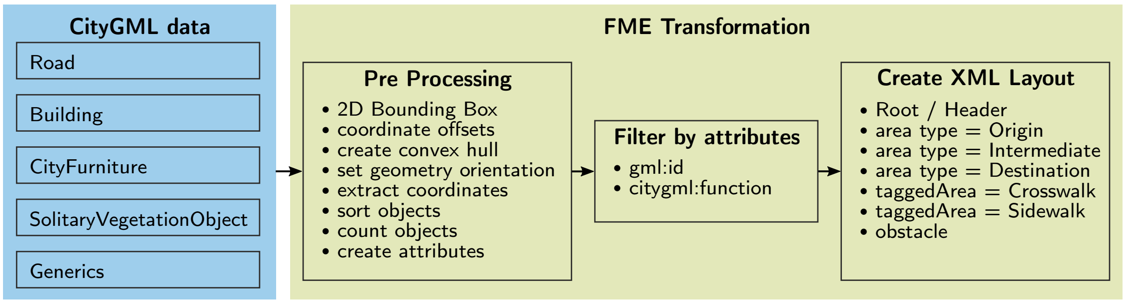


Figure 7. Workflow: Generating a MomenTUM scenery layout from CityGML using FME.

The resulting simulation scenery layout is shown in Figure 8. While obstacles generated from CityGML like Buildings, Vegetation and CityFurniture objects (red) cannot be entered, sidewalks (green) and crosswalks (yellow) are preferably used by pedestrians. As an intermediate solution, origins (dark blue), intermediates (pink) and destinations (light blue) are currently created manually. Here, CityGML building models in LoD3 could be used from other data sources, since OpenDRIVE does not support the modeling of doors, for instance. This information in combination with other city objects such as public plazas or bus stations could easily be integrated within the workflow to generate a simulation layout.

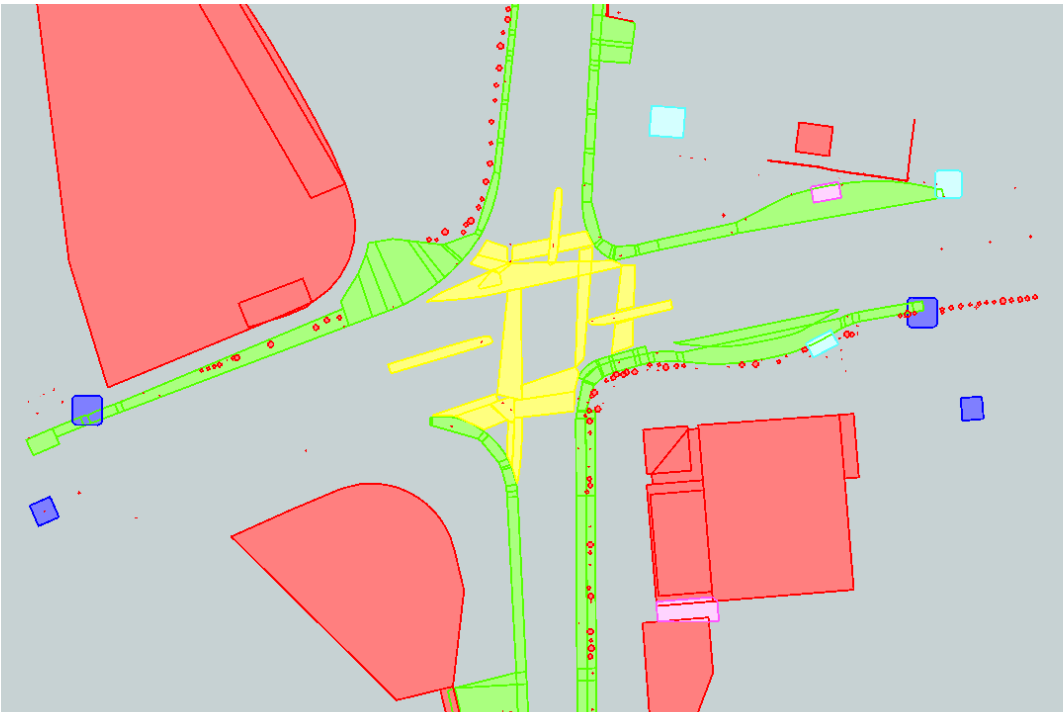


Figure 8. Simulation scenery generated from CityGML data.

3.4. Road Space Information Model

Since the concept of the spatio-semantic road space model is based on the duality of OpenDRIVE and CityGML, the entities are represented by different classes of the respective standards. Furthermore, they are also modeled in the application-specific target formats, in this case the MomenTUM layout and OpenSceneGraph (OSG). Table 1 lists all feature types used to describe the road space. Furthermore, it provides an overview of the correspondences between the modeling entities of the respective standards and formats. The mappings were carried out for the development of the proof of concept and should therefore not be conceived as final. In particular with respect to CityGML 3.0 or newer OpenDRIVE versions, the mappings between the standards must be adapted accordingly. Moreover, the geometry types that are most commonly assigned to the entities are listed. For example, the OpenDRIVE class `RoadObject` can also represent points and areas depending on the dimension values assigned.

Table 1. Entities of the road space model and their correspondences across the standards and application-specific target formats. For improved readability, the class names are partly abbreviated and written in camel case notation.

Entities	Attributes	Geometry	Spatio-Semantic Road Space Model		Target Formats	
			OpenDRIVE 1.5	**CityGML 2.0**	**MomenTUM Layout (2D)**	**VTD OpenSceneGraph**
Road						
Road reference line		line	`PlanView, ElevationProfile`	`GenericCityObject`		
Lane reference line		line	`LaneOffset`	`GenericCityObject`		
Center lane		line	`LaneSectionCenterLane`	`GenericCityObject`		
Driving lane		surface	`LaneSectionLRLane` (type: driving)	`TrafficArea`		`Geometry`
Sidewalk lane		surface	`LaneSectionLRLane` (type: sidewalk)	`TrafficArea`	`TaggedArea` (type: sidewalk)	`Geometry`
Shoulder lane		surface	`LaneSectionLRLane` (type: shoulder)	`TrafficArea`		`Geometry`
Crosswalk		surface	`RoadObject` (type: roadMark)	`GenericCityObject`	`TaggedArea` (type: crosswalk)	`Geometry`
Bus		surface	`RoadObject` (type: roadMark)	`GenericCityObject`		`Geometry`
Raised median		solid	`RoadObject` (type: barrier)	`GenericCityObject`		`Geometry`
Traffic island		solid	`RoadObject` (type: barrier)	`GenericCityObject`		`Geometry`
Road Mark						
Lane boundary	type (solid, broken, . . .), weight, color	surface	`RoadMark`	`TrafficArea`		`Geometry`
Road mark	name (arrow, stop line, . . .)	surface	`RoadObject` (type: roadMark)	`GenericCityObject`		`Geometry`
City Furniture						
Bench		solid	`RoadObject` (type: obstacle)	`CityFurniture`	`Obstacle`	`Geometry`
Controller box		solid	`RoadObject` (type: obstacle)	`CityFurniture`	`Obstacle`	`Geometry`
Fence		solid	`RoadObject` (type: barrier)	`CityFurniture`	`Obstacle`	`Geometry`
Guard rail		solid	`RoadObject` (type: barrier)	`CityFurniture`	`Obstacle`	`Geometry`
Railing		solid	`RoadObject` (type: barrier)	`CityFurniture`	`Obstacle`	`Geometry`
Street lamp		solid	`RoadObject` (type: pole)	`CityFurniture`	`Obstacle`	`Geometry`
Traffic light		point	`Signal`	`CityFurniture`		
Traffic light pole		solid	`RoadObject` (type: pole)	`CityFurniture`	`Obstacle`	`Geometry`
Traffic sign		point	`Signal`	`CityFurniture`		
Traffic sign pole		solid	`RoadObject` (type: pole)	`CityFurniture`	`Obstacle`	`Geometry`
Wall		solid	`RoadObject` (type: barrier)	`CityFurniture`	`Obstacle`	`Geometry`
Vegetation						
Tree		solid	`RoadObject` (type: tree)	`SolitaryVegetationObject`	`Obstacle`	`Geometry`
Vegetation		solid	`RoadObject` (type: vegetation)	`SolitaryVegetationObject`	`Obstacle`	`Geometry`
Building						
Building		solid	`RoadObject` (type: building)	`Building`	`Obstacle`	`Geometry`
Door				`Door`	`Area` (type: origin, destination)	`Geometry`

3.5. Runtime of the Environment Simulation

The software VTD is developed by *Vires Simulationstechnologie GmbH* for providing a virtual test environment for developing advanced driver assistance systems [13]. It can simulate the vehicle's dynamics, its sensors as well as drivers and enables the interfacing to the systems under test. VTD is designed in a very modular manner enabling the exchange of components, such as the intelligent driver model or the single-track model for lateral vehicle dynamics. However, Sippl et al. have identified a deficiency with respect to a development framework for pedestrian behavior models within VTD. It is possible to define path and motion sequences for pedestrians, but an absence of pedestrian behavior models is prevalent. The simulation of Vulnerable Road Users (VRUs) and their interaction with automated vehicles becomes particularly important for validating automated driving functions in urban environments [18]. Even though MomenTUM was not initially tailored towards the validation of driving functions, the framework with its already implemented behavior models can be used to evaluate new behavior modeling approaches. Therefore, Sippl et al. have developed a distributed real-time simulation based on VTD and MomenTUM. Figure 9 shows the architecture used for coupling the two simulators at runtime, which constitutes a modified version of Sippl et al.'s work. Here, MomenTUM starts the message broker ActiveMQ and sends the pedestrian states towards an intermediary process, which is implemented within the development software "Automotive Data and Time-Triggered Framework (ADTF)". The intermediary process generates, absorbs and controls the pedestrians within VTD by using the Runtime Data Bus (RDB) interface. To allow the pedestrian behavior models to react to the simulated VTD-vehicles, the vehicle states are sent to MomenTUM through the intermediary process. The clocks of the two simulators are synchronized via the Simulation Control Protocol (SCP), whereas VTD acts as timing master.

The operational behavior model of the pedestrians is based on Helbing and Molnár's social force model [53]. Zeng et al. extended the social force model by *inter alia* adding a repulsive force from vehicles and a force that gravitates towards the inside of the crosswalk [54,74]. A navigation graph is generated for route finding on the tactical level, whereas the edges follow a risk-based weighting. An edge located on a sidewalk will have a lower risk-weighting than an edge located on the road. The pedestrian behavior for finding the route is modeled via minimizing the risk-weighted graph and allowing for different risk tolerances. At the strategic level, the pedestrian's target selection is modeled using an Origin-Destination (OD) matrix with its respective areas shown in Figure 8. Since the objective of this work is to research road space modeling, we refer to Sippl et al. for a more detailed discussion on the used behavior models [18].

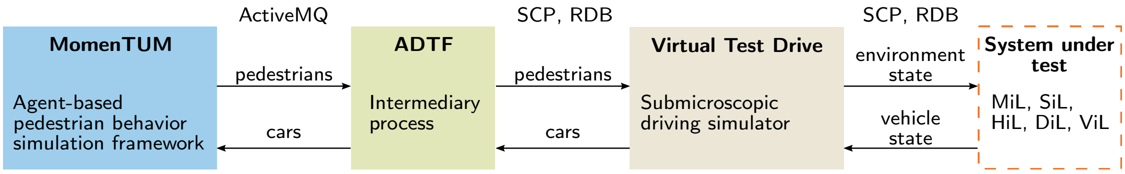


Figure 9. Communication between MomenTUM and VTD with the conceptual integration of a system under test. Adapted from Sippl et al. [18].

3.6. First Results of the Coupled Simulations

Figures 10 and 11 show screenshots of the successfully performed distributed simulation during runtime based on the discussed concept. In the simulated scenario, seven pedestrians cross the intersection from south to north in Figure 10 and from right to left in Figure 11. Figure 10 shows screenshots of VTD's scenario editor, which is used to set the initial positions of the vehicles and provides a 2D visualization during simulation runtime. The basis for this 2D visualization is solely the OpenDRIVE dataset. Figure 11 shows a 3D visualization of the ego-vehicle in red within its environment, whereby the static elements are not textured. VTD's image generator is based on

OpenSceneGraph (OSG) and the static environmental objects have been transformed using the library libcitygml [87].

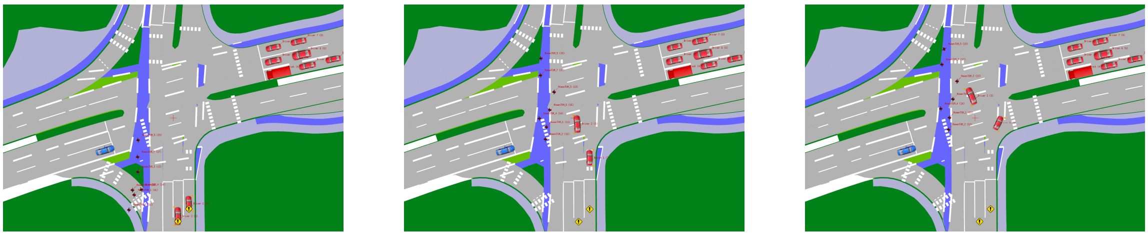


Figure 10. Bird's eye view of the simulation scenario at runtime visualized by VTD's scenario editor.

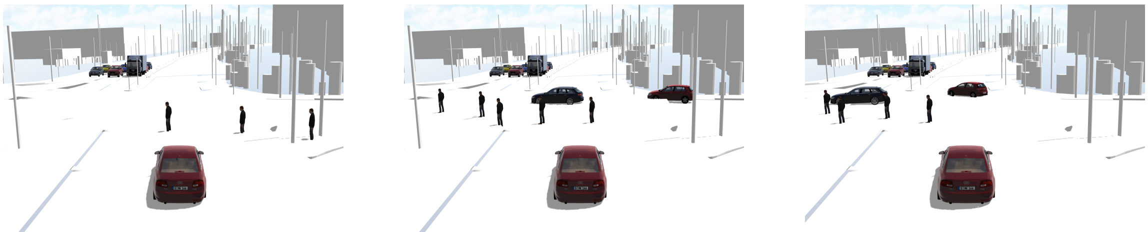
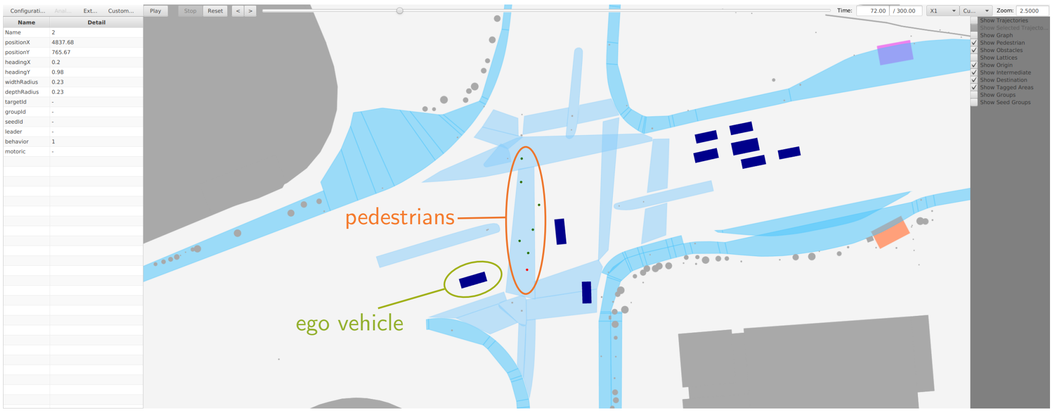


Figure 11. 3D visualization of the simulation scenario at runtime, whereby the vehicles are inserted by VTD and the static object geometries were generated from CityGML before runtime.

The pedestrian simulator MomenTUM was configured to write the states of the simulated pedestrians as well as the vehicles received from VTD into a CSV file. The recorded trajectories of the dynamic objects can be replayed and examined in MomenTUM's visualizer, as shown in Figure 12.



Name	Detail
Name	2
positionX	4837.68
positionY	765.67
headingX	0.2
headingY	0.98
widthRadius	0.23
depthRadius	0.23
targetId	-
groupId	-
seedId	-
leader	-
behavior	1
motoric	-

Time: 72.00 / 300.00

Zoom: 2.5000

- □ Show Trajectories
- □ Show Selected Trajecto...
- □ Show Graph
- ☑ Show Pedestrian
- ☑ Show Obstacles
- □ Show Lattices
- ☑ Show Origin
- ☑ Show Intermediate
- ☑ Show Destination
- ☑ Show Tagged Areas
- □ Show Groups
- □ Show Seed Groups

Figure 12. MomenTUM's visualizer used for replaying and analyzing the simulated scenario based on the recorded trajectories.

4. Discussion

In the following, the obtained results are discussed, and architectural extension potentials of the proposed concept are outlined.

4.1. Assessment of the Research Questions and Hypothesis

To determine the requirements of a holistic environment simulation for research Question 1, a literature review was carried out in Section 2. The identified requirements comprise not only aspects of transport, road user and city modeling, but also contain technical simulator as well as interface requirements. However, this list cannot be regarded as exhaustive, as behavior models for motorcyclists, e-scooter riders or even skateboarders are not discussed. Based on the identified requirements, a concept for spatio-semantic road space modeling is developed to address research Question 2. To satisfy the diverse requirements from the different domains, a method for combining the standards OpenDRIVE and CityGML is introduced. Together they enable the construction of model derivations by means of established transformation tools to serve various environment simulators. The convenience of generating further model derivations requires more examination though, since the MomenTUM-specific layout and the OSG dataset constitute only a first selection. The results in Table 1 show that Hypothesis 1 can be accepted in the context of the successfully conducted proof of concept. However, the proof of concept only represents an initial starting point, and thus further evaluations are necessary for a general acceptance of Hypothesis 1. For instance, a combined microscopic traffic simulation and a bicycle simulation based on the presented concept is yet to be investigated.

4.2. Potentials of the Concept

The proposed concept is intended to integrate additional components and create further application-specific model derivations. An exemplary extended architecture is shown in Figure 13 and serves as an orientation for the further discussion. First, the presented transformer needs further refinement. The objective is to obtain a CityGML representation that is as close as possible to the concepts of OpenDRIVE, so that subsequent processing steps can then be carried out by other tools. This applies to the CityGML version 2.0 currently in use, but also to version 3.0 [76]. Therefore, the lane topologies should be implemented in the transformer and then be exportable to the Transportation module of CityGML 3.0. As the architecture of the software is designed to allow new readers, transformer elements and writers to be added, this would also enable the export into specifically prepared map formats, which are used for guidance and stabilization in the automated driving system. In addition, the transformation processes should be configurable by the user. Since the types of the road objects were described by a user-defined name in earlier OpenDRIVE versions, the mapping rules should be definable in the configuration. For this purpose, the implementation of an internal Domain Specific Language (DSL) based on Kotlin is considered, which allows type-safe configurations or even the writing of the transformation recipe. Furthermore, the closing of overlapping lane geometries with the help of FME should be further investigated.

As a next step towards harmonizing the two standards, the OpenDRIVE concepts could be realized as a CityGML ADE. This would enable the automatic mapping to an extension of the 3DCityDB schemas. The 3DCityDB is an open source geodatabase capable of storing and managing semantic 3D city models [88]. Its schema is based on CityGML and can be deployed on both Oracle Spatial/Locator as well as PostgreSQL/PostGIS database solutions. The 3DCityDB provides functionalities for the analysis of spatial data, which are needed for instance for the processing of FCD. Furthermore, the Dynamizer module of CityGML 3.0 allows the representation of time-dependent properties, such as the time-series data from IoT sensors. Such sensors could be used to observe the traffic volumes in real time, which in turn is relevant for the calibration and validation of traffic simulations. Furthermore, an increasing number of countries already provide area-wide LoD2 building models that are available as CityGML datasets anyway. These datasets in combination with functionalities of a geodatabase can facilitate and enhance the transport modeling tasks discussed in Section 2.1.

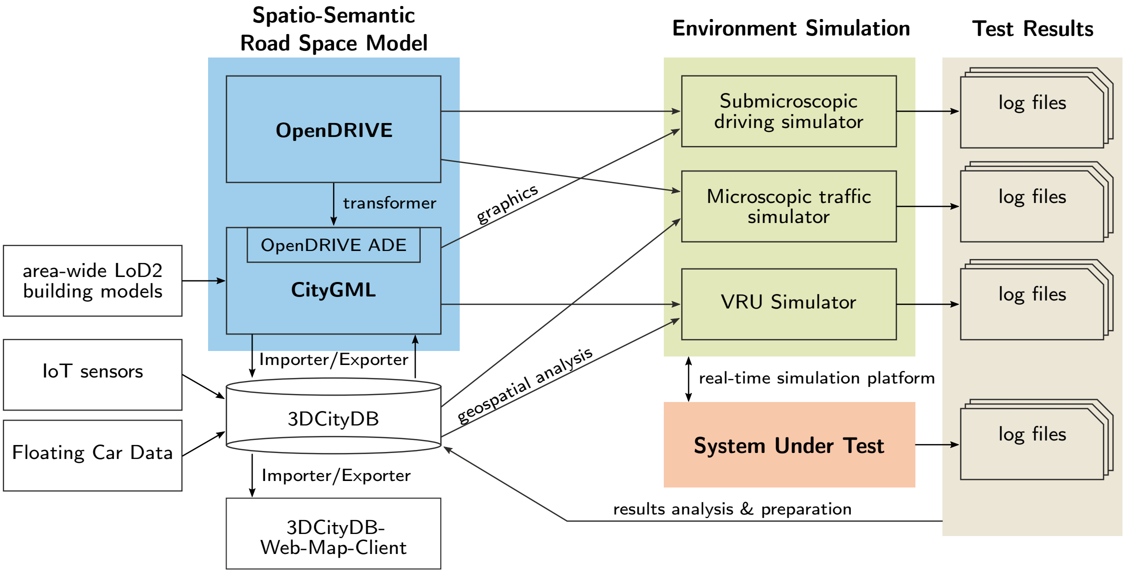


Figure 13. Proof of concept extended by further complementary tools, data sources and connections still to be investigated.

There exists a range of contributing factors that are responsible for road user accidents [89,90]. Spatial analyses of the scenery could enhance the parameterization and simulation of VRU behavior models. This includes the generation of visibility restriction maps or the analysis of sidewalk widths, for example. Furthermore, the analysis of dynamic content, such as the number of emergency braking maneuvers or path choice probabilities based on detected trajectories, could also support the parameterization of behavior models.

Another subject to be investigated is the transformations into further file formats. By means of the software FME the CityGML datasets can be transformed to glTF, OBJ and to the Unreal Game Engine. However, this still requires the replacement with more detailed models like traffic signs and lights. After the execution of the distributed simulation, the trajectories of all agents in the simulation are available as a log files. To facilitate the evaluation of the simulation results for humans, the results can be transferred back to the city model and then visualized in a web browser. As presented by Ruhdorfer et al. [91], traffic situations with dynamic elements can be visualized and made accessible to third parties by converting and loading them into the 3DCityDB-Web-Map-Client. The traffic simulation results can also be transferred back to the city model to serve as an information basis for further simulations, such as noise propagation simulations.

5. Conclusions and Future Works

A holistic environment simulation for testing automated driving systems poses various requirements to the spatio-semantic road space modeling. To meet these requirements, this paper proposes the concept of the OpenDRIVE-CityGML duality. The concept completely avoids modifications to both the standards and the simulators to ensure immediate applicability of the existing systems and datasets. Since a consistent model representation of the road space in both standards is essential for further applications, a transformer is introduced which can convert available OpenDRIVE to CityGML datasets. This enables existing HD maps to be processed with tools from the GIS domain. As a proof of concept, a traffic scenario at an accident-prone intersection is simulated by coupling the submicroscopic driving simulator VTD and the pedestrian simulator MomenTUM at runtime. Therefore, the respective target formats of the individual simulators are automatically derived from the OpenDRIVE-CityGML duality before runtime. Alongside the discussion of the research results, further architectural potentials of the concept are also assessed. In the next step, it should be investigated how and to what extent the proposed approach can support the validation of behavior models, the exploratory identification of scenarios [28] or the development of novel communication concepts [43].

Research Limitations

So far, the proposed concept was only applied for a small geographic area in Ingolstadt. Due to the use of open standards, it is expected that this methodology can be directly transferred to other and wider areas. However, this claim has yet to be substantiated. Moreover, the requirements as well as the simulation of further VRUs, such as cyclists, e-scooters and wheelchair users, still must be examined in more detail. Besides, the coupling of the simulators at runtime constitutes only a prototypical first step. At a later stage, a solution such as the Functional Engineering Platform [15] should be deployed, which ensures real-time communication and manages the simulator states.

This paper does not address the simulation of vehicle sensors, which are indispensable as they represent the interface between the automated driving system and its environment. Here, the spatio-semantic road space model should support the phenomenological as well as the physically based sensor simulation [31]. Consequently, the research question arises of how and with what additional material property information the road space model needs to be enriched. Validated and real-time-capable sensor models are essential for testing the correct system functionality and thus ensuring road safety.

Author Contributions: Conceptualization, B.S.; methodology, B.S. and C.B.; software, B.S. and C.B.; resources, B.S. and C.B.; writing—original draft preparation, B.S. and C.B.; writing—review and editing, T.H.K.; visualization, B.S. and C.B.; supervision, T.H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the German Federal Ministry of Transport and Digital Infrastructure (BMVI) within the *Automated and Connected Driving* funding program under Grant No. 16AVF2145A (SAVe).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gao, P.; Kaas, H.W.; Müller, N.; Wee, D.; Hensley, R.; Guan, M.; Möller, T.; Eckhard, G.; Bray, G.; Beiker, S.; et al. Automotive Revolution–Perspective towards 2030. Technical Report. 2016. Available online: http://hdl.voced.edu.au/10707/412253 (accessed on 13 April 2020).
2. Azmat, M.; Kummer, S.; Moura, L.T.; Gennaro, F.D.; Moser, R. Future Outlook of Highway Operations with Implementation of Innovative Technologies Like AV, CV, IoT and Big Data. *Logistics* **2019**, *3*, 15. [CrossRef]
3. Chan, C.Y. Advancements, Prospects, and Impacts of Automated Driving Systems. *Int. J. Transp. Sci. Technol.* **2017**, *6*, 208–216. [CrossRef]
4. Di Fabio, U.; Broy, M.; Brüngger, R.J.; Eichhorn, U.; Grunwald, A.; Heckmann, D.; Hilgendorf, E.; Kagermann, H.; Losinger, A.; Lutz-Bachmann, M.; et al. Ethics Commission: Automated and Connected Driving. 2017. Available online: https://www.bmvi.de/SharedDocs/EN/publications/report-ethics-commission (accessed on 29 March 2020).
5. Jing, P.; Xu, G.; Chen, Y.; Shi, Y.; Zhan, F. The Determinants behind the Acceptance of Autonomous Vehicles: A Systematic Review. *Sustainability* **2020**, *12*, 1719. [CrossRef]
6. Wintersberger, S.; Azmat, M.; Kummer, S. Are We Ready to Ride Autonomous Vehicles? A Pilot Study on Austrian Consumers' Perspective. *Logistics* **2019**, *3*, 20. [CrossRef]
7. Arena, F.; Pau, G.; Collotta, M. A Survey on Driverless Vehicles: From Their Diffusion to Security Features. *J. Internet Serv. Inf. Secur.* **2018**, *8*, 1–19. [CrossRef]
8. Wood, M.; Robbel, P.; Maass, M.; Tebbens, R.D.; Meijs, M.; Harb, M.; Reach, J.; Robinson, K.; Wittmann, D.; Srivastava, T.; et al. Safety First for Automated Driving. Technical Report. 2019. Available online: https://connectedautomateddriving.eu/wp-content/uploads/2019/09/Safety_First_for_Automated_Driving.pdf (accessed on 29 March 2020).
9. Wachenfeld, W.; Junietz, P.; Wenzel, R.; Winner, H. The Worst-Time-to-Collision Metric for Situation Identification. In Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, 19–22 June 2016; pp. 729–734. [CrossRef]

10. Von Neumann-Cosel, K. Virtual Test Drive. Ph.D. Thesis, Technical University of Munich, Department of Informatics, Munich, Germany. 2014. Available online: https://mediatum.ub.tum.de/1126934 (accessed on 29 March 2020).
11. Schuldt, F. Ein Beitrag für den Methodischen Test von Automatisierten Fahrfunktionen mit Hilfe von Virtuellen Umgebungen. Ph.D. Thesis, Department of Electrical Engineering, Information Technology, Physics, Technical University Braunschweig, Braunschweig, Germany, 2017. [CrossRef]
12. Riedmaier, S.; Nesensohn, J.; Gutenkunst, C.; Duser, T.; Schick, B.; Abdellatif, H. Validation of X-in-the-Loop Approaches for Virtual Homologation of Automated Driving Functions. In Proceedings of the 11th Graz Symposium Virtual Vehicle (GSVF), Graz, Austria, 15–16 May 2018.
13. Von Neumann-Cosel, K.; Dupuis, M.; Weiss, C. Virtual Test Drive-Provision of a Consistent Tool-Set for [d, h, s, v]-in-the-Loop. In Proceedings of the 2009 Driving Simulation Conference Europe (DSC), Monte Carlo, Monaco, 4–6 February 2009.
14. Bock, F.; Sippl, C.; Siegl, S.; German, R. Status Report on Automotive Software Development. In *Automotive Systems and Software Engineering: State of the Art and Future Trends*; Dajsuren, Y., van den Brand, M., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 29–57. [CrossRef]
15. Stadler, C.; Gruber, T. Functional Engineering Platform—A Continuous Approach Towards Function Development. In *Simulation and Testing for Vehicle Technology*; Gühmann, C., Riese, J., von Rüden, K., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 69–84. [CrossRef]
16. Schiller, M.; Dupius, M.; Krajzewicz, D.; Kern, A.; Knoll, A. Multi-Resolution Traffic Simulation for Large-Scale High-Fidelity Evaluation of VANET Applications. In *Simulating Urban Traffic Scenarios*; Behrisch, M., Weber, M., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 17–36. [CrossRef]
17. Semrau, M.; Erdmann, J. Simulation Framework for Testing ADAS in Chinese Traffic Situations. In Proceedings of the 2016 SUMO User Conference, Berlin, Germany, 23–25 May 2016; pp. 103–115.
18. Sippl, C.; Schwab, B.; Kielar, P.; Djanatliev, A. Distributed Real-Time Traffic Simulation for Autonomous Vehicle Testing in Urban Environments. In Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2562–2567. [CrossRef]
19. Association for Standardization of Automation and Measuring Systems e.V.. ASAM OpenDRIVE 1.6 Base Standard. Technical Report. 2020. Available online: https://www.asam.net/standards/detail/opendrive/ (accessed on 29 March 2020).
20. Kolbe, T.H. Representing and Exchanging 3D City Models with CityGML. In *3D Geo-Information Sciences*; Cartwright, W., Gartner, G., Meng, L., Peterson, M.P., Lee, J., Zlatanova, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 15–31. [CrossRef]
21. Gröger, G.; Kolbe, T.H.; Nagel, C.; Häfele, K.H. *OGC City Geography Markup Language CityGML Encoding Standard*; Technical Report; OGC Doc.No. 12-019; 2012. Available online: https://www.ogc.org/standards/citygml (accessed on 29 April 2020).
22. VIRES Simulationstechnologie GmbH. Virtual Test Drive. Available online: https://vires.com/vtd-vires-virtual-test-drive/ (accessed on 29 March 2020).
23. Kielar, P.M.; Biedermann, D.H.; Borrmann, A. *MomenTUMv2: A Modular, Extensible, and Generic Agent-Based Pedestrian Behavior Simulation Framework*; Technical University of Munich, Department of Civil, Geo and Environmental Engineering: Munich, Germany, 2016. [CrossRef]
24. Kielar, P.M. Kognitive Modellierung Und Computergestützte Simulation Der Räumlich-Sequenziellen Zielauswahl von Fußgängern. Ph.D. Thesis, Technical University of Munich, Department of Civil, Geo and Environmental Engineering, Munich, Germany, 2017. Available online: https://mediatum.ub.tum.de/1359816 (accessed on 29 March 2020).
25. Menzel, T.; Bagschik, G.; Maurer, M. Scenarios for Development, Test and Validation of Automated Vehicles. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1821–1827. [CrossRef]
26. PEGASUS Project Partners. PEGAUS Method: An Overview. Technical Report. 2019. Available online: https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/PEGASUS-Gesamtmethode.pdf (accessed on 29 March 2020).

27. Sippl, C.; Bock, F.; Lauer, C.; Heinz, A.; Neumayer, T.; German, R. Scenario-Based Systems Engineering: An Approach Towards Automated Driving Function Development. In Proceedings of the 13th Annual IEEE International Systems Conference (SYSCON), Orlando, FL, USA, 8–11 April 2019; pp. 1–8. [CrossRef]
28. Sippl, C.; Bock, F.; Huber, B.; Djanatliev, A.; German, R. Identifying relevant traffic situations based on human decision making. In Proceedings of the 19th Stuttgart International Symposium, Wiesbaden, Germany, 19–20 March 2019; pp. 1003–1017. [CrossRef]
29. Bagschik, G.; Menzel, T.; Maurer, M. Ontology Based Scene Creation for the Development of Automated Vehicles. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1813–1820. [CrossRef]
30. Hoogendoorn, S.P.; Bovy, P.H.L. State-of-the-Art of Vehicular Traffic Flow Modelling. *Proc. Inst. Mech. Eng. Part I J. Syst. Control. Eng.* **2001**, *215*, 283–303. [CrossRef]
31. Schwab, B.; Kolbe, T.H. Requirement Analysis of 3D Road Space Models for Automated Driving. *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2019**, *IV-4/W8*, 99–106. [CrossRef]
32. Richter, A.; Löwner, M.O.; Ebendt, R.; Scholz, M. Towards an Integrated Urban Development Considering Novel Intelligent Transportation Systems. *Technol. Forecast. Soc. Chang.* **2020**, *155*, 119970. [CrossRef]
33. Ortúzar, J.d.D.; Willumsen, L.G.; Consultancy, L.W. *Modelling Transport*, 4th ed.; John Wiley & Sons, Ltd.: Chichester, UK, 2011.
34. Biljecki, F.; Arroyo Ohori, K.; Ledoux, H.; Peters, R.; Stoter, J. Population Estimation Using a 3D City Model: A Multi-Scale Country-Wide Study in the Netherlands. *PLoS ONE* **2016**, *11*, e0156808. [CrossRef]
35. Kaden, R.; Kolbe, T.H. City-Wide Total Energy Demand Estimation of Buildings Using Semantic 3D City Models and Statistical Data. *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2013**, *II-2/W1*, 163–171. [CrossRef]
36. Bonnel, P.; Fekih, M.; Smoreda, Z. Origin-Destination Estimation Using Mobile Network Probe Data. *Transp. Res. Procedia* **2018**, *32*, 69–81. [CrossRef]
37. Donges, E. Driver Behavior Models. In *Handbook of Driver Assistance Systems*; Winner, H., Hakuli, S., Lotz, F., Singer, C., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 19–33. [CrossRef]
38. Rasmussen, J. Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models. *IEEE Trans. Syst. Man, Cybern.* **1983**, *SMC-13*, 257–266. [CrossRef]
39. Michon, J.A. A Critical View of Driver Behavior Models: What Do We Know, What Should We Do? In *Human Behavior and Traffic Safety*; Evans, L., Schwing, R.C., Eds.; Springer: Boston, MA, USA, 1985; pp. 485–524. [CrossRef]
40. Donges, E. A Two-Level Model of Driver Steering Behavior. *Hum. Factors* **1978**, *20*, 691–707. [CrossRef]
41. Andersen, G.J.; Sauer, C.W. Optical Information for Car Following: The Driving by Visual Angle (DVA) Model. *Hum. Factors J. Hum. Factors Ergon. Soc.* **2007**, *49*, 878–896. [CrossRef] [PubMed]
42. Brackstone, M.; McDonald, M. Car-Following: A Historical Review. *Transp. Res. Part Traffic Psychol. Behav.* **1999**, *2*, 181–196. [CrossRef]
43. Arena, F.; Pau, G.; Severino, A. V2X Communications Applied to Safety of Pedestrians and Vehicles. *J. Sens. Actuator Netw.* **2020**, *9*, 3. [CrossRef]
44. Xu, J.; Li, Y.; Lu, G.; Zhou, W. Reconstruction Model of Vehicle Impact Speed in Pedestrian–Vehicle Accident. *Int. J. Impact Eng.* **2009**, *36*, 783–788. [CrossRef]
45. Rubrecht, B.; Bastien, C.; Davies, H.; Wellings, R.; Burnett, B. Numerical Validation of the Pedestrian Crossing Speed Calculator (PCSC) Using Finite Element Simulations. *Glob. J. Forensic Sci. Med. Forensic* **2019**. *1*, GJFSM-19-RA-525.
46. Hoogendoorn, S.P.; Bovy, P.H.L.; Daamen, W. Microscopic Pedestrian Wayfinding and Dynamics Modelling. In *Pedestrian and Evacuation Dynamics*; Schreckenberg, M., Sharma, S.D., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 123–154.
47. Papadimitriou, E.; Yannis, G.; Golias, J. A Critical Assessment of Pedestrian Behaviour Models. *Transp. Res. Part Traffic Psychol. Behav.* **2009**, *12*, 242–255. [CrossRef]
48. Kielar, P.M.; Borrmann, A. Modeling Pedestrians' Interest in Locations: A Concept to Improve Simulations of Pedestrian Destination Choice. *Simul. Model. Pract. Theory* **2016**, *61*, 47–62. [CrossRef]
49. Bierlaire, M.; Robin, T. Pedestrians Choices. In *Pedestrian Behavior*; Timmermans, H., Ed.; Emerald Group Publishing Limited: Bingley, UK, 2009; pp. 1–26. [CrossRef]

50. Kneidl, A.; Borrmann, A.; Hartmann, D. Generation and Use of Sparse Navigation Graphs for Microscopic Pedestrian Simulation Models. *Adv. Eng. Inform.* **2012**, *26*, 669–680. [CrossRef]
51. Hartmann, D. Adaptive Pedestrian Dynamics Based on Geodesics. *New J. Phys.* **2010**, *12*, 043032. [CrossRef]
52. Blue, V.J.; Adler, J.L. Cellular Automata Microsimulation for Modeling Bi-Directional Pedestrian Walkways. *Transp. Res. Part B Methodol.* **2001**, *35*, 293–312. [CrossRef]
53. Helbing, D.; Molnár, P. Social Force Model for Pedestrian Dynamics. *Phys. Rev. E* **1995**, *51*, 4282–4286. [CrossRef] [PubMed]
54. Zeng, W.; Chen, P.; Nakamura, H.; Iryo-Asano, M. Application of Social Force Model to Pedestrian Behavior Analysis at Signalized Crosswalk. *Transp. Res. Part Emerg. Technol.* **2014**, *40*, 143–159. [CrossRef]
55. Twaddle, H.A. Development of Tactical and Operational Behaviour Models for Bicyclists Based on Automated Video Data Analysis. Ph.D. Thesis, Technical University of Munich, Department of Civil, Geo and Environmental Engineering, Munich, Germany, 2017. Available online: https://mediatum.ub.tum.de/1366878 (accessed on 29 March 2020).
56. Hoogendoorn, S.P.; Daamen, W. Microscopic Calibration and Validation of Pedestrian Models: Cross-Comparison of Models Using Experimental Data. In *Traffic and Granular Flow'05*; Schadschneider, A., Pöschel, T., Kühne, R., Schreckenberg, M., Wolf, D.E., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 329–340. [CrossRef]
57. Heo, J.; Jeong, S.; Park, H.K.; Jung, J.; Han, S.; Hong, S.; Sohn, H.G. Productive High-Complexity 3D City Modeling with Point Clouds Collected from Terrestrial LiDAR. *Comput. Environ. Urban Syst.* **2013**, *41*, 26–38. [CrossRef]
58. IPG Automotive GmbH. CarMaker: Virtual Testing of Automobiles and Light-Duty Vehicles. Available online: https://ipg-automotive.com/products-services/simulation-software/carmaker/ (accessed on 29 March 2020).
59. Vector Informatik GmbH. DYNA4–Virtual Test Driving. Available online: https://www.vector.com/dyna4 (accessed on 29 March 2020).
60. Mechanical Simulation Corporation. CarSim. Available online: https://www.carsim.com/products/carsim/ (accessed on 29 March 2020).
61. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. In Proceedings of the 1st Annual Conference on Robot Learning (CoRL), Mountain View, CA, USA, 13–15 November 2017; pp. 1–16. Available online: http://proceedings.mlr.press/v78/dosovitskiy17a/dosovitskiy17a.pdf (accessed on 29 March 2020).
62. Casas, J.; Ferrer, J.L.; Garcia, D.; Perarnau, J.; Torday, A. Traffic Simulation with Aimsun. In *Fundamentals of Traffic Simulation*; Barceló, J., Ed.; Springer: New York, NY, USA, 2010; Volume 145, pp. 173–232. [CrossRef]
63. PTV Planung Transport Verkehr AG. PTV Vissim. Available online: https://www.ptvgroup.com/en/solutions/products/ptv-vissim/ (accessed on 29 March 2020).
64. Krajzewicz, D.; Erdmann, J.; Behrisch, M.; Bieker, L. Recent Development and Applications of SUMO-Simulation of Urban MObility. *Int. J. Adv. Syst. Meas.* **2012**, *5*, 128–138.
65. Saidallah, M.; El Fergougui, A.; Elalaoui, A.E. A Comparative Study of Urban Road Traffic Simulators. *Matec Web Conf.* **2016**, *81*, 05002. [CrossRef]
66. Mubasher, M.M.; Syed Waqar ul Qounain, J. Systematic Literature Review of Vehicular Traffic Flow Simulators. In Proceedings of the 2015 International Conference on Open Source Software Computing (OSSCOM), Amman, Jordan, 10–13 September 2015; pp. 1–6. [CrossRef]
67. Barceló, J. (Ed.) *Fundamentals of Traffic Simulation*; International Series in Operations Research & Management Science; Springer: New York, NY, USA, 2010; Volume 145. [CrossRef]
68. Wagoum, A.U.K.; Chraibi, M.; Zhang, J.; Lämmel, G. JuPedSim: An Open Framework for Simulating and Analyzing the Dynamics of Pedestrians. In Proceedings of the 3rd Conference of Transportation Research Group of India (CTRG), Kolkata, India, 17–20 December 2015.
69. Curtis, S.; Best, A.; Manocha, D. Menge: A Modular Framework for Simulating Crowd Movement. *Collect. Dyn.* **2016**, *1*, 1–40. [CrossRef]
70. Kleinmeier, B.; Zönnchen, B.; Gödel, M.; Köster, G. Vadere: An Open-Source Simulation Framework to Promote Interdisciplinary Understanding. *Collect. Dyn.* **2019**, *4*. [CrossRef]
71. PTV Planung Transport Verkehr AG. PTV Viswalk. Available online: https://www.ptvgroup.com/en/solutions/products/ptv-viswalk/ (accessed on 29 March 2020).

72. accu:rate GmbH. Crowd:It. Available online: https://www.accu-rate.de/en/software-crowd-it-en/ (accessed on 29 March 2020).
73. Strigel, E.; Meissner, D.; Seeliger, F.; Wilking, B.; Dietmayer, K. The Ko-PER Intersection Laserscanner and Video Dataset. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 1900–1901. [CrossRef]
74. Zeng, W.; Chen, P.; Yu, G.; Wang, Y. Specification and Calibration of a Microscopic Model for Pedestrian Dynamic Simulation at Signalized Intersections: A Hybrid Approach. *Transp. Res. Part C Emerg. Technol.* **2017**, *80*, 37–70. [CrossRef]
75. Association for Standardization of Automation and Measuring Systems e.V.. ASAM OpenCRG 1.1.2. Technical Report. 2018. Available online: https://www.asam.net/standards/detail/opencrg/ (accessed on 29 March 2020).
76. Kutzner, T.; Chaturvedi, K.; Kolbe, T.H. CityGML 3.0: New Functions Open Up New Applications. *PFG J. Photogramm. Remote. Sens. Geoinf. Sci.* **2020**. [CrossRef]
77. Beil, C.; Kolbe, T.H. CityGML and the Streets of New York – A Proposal for Detailed Street Space Modelling. *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2017**, *IV-4/W5*, 9–16. [CrossRef]
78. Labetski, A.; van Gerwen, S.; Tamminga, G.; Ledoux, H.; Stoter, J. A Proposal for an Improved Transportation Model in CityGML. *ISPRS Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2018**, *XLII-4/W10*, 89–96. [CrossRef]
79. Biljecki, F.; Kumar, K.; Nagel, C. CityGML Application Domain Extension (ADE): Overview of Developments. *Open Geospat. Data Softw. Stand.* **2018**, *3*, 13. [CrossRef]
80. Van den Brink, L.; Stoter, J.; Zlatanova, S. UML-Based Approach to Developing a CityGML Application Domain Extension. *Trans. GIS* **2013**, *17*, 920–942. [CrossRef]
81. Safe Software Inc. FME–Feature Manipulation Engine. Available online: https://www.safe.com/fme/ (accessed on 29 March 2020).
82. Jemerov, D.; Isakova, S. *Kotlin in Action*; Manning Publications Co.: Shelter Island, NY, USA, 2017.
83. Martin, R.C. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*, 1st ed.; Robert C. Martin Series; Prentice Hall: Upper Saddle River, NJ, USA, 2017.
84. MapStruct. MapStruct – Java Bean Mappings, the Easy Way! Available online: https://mapstruct.org (accessed on 29 March 2020).
85. Nagel, C. Citygml4j. Available online: https://github.com/citygml4j/citygml4j (accessed on 29 March 2020).
86. Chair of Geoinformatics, Technical University of Munich. 3DCityDB-Web-Map-Client. Available online: https://github.com/3dcitydb/3dcitydb-web-map (accessed on 29 March 2020).
87. Klimke, J. Libcitygml. Available online: https://github.com/jklimke/libcitygml (accessed on 29 March 2020).
88. Yao, Z.; Nagel, C.; Kunde, F.; Hudra, G.; Willkomm, P.; Donaubauer, A.; Adolphi, T.; Kolbe, T.H. 3DCityDB—A 3D Geodatabase Solution for the Management, Analysis, and Visualization of Semantic 3D City Models Based on CityGML. *Open Geospat. Data, Softw. Stand.* **2018**, *3*. [CrossRef]
89. Casado-Sanz, N.; Guirao, B.; Lara Galera, A.; Attard, M. Investigating the Risk Factors Associated with the Severity of the Pedestrians Injured on Spanish Crosstown Roads. *Sustainability* **2019**, *11*, 5194. [CrossRef]
90. Congiu, T.; Sotgiu, G.; Castiglia, P.; Azara, A.; Piana, A.; Saderi, L.; Dettori, M. Built Environment Features and Pedestrian Accidents: An Italian Retrospective Study. *Sustainability* **2019**, *11*, 1064. [CrossRef]
91. Ruhdorfer, R.; Willenborg, B.; Sindram, M. Coupling of Traffic Simulations and Semantic 3D City Models. *gis.Science* **2018**, *3*, 101–109. Available online: https://gispoint.de/artikelarchiv/gis/2018/gisscience-ausgabe-32018/4611-kopplung-von-verkehrssimulationen-und-semantischen-3d-stadt-modellen.html (accessed on 29 March 2020).