Specifying Autonomous Driving Scenarios

Yue Yu*, Tiexin Wang*, Tao Yue*
*Nanjing University of Aeronautics and Astronautics
Email: * {yuyue, tiexin.wang, taoyue}@nuaa.edu.cn

***Abstract*—Defects in Autonomous driving systems (ADSs) might result in catastrophic losses of lives and properties. To avoid such defects, we need to first ensure high quality requirements, which highly possibly would lead to the delivery of high-quality ADSs. Specifying requirements for ADSs, a method needs to have terms/notations specific to ADSs such as complex traffic environments (e.g., pedestrians, roads). Use case modeling is commonly practiced in industry for requirements specification and modeling. In this paper, we propose a novel use case modeling methodology, named RUCM4ADS, which specializes the Restricted Use Case Modeling (RUCM). RUCM4ADS aims to specify ADS scenarios by integrating elements from both the autonomous driving domain and Operational World Model (OWM) Ontology. Accompanied with RUCM4ADS, we also develop an editor for it. To evaluate RUCM4ADS, we conducted one real-world case study with 10 use cases. We also conducted a preliminary controlled experiment, in a laboratory setting, to evaluate the applicability of RUCM4ADS. Results show that RUCM4ADS can be used for modeling ADS scenarios and has the potential to improve the overall applicability for specifying ADS scenarios as use case models.**

***Index Terms*—meta-modeling, autonomous driving system, environment, use case modeling**

I. Introduction

With the application of technologies such as artificial intelligence, Autonomous Driving Systems (ADSs) have evolved rapidly over the last decade. Though with a bright future ahead, developing reliable ADSs is still a challenge, as ensuring the safety of ADSs is critical and any unsafe behavior might lead to catastrophic consequences such as significant losses of lives. The delivery of high-quality ADSs depends on high quality requirements. Specifying requirements for ADSs, a method needs to have terminologies specific to ADSs such as complex traffic environments (e.g., pedestrians, roads), driving tasks classification (e.g., Strategic, Manoeuvring) and autonomous vehicle states (e.g., Velocity).

Use case modeling has been widely applied to specify requirements as use cases with natural language [1]. It has been also applied for specifying requirements in ADSs [2]. A large number of use case modeling techniques exist in the literature. Restricted Use Case Modeling (RUCM) [3] is one of them, which was originally proposed in [4], later on extended for serving various purposes such as test generations [5], [6], modeling uncertainty requirements [7]. The usefulness of RUCM has been demonstrated in these related works. In addition, use case models specified/modeled with RUCM can be transformed into other artifacts such as UML analysis models [8] and executable test cases [9].

The paper is structured as follows. Section II introduces RUCM, defines scenarios in autonomous driving and presents a running example. Section III discusses RUCM4ADS in detail. In Section IV, we demonstrate the applicability of RUCM4ADS. Section V presents the related work, and we conclude the paper in Section VI.

II. Background and running example

Considering that RUCM is generic, in the sense that it is not designed for a specific domain, for ADSs, we need to extend it for structuring and specifying ADS use case models with concepts specific to ADSs. Therefore, in this paper, we propose RUCM4ADS. To devise it, we extracted domain concepts of ADSs from *Wise Drive: Requirements Analysis Framework for Automated Driving Systems* [10] [11], and proposed a metamodel (named UCMeta4ADS) by extending UCMeta, the metamodel of RUCM. With UCMeta4ADS, a use case model specified in RUCM4ADS can be automatically instantiated as an instance of a UCMeta4ADS. Moreover, we implemented a tool for RUCM4ADS to ease the specification of use case models.

RUCM was proposed in [3] [4] with the aim to reduce ambiguity of requirements in natural language and to facilitate automated analyses. Its corresponding metamodel, UCMeta, can be used to automatically formalize RUCM models such that instances of UCMeta can be further automatically transformed into other artifacts such as UML activity diagrams [12]. UCMeta was specified with the OMG's standard Meta-Object Facility (MOF) [13]. Since RUCM was initially proposed, multiple extensions have been proposed, as we have discussed in Section I.

According to the definition in Wise Drive [10] published by the Waterloo Intelligent Systems Engineering Lab, in the autonomous driving domain, in a discrete time model, a scenario is defined as a time-stamped sequence of scenes (i.e., snapshots of the world state), or situations (i.e., scenes combining behaviors and states of all dynamic objects in the environment). Elrofai et al. [14] proposed that a scenario is the continuous change of the dynamic environment around the subject vehicle within a specific time range, including the vehicle's behavior in the environment. Based on these definitions, we propose to specify such scenarios in the autonomous driving domain as use case models.

978-1-6654-1898-0/21/$31.00 ©2021 IEEE
DOI 10.1109/REW53955.2021.00079

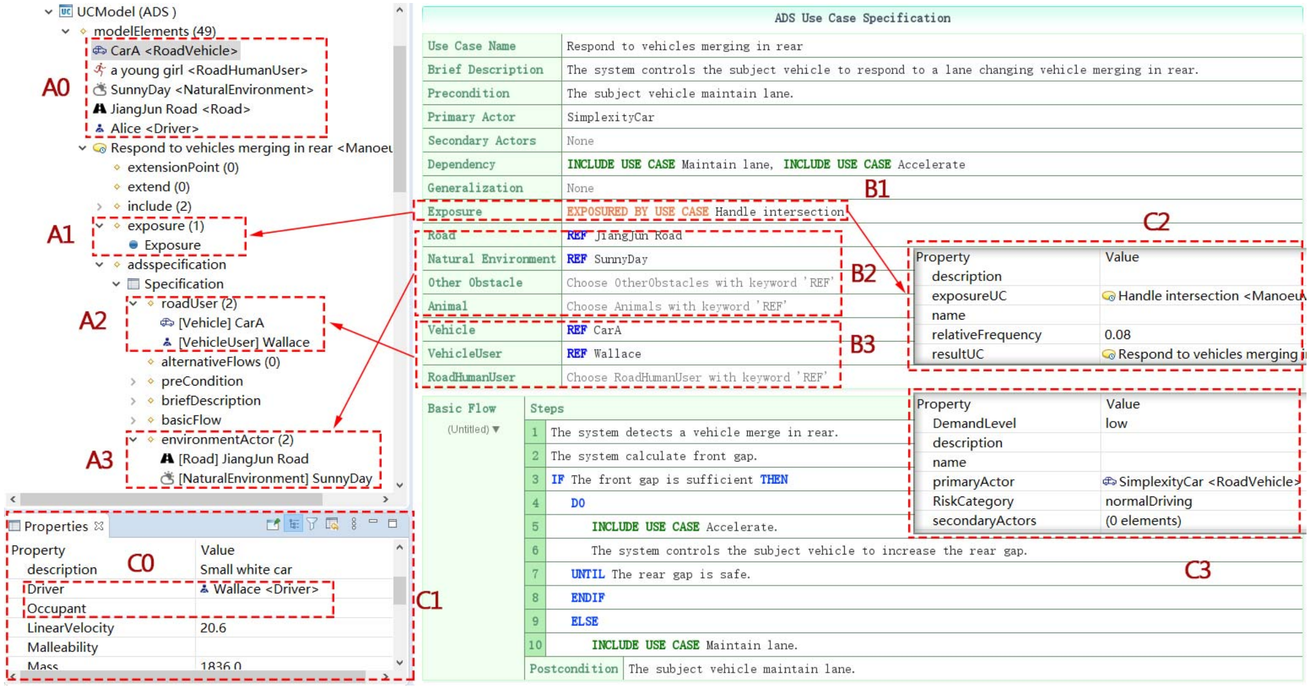

Fig. 1: Specifying use case specifications with RUCM4ADS - An Example

We use the ADS scenario *Respond to vehicles merging in rear* (Fig. 1) as a running example to illustrate key features of RUCM4ADS. This use case describes that an ADS controls the subject vehicle to respond to a lane changing vehicle merging in rear; the subject vehicle should take actions to deal with this situation. In Section 3, we illustrate RUCM4ADS in detail with the running example.

III. The RUCM4ADS Methodology

RUCM4ADS is defined to capture key features in autonomous driving scenarios as use case models. It provides concepts and methods to specify road environment, driving behaviors and vehicle states. Since RUCM4ADS is an extension to RUCM, RUCM4ADS inherits the RUCM features of specifying basic and alternative flows of events, etc. In the rest of this section, we introduce RUCM4ADS, by discussing its metamodel, UCMeta4ADS, which is presented in Fig. 2 and Fig. 3 as UML class diagrams. Each class is stereotyped with either «RUCM4ADS» or «RUCM», which, respectively, represent newly introduced metaclasses in RUCM4ADS or original RUCM metaclasses.

A. Actors

To characterize road environment, the Waterloo Intelligent Systems Engineering lab proposed the Operational World Model (OWM) ontology in Wise Drive [10]. Since the OWM ontology was designed to provide a context for specifying ADS driving behavior requirements, which is consistent with the goal of this paper, we opt for this ontology as an important reference to design our actor metamodel.

According to Wise Drive, the ontology in ADSs at the top level is organized into five categories: 1) *road structure*; 2) *road users*, including vehicles, cyclists, and pedestrians; 3) *animals*; 4) *other obstacles*; and 5) *environmental conditions*. We adapted this classification and proposed the classification of RUCM4ADS actors, which, as shown in Fig. 2, are: 1) *Natural Environment*, 2) *Road*, 3) *Animal*, 4) *HumanActor*, and 5) *Object*.

1) Object: In OWM, objects can be fixed or non-fixed; fixed objects is stationary and non-fixed objects can be stationary or moving. According to [10], fixed objects have the following properties: 1) *position* and *orientation*, 2) *extent*, 3) *mass* and inertia, and 4) *malleability*. Non-fixed objects additionally have properties: 1) *linear velocity*, 2) *angular velocities*, 3) *acceleration*, and 4) *angular accelerations*. Position, size, mass and malleability are the most relevant properties of objects to traffic. For instance, *mass* matters because very small objects can be ignored since they can easily fit under the vehicle without causing any trouble. Moreover, the *malleability* of an object is about whether it is rigid (e.g., rocks) or deformable (e.g., clod). We capture these properties as the attributes of metaclass *Object* (Fig. 2).

Object is further classified into *Vehicle* and *OtherObstacle*. defined in Wise Drive, an obstacle is any object that is in the path of the subject vehicle. The major class of objects found on a roadway is *Vehicle*, so metaclass *OtherObstacle* focuses on other types of objects that may occur on a roadway such as barriers, traffic signs, and large stones. *Vehicle* can be further classified into *RoadVehicle* (e.g., cars, buses, motorcycles), *OffRoadVehicle* (e.g., animal-drawn vehicles, tracked vehicles)

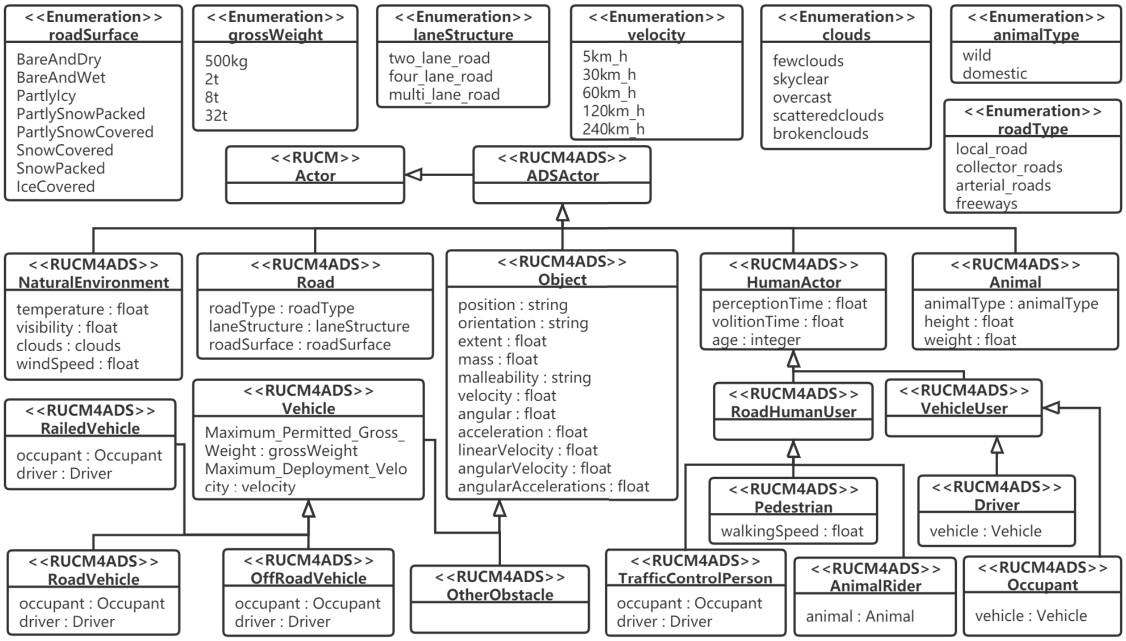

Fig. 2: Actor metamodel

and *RailedVehicle* (e.g., railway trains, street cars). In the RUCM4ADS editor, as shown in *C1* of Fig. 1, values of attributes of a *vehicle* are specified in the property window.

2) HumanActor: We classify *HumanActor* into *VehicleUser* and *RoadHumanUser*. Based on Wise Drive, we further classify *VehcileUser* into *Driver* and *Occupant*, and *RoadHuamnUser* into *Pedestrian*, *AnimalRider* and *TrafficControlPerson*. Human is an unstable factor in road environments due to the high unpredictability of their intentions and behaviors. As reported in the domains of traffic psychology [15] [16] and traffic sociology [17], factors of influencing human behaviors include: 1) physical and cognitive capacity, 2) skills and experience in road traffic, 3) personality factors, 4) attitudes and beliefs relative to objects, 5) situations and practices in road traffic, and 6) emotions. Considering it is currently unrealistic or very difficult to know such detailed properties of pedestrians, we therefore select three measurable properties in UCMeta4ADS for *HumanActor*: *age*, *perceptionTime* and *volitionTime*. Particularly, *perceptionTime* is for seeing or discerning an object or event, and *volitionTime* is the time required to initiate an action, e.g., engaging the brake [18]. We classify *HumanActor* into *RoadHumanUser* and *VehicleUser*. Just as their names imply, *RoadHumanUser* are human on the road including *Pedestrian*, *TrafficControlPerson* and *AnimalRider*, while *VehicleUser* are human inside the vehicle including *Driver* and *Occupant*. As *C0* of Fig. 1 shows, as an reference attribute, *Driver* and *Occupant* can be referenced by a *Vehicle* in the property window of *CarA*.

3) Natural Environment: Metaclass *Natural Environment* describes the natural (driving) environment, which is about atmospheric conditions, lighting conditions and weather-related road surface conditions. Based on Wise Drive [10], we capture four attributes, values of which are relatively easy to obtain and most relevant to driving as the properties of metaclass *Natural Environment*: *temperature*, *visibility*, *clouds* and *windSpeed*. Road surface conditions are described in metaclass *Road*.

4) Road: Road is the main environmental element of autonomous driving, almost all driving tasks take place on the road. Therefore, it is necessary to describe the road structure in detail when depicting driving behaviors. In UCMeta4ADS, metaclass *Road* contains three properties: *roadType*, *laneStructure* and *roadSurface*, each of which is typed with an enumeration. For instance, enumeration *roadType* defines four literals: local_road, collector_roads, arterial_roads and freeways (Fig. 2).

5) Animal: Most relevant properties of animals entering roadways are their sizes, masses, and whether they are supervised [10]. Hence, we design *height*, *weight*, and *AnimalType* as the properties of metaclass *Animal*, among which *AnimalType* is an enumeration with two literals: *wild* and *domestic*.

B. Use Cases

To structure ADS scenarios, we intensively extend metaclass *UseCase* (Fig. 3) from RUCM.

1) ADS_UC: A non-trivial autonomous driving behavior may contain a set of tasks at different levels. Hence, in RUCM4ADS, we divide driving behaviors at three levels: strategic, tactical, and operational [19]. Consequently, we

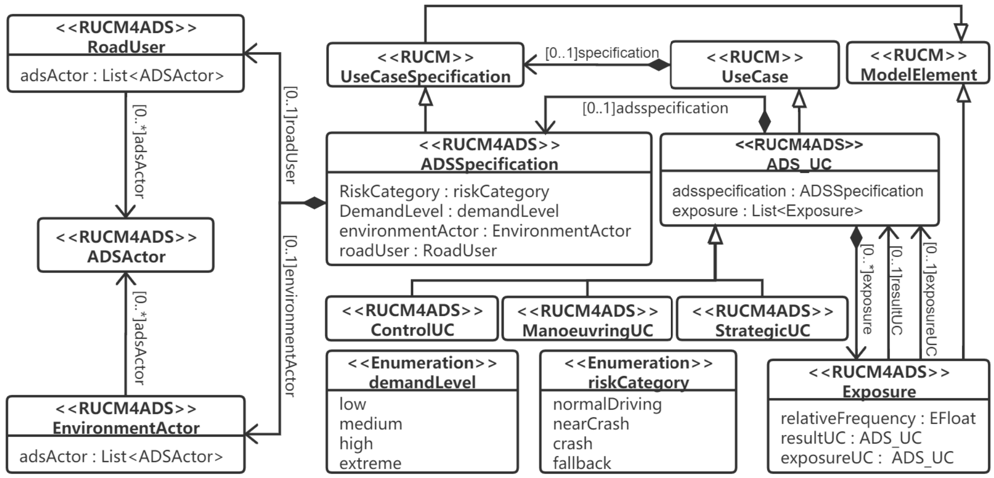

Fig. 3: Use case and use case specification metamodel

structure ADS use cases into three levels: *ControlUC*, *ManoeuvringUC*, and *StrategicUC*.

ControlUC is for describing basic motion control tasks, including longitudinal control (e.g., acceleration, deceleration, and speed maintenance) and lateral control (e.g., straight driving, cornering, and swerving). Tasks that need to consider road structures, obstacles, traffic regulations, or interactions with other road users are not within the scope of *ControlUC*. *ManoeuvringUC* focuses on behaviors of the subject vehicle in terms of its actual movements (e.g., *Maintain lane*, *Change lane* and *Turn into a same-direction traffic lane*) and its interactions with other *ADS Actor*s (e.g., *Overtake*, *Pedestrian crossing handling*) [11]. A *ManoeuvringUC* usually contains a set of basic motion control tasks and some other *ManoeuvringUC*s, which can be specified via include relationships defined in the UseCase package of the standard UML. For example, as shown in Fig. 1, in step 5 of the basic flow of *ManoeuvringUC* (i.e., *Respond to vehicles merging in rear*), use case *Accelerate* (which is of the *ControlUC* type) is included.

StrategicUC describes the general plan of the subject vehicle, usually a long time constant driving task such as *cross the tunnel* and *drive over the highway*. Flow of events of a *StrategicUC* usually contains a set of manoeuvring tasks, but should not contain control level tasks.

2) Exposure: As shown in Fig. 3, we newly introduce metaclass *Exposure*, which extends *ModelElement* of RUCM, to describe an event that enables a situation to occur [20] [21]. In ADS scenarios, one driving behavior is often triggered by another, *Exposure* in RUCM4ADS can then be used to capture such causal relationships of driving behaviors. For instances, violating a red light might be an exposure for a collision, and passing crossroads might be an exposure for running a red light. As shown in Fig. 3, metaclass *Exposure* is characterized with three attributes: *resultUC* (e.g., running a red light), *exposureUC* (e.g., passing the crossroads), *relativeFrequency* (indicating the occurrence frequency of the exposure). For instance, if we can obtain the number of red-light violations per signalized intersection entries, we can get the value of *relativeFrequency*. In RUCM4ADS, we also define the keyword EXPOSURED BY USE CASE and a specific field for invoking an exposure, as shown in *B1* of Fig. 1. *C2* of Fig. 1 is the property window of *Exposure*.

C. Use Case Specifications

We define metaclass *ADSSpecification*, which extends RUCM's *UseCaseSpecification*, as shown in Fig. 3. An ADS use case specification is characterized with two attributes: *RiskCategory* and *DemandLevel*. *RiskCategory* is typed with enumeration *riskCategory* of four literals: *normalDriving*, *nearCrash*, *crash*, and *fallback*, based on the risk category provided in [10]. ADS scenarios can also be classified according to *DemandLevel*: *low*, *medium*, *high*, and *extreme*. Inspired by [10], we introduce *demandLevel* to measure the complexity of road environment and difficulty of driving tasks. For example, driving on a sunny day with few pedestrians on clean roads should be considered as a situation with a low demand on the subject vehicle. As the complexity of road conditions increase, the demand level increases. An ADS may drive properly in low or medium demand situations, but may behave abnormally when facing high or extreme demand situations. As shown in *C3* of Fig. 1, the *DemandLevel* of use case Respond to vehicles merging in rear is *low*, the *RiskCategory* is *normalDriving*.

As shown in Fig. 3, in addition to elements from RUCM (e.g., flows of events, preconditions and postconditions), *ADSSpecification* also has *EnvironmentActor* and RoadUser, which refer to different types of ADS actors. As part of the metamodel, we define the following constraints to restrict the reference of the two metaclasses to *ADSActor*; *EnvironmentActor* can refer to *Road*, *NaturalEnvironment*, *Animal* and

OtherObstacle, while *RoadUser* can refer to *RoadHumanUser*, *VehicleUser*, and *Vehicle*.

As shown in Fig. 1, in *ADSSpecification*, the references from *EnvironmentActor* or *RoadUser* should be manually specified in fields *B2* and *B3* through keyword REF. The tool automatically populates and maintains instances of *EnvironmentActor*, *RoadUser*, and the references, as shown in *A2* and *A3* of Fig. 1.

IV. Evaluation

In this section, we present a case study we conducted, and summarize key results of a controlled experiment.

A. Case Study

Due to the complexity of autonomous driving, ADSs can have countless scenarios. To validate RUCM4ADS, we selected a set of typical ADS scenarios, together named *Drive through one block*. It is the most basic situation when driving in a city. In this scenario, the ADS controls the subject vehicle drive through a block; the driving environment includes several streets and intersections, and driving behaviors include interactions with other road users and obstacles on the road.

As shown in TABLE I, we define 1 strategic use case (i.e., *Drive through one block*), 6 manoeuvring use cases (i.e., *Maintain lane*, *Interact with road users*, *Respond to vehicles merging in rear*, *Handle obstacles*, *Interact with traffic control person*, and *Run a red light*), and 3 control use cases (i.e., *Set up forward*, *Accelerate* and *Decelerate*). Therefore, in total we specified 10 use cases, which all together contain 10 basic flows, 11 alternative flows, and 143 steps (TABLE I). There are 7 *Exposure* relationships among these use cases. We also identified and specified 11 actors, which are classified into two types: *EnvironmentActor* and *RoadUser*. These actors were referenced 33 times in the 10 use case specifications: 17 for *EnvironmentActor* and 15 for *RoadUser*.

These results give us confidence that RUCM4ADS has implemented required domain concepts for enabling effective use case modeling of autonomous driving.

B. Controlled Experiment

In a laboratory setting of a graduate student course, we conducted a controlled experiment to gain an initial understanding about the applicability of RUCM4ADS. The subjects of the experiment are 85 graduate students, who received training on RUCM and RUCM4ADS, as part of the course curriculum. The subjects were divided into two groups based on results of an assignment before the experiment. These two groups were asked to finish the same use case modeling task with the respective tool of each method. After completing the task, a comprehension questionnaire-based survey was conducted to investigate subjective opinions of the subjects on the applicability of RUCM4ADS and RUCM. Results show that 83% subjects of the RUCM4ADS group were completely and generally comfortable when using RUCM4ADS, while it was 64% for the RUCM group. Moreover, 67% RUCM subjects (completely and generally) agreed that they understood the given ADS scenario, which is however 78% for RUCM4ADS. From these preliminary results, we can see that to certain extent, RUCM4ADS can help users better structure ADS scenarios, when comparing with RUCM, and the RUCM4ADS methodology can improve users' understanding of ADS scenarios.

TABLE I: Descriptive statistics of RUCM4ADS use case specifications

ADS Use Case	**# of env. actors**	**# of RU actors**	**# of Expo -sure**	**# of basic flows**	**# of alt. flows**	**# of steps**
Drive through one block (S)	3	1	0	1	2	13
Maintain lane (M)	5	4	0	1	3	37
Handle obstacle (M)	2	0	0	1	2	11
Respond to vehicles merging in rear (M)	2	2	1	1	0	10
Run a red light (M)	2	1	2	1	1	15
Interact with traffic control person (M)	1	1	1	1	1	8
Interact with road users (M)	2	3	3	1	0	39
Set up forward (C)	0	1	0	1	0	4
Decelerate (C)	0	1	0	1	1	3
Accelerate (C)	0	1	0	1	1	3
Total	**17**	**15**	**7**	**10**	**11**	**143**

V. Related work

Use case modeling is a commonly applied requirements specification method in practice [22], which is also used to support various types of requirements analyses and generations of downstream artifacts such as UML analysis models and test cases [23] [24]. Use case modeling (especially illustrating use cases, actors, and their relations as use case diagrams) has been integrated in major modeling tools such as MagicDraw and Enterprise Architect. Moreover, in the system engineering domain, requirements diagrams of SysML are often used for requirements specification and analysis. In short, these standardized notations, which are supported by well-known main stream modeling tools, are very generic. Similarly, RUCM [4] was proposed initially as a generic methodology for use case modeling, but later on it was extended for specific purposes such as supporting automated test generation [25]. In this work, we aim to specialize RUCM for autonomous driving, and hence propose RUCM4ADS. We choose RUCM because it has been successfully applied in the automotive domain [26].

To construct RUCM4ADS, we referenced to two contributions from the literature. One is Wise Drive: Requirements Analysis Framework for Automated Driving Systems [11] [10] from the Waterloo Intelligent Systems Engineering Lab. Wise Drive provides important concepts for specifying driving tasks and road environment. The second work we referenced is [2] from which we extracted information regarding characteristics and properties of autonomous vehicles and their occupant's such as *Maximum permitted gross weight*.

In [27], a high-level taxonomy for operational design domain (ODD) is presented, which includes ODD elements such as roadway types and weather. The taxonomy offers a checklist to organize various ODD elements for autonomous driving. Many RUCM4ADS metamodel elements conform to ODD attributes such as physical infrastructure (e.g., roadway surfaces), operation constraints (e.g., vehicle speed limit) and environmental conditions (e.g., weather). Therefore, we consider that RUCM4ADS use case models have the potential to be referenced for identifying ODDs for specific ADSs.

VI. Conclusion and Future Work

Delivering high-quality ADSs requires high-quality requirements. In this paper, we present RUCM4ADS, a use case modeling approach, particularly for specifying ADS scenarios as use case models. RUCM4ADS is equipped with an editor that implements the RUCM4ADS Metamodel and facilitates the application of the RUCM4ADS methodology. We conducted preliminary evaluation of RUCM4ADS. Results show good applicability of RUCM4ADS when comparing with another generic use case modeling approach. In future work, we will conduct more case studies and controlled experiments to evaluate RUCM4ADS. We also plan to provide the capability of automatically transforming RUCM4ADS use case models into other artifacts such as test cases.

References

[1] T. Yue, H. Zhang, S. Ali, and C. Liu, "A practical use case modeling approach to specify crosscutting concerns," in *International Conference on Software Reuse*, pp. 89–105, Springer, 2016.

[2] W. Wachenfeld, H. Winner, J. C. Gerdes, B. Lenz, M. Maurer, S. Beiker, E. Fraedrich, and T. Winkle, "Use cases for autonomous driving," in *Autonomous driving*, pp. 9–37, Springer, 2016.

[3] T. Yue, L. C. Briand, and Y. Labiche, "Facilitating the transition from use case models to analysis models: Approach and experiments," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 22, no. 1, pp. 1–38, 2013.

[4] T. Yue, L. C. Briand, and Y. Labiche, "A use case modeling approach to facilitate the transition towards analysis models: Concepts and empirical evaluation," in *International conference on model driven engineering languages and systems*, pp. 484–498, Springer, 2009.

[5] H. Zhang, T. Yue, S. Ali, J. Wu, and C. Liu, "A restricted natural language based use case modeling methodology for real-time systems," in *2017 IEEE/ACM 9th International Workshop on Modelling in Software Engineering (MiSE)*, pp. 5–11, IEEE, 2017.

[6] C. Wang, F. Pastore, A. Goknil, L. Briand, and Z. Iqbal, "Automatic generation of system test cases from use case specifications," in *Proceedings of the 2015 international symposium on software testing and analysis*, pp. 385–396, 2015.

[7] M. Zhang, T. Yue, S. Ali, B. Selic, O. Okariz, R. Norgre, and K. Intxausti, "Specifying uncertainty in use case models," *Journal of Systems and Software*, vol. 144, pp. 573–603, 2018.

[8] T. Yue, *Automatically deriving a UML analysis model from a use case model*. PhD thesis, Carleton University, 2010.

[9] M. Zhang, T. Yue, S. Ali, H. Zhang, and J. Wu, "A systematic approach to automatically derive test cases from use cases specified in restricted natural languages," in *International Conference on System Analysis and Modeling*, pp. 142–157, Springer, 2014.

[10] K. Czarnecki, "Operational design domain for automated driving systems," 2018.

[11] K. Czarnecki, "Automated driving system (ads) task analysis–part 2: Structured road maneuvers," *Waterloo Intelligent Systems Engineering Lab (WISE) Report, University of Waterloo*, 2018.

[12] T. Yue, L. C. Briand, and Y. Labiche, "atoucan: an automated framework to derive uml analysis models from use case models," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 24, no. 3, pp. 1–52, 2015.

[13] O. OMG, "Meta object facility (mof) core specification," 2014.

[14] H. Elrofai, D. Worm, and O. O. den Camp, "Scenario identification for validation of automated driving functions," in *Advanced Microsystems for Automotive Applications 2016*, pp. 153–163, Springer, 2016.

[15] B. E. Porter, *Handbook of traffic psychology*. Academic Press, 2011.

[16] D. Shinar, *Traffic safety and human behavior*. Emerald Group Publishing, 2017.

[17] J. P. Rothe, "Traffic sociology: Social patterns of risk," *International journal of adolescent medicine and health*, vol. 5, no. 3-4, pp. 187–198, 1992.

[18] R. Layton and K. Dixon, "Stopping sight distance," *Kiewit Center for Infrastructure and Transportation, Oregon Department of Transportation*, 2012.

[19] J. A. Michon, "A critical view of driver behavior models: what do we know, what should we do?," in *Human behavior and traffic safety*, pp. 485–524, Springer, 1985.

[20] R. Elvik, "Bidrag til en generell teori om forholdet mellom," *Transportation Research Record*, vol. 2103, pp. 25–31, 2014.

[21] R. Elvik, A. Erke, and P. Christensen, "Elementary units of exposure," *Transportation research record*, vol. 2103, no. 1, pp. 25–31, 2009.

[22] R. Salay, M. Chechik, J. Horkoff, and A. Di Sandro, "Managing requirements uncertainty with partial models," *Requirements Engineering*, vol. 18, no. 2, pp. 107–128, 2013.

[23] T. Yue, S. Ali, and M. Zhang, "Rtcm: A natural language based, automated, and practical test case generation framework," in *Proceedings of the 2015 International Symposium on Software Testing and Analysis*, ISSTA 2015, (New York, NY, USA), p. 397–408, Association for Computing Machinery, 2015.

[24] L. Briand and Y. Labiche, "A uml-based approach to system testing," *Software and Systems Modeling*, vol. 1, no. 1, pp. 10–42, 2002.

[25] C. Wang, F. Pastore, A. Goknil, and L. Briand, "Automatic generation of acceptance test cases from use case specifications: an nlp-based approach," *IEEE Transactions on Software Engineering*, 2020.

[26] I. Hajri, A. Goknil, L. C. Briand, and T. Stephany, "Applying product line use case modeling in an industrial automotive embedded system: Lessons learned and a refined approach," in *2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, pp. 338–347, IEEE, 2015.

[27] D. NHTSA, "812 623: A framework for automated driving system testable cases and scenarios," 2018.