Keywords: ontology evaluation; metrics; shared autonomous vehicles

Tiago PRATA[1]*, Miguel MIRA DA SILVA[2], Flávia SANTORO[3], António REIS PEREIRA[4]

ASSESSING THE DEVELOPMENT OF AUTONOMOUS CARS

Summary. With recent innovations regarding autonomous vehicles and the fact that several vehicle brands have started to deploy autonomous driving functionalities, it is still unknown what these innovations may offer to social lives. Owing to the ability to autonomously drive from one location to another, the concept of shared autonomous vehicles was created to let an individual turn their assets into a source of income while other individuals could use this service without having to own a vehicle. The development of this emerging concept was aided by an evaluation of an ontology already presented regarding the topic of shared autonomous vehicles performed by three different frameworks (OQuaRE, OntoMetrics and OOPS) that generally agreed with the validity of the proposed ontology.

1. INTRODUCTION

In several locations, like Beijing (China) and Phoenix (US), autonomous vehicles are already a reality being explored under shared concepts by dedicated companies. Such a service can be the solution to several urban problems, like mobility constraints, parking unavailability and carbon emissions. [12] developed a study in the city of Lisbon measuring the effect of the replacement of existing personal cars by shared autonomous vehicles, concluding it could be possible to ensure present mobility needs with just 10% of the present car park, reducing 30% of carbon emissions and eliminating all surface parking.

The positive effects of shared autonomous vehicles are impressive. There are inhibitors, but the enablers outweigh the inhibitors and a set of pre-conditions have been suggested to expand the service to more cities [13]. Those pre-conditions can be represented by an ontology [12].

Before a product is available to its regular customer, it goes through a certain number of evaluation steps to deliver a good product to the market. The same should happen with the most diverse ontologies that are published or have plans to be published. The ontology design offers a powerful way to represent knowledge independently of the domain we are investigating allowing the developer to accurately detail any desired topic about any domain [3].

Despite its usefulness, the creation of an ontology is a hard task that requires not only a considerable amount of effort but also a great amount of time to achieve a complete and well-rounded work. Developers found a way to reuse existing ontologies to shorten their workload by modifying or continuing work from other developers who had already started an ontology about a certain topic since different the appearance of new ones was a normal happening [7].

[1] Instituto Superior Técnico; Av. Rovisco Pais 1, 1049-001 Lisbon, Portugal; e-mail: tiago.prata@tecnico.ulisboa.pt; orcid.org/0000-0002-8929-4097
[2] INOV; Rua Alves Redol 9, 1000-029 Lisbon, Portugal; e-mail: mms@tecnico.ulisboa.pt; orcid.org/0000-0002-0489-4465
[3] Inteli; Estrada Paço do Lumiar, Campus do Inteli Edif. M-8, 1600-546 Lisbon, Portugal; e-mail: flaviamariasantoro@gmail.com; orcid.org/0000-0003-3421-1984
[4] Instituto Superior Técnico; Av. Rovisco Pais, 1, 1049-001 Lisbon, Portugal; e-mail: reis.pereira@tecnico.ulisboa.pt; orcid.org/0000-0002-8408-2554
* Corresponding author. E-mail: tiago.prata@tecnico.ulisboa.pt

In the meantime, the reuse of an ontology makes it possible and more than viable to shorten the time consumed when developing an ontology. There is a need to evaluate the work previously developed to ensure there will not be a need to undo all the work done to correct some issues regarding the ontology that was used as a skeleton.

Recognizing the importance of evaluating ontologies is an essential first step. However, despite its importance, most works do not go through an evaluation phase. Depending on the ontology domain, an evaluated ontology may help the development of an area in case it is a recent topic.

To address this problem, we evaluated an ontology [11] regarding shared autonomous vehicles using different evaluation frameworks. When doing this evaluation, we pretended to document relevant mistakes and the lessons we learned by doing this evaluation in the hope that other developers could see the benefits of assessing their work.

This paper is divided into the following sections. Section 2 gives a theoretical background of ontologies and their evaluation. Section 3 enumerates and describes the different frameworks used to evaluate a selected ontology regarding shared autonomous vehicles. Section 4 briefly presents the topic of the ontology evaluated, the results obtained from the evaluation and our thoughts about the obtained results. Section 5 concludes the paper.

2. RESEARCH BACKGROUND

In this section, we introduce the concepts of ontology and ontology evaluation, which are the main concepts of this paper.

2.1. Ontology

An ontology is an explicit specification of a conceptualization [10]. Due to the large community of developers who use ontologies and the different candidates of ontologies about a certain topic, it is easy to understand why ontologies are a good choice to represent knowledge [3].

Despite the flexibility and accuracy ontologies offer, they have a great drawback, which is the time it takes to create them from scratch. Not only do ontologies require a lot of attention and time, but they also require the developer to spend a great amount of time researching to avoid inconsistencies that may cause an ontology to be discarded by other developers who might be looking for ontologies to reuse for their work [2].

2.2. Ontology Evaluation

As previously mentioned, ontologies require a lot of time and effort to be built and even though a result may be obtained, it does not mean it has no flaws. Developers have an interest to design ontologies with high quality, few inconsistencies, and a reduced amount of errors.

The evaluation phase on an ontology development cycle is recommended to ensure that the mentioned problems do not happen because not only will it give the developer a way to assess the quality of their work, but it will also warn the developers about possible common mistakes [4, 8]. Having an ontology with few to no inconsistencies and mistakes regarding the semantics and syntax will not only improve the quality of the work performed but also enable that work to be reused by other developers who may want to continue to work on the same subject.

Different frameworks were proposed to evaluate ontologies. However, other frameworks assess different aspects of the ontology due to the necessity of their proposals. Depending on their needs, developers can choose from a handful of frameworks that can evaluate either sets of ontologies, making it easier for them to choose one from a set, or a single ontology to assess its quality [1].

3. EVALUATION FRAMEWORKS

In this section, we explain how the frameworks used to evaluate the ontology we had in hand work and which aspects they try to evaluate in an ontology.

As already mentioned, an ontology should have an evaluation phase not only to guarantee it has almost no errors that may affect the overall quality of the ontology but also to ensure that it can be reused by any other developer without the need to correct any possible errors.

Three different frameworks were chosen to evaluate our ontology to complement the results of each and evaluate specific aspects of the ontology like common mistakes normally committed by people who are new to the ontology world. These frameworks were: OOPS, OntoMetrics and OQuaRE (displayed in Table 1). We will dive into their functionalities in this section.

Table 1

Frameworks Used

Frameworks	Main Purpose	Metrics Used	References
OOPS	Detect different types of bad practices when developing an ontology		[8]
OntoMetrics	Metrics calculation to assess certain characteristics of an ontology	Schema Metrics Graphic Metrics Knowledgebase Metrics Class Metrics	[7]
OQuaRE	Evaluate the quality of an ontology based on SQuaRE	Graphic Metrics	[5, 6, 9]

Evaluating an ontology can be a difficult task since the developer may need to install different plugins that may not work in newer versions of ontology editors, leaving the developer to try different versions of the same plugin without any success [8].

With this in mind, OOPS was made available online to eliminate the trial-and-error part of finding the correct plugin. OOPS is a different framework from the usual framework since it does not calculate quantitative metrics. Intending to help newer developers, this framework tries to identify bad practices presented in the ontology that is being evaluated by consulting a catalog of pitfalls that were collected after manually reviewing several ontologies.

To use this framework, the user needs to first access the website of the framework and give the ontology as input either by giving the ontology URI or the OWL code. Before the scan of pitfalls begins, the user is given the choice to select which pitfalls he wants to be looked for, either by groups or by specific pitfalls. The scanner phase has two sub-phases: the first is locating the different pitfalls the framework has in its catalog. In case a pitfall is detected, the second sub-phase kicks in and a suggestion is formulated to help the developer correct the pitfall.

At the end of the scanning phase, a list with all the pitfalls is given to the user, informing them of several aspects of the pitfalls, namely the name, a detailed explanation, the number of occurrences and the level of importance of that pitfall (critical, important and minor). A badge regarding detected pitfalls is also provided so that users can add it to the documentation of the ontology. There are four types of badges: Free of Pitfalls, Minor Pitfalls Detected, Important Pitfalls Detected and Critical Pitfalls Detected. Despite the need to evaluate ontologies to ensure the completeness of the work done by the developer, there is no framework that developers recognize as better than the others, so we decided to use more than one framework to have more than one set of results and to be able to compare their results.

The second framework used to evaluate the ontology we had in hand was the OntoMetrics framework. Like the previous one, OntoMetrics is available online and works as a metric calculator for developers. OntoMetrics was not only made available online and recognized the common ontology formats RDF and OWL to avoid problems regarding the installation and old plugins that might not work in newer development environments. In case the developer prefers, OntoMetrics also offers the possibility of uploading the ontology via URL or even copying the ontology code and pasting it on the website. As mentioned before, OntoMetrics is meant to be a metric calculator, so it evaluates any ontology with four types of metrics: Schema, Graph, Knowledgebase and Class plus the OWL-API counting metrics [7]. The metrics are displayed in Table 2.

Since this framework was designed to offer developers an efficient and effective way to evaluate their work by evaluating different aspects of any ontology due to the four types of metrics it evaluates, we chose to select this framework to be part of the evaluation phase of the ontology we had in hand.

OQuaRE was selected as the third framework to evaluate the ontology that was at our disposal to give us another set of results so that we could compare the results obtained from the previous framework to those obtained from this one.

OQuaRE was inspired by an already existing framework, SQuaRE. Like the other evaluation methods, this framework pretends to help ontology developers improve their work. Since it is inspired by another framework, OQuaRE inherited some qualities from that framework, including the ability to recognize the following characteristics: reliability, operability, maintainability, compatibility, transferability and functional adequacy [6]. This framework also offers developers the ability to recognize the quality characteristics of structural, performance efficiency, and quality to give developers a reason to use OQuaRE instead of SQuaRE. The implementation used from OQuaRE at the time of writing was composed of the 19 different metrics [6] described in Table 3. This version of OQuaRE uses scaling functions to calculate quality scores from the metrics results so that the results can be easily interpreted [9].

Table 2

OntoMetrics Metrics [7]

Schema Metrics	**Graph Metrics**	**Knowledgebase Metrics**	**Class Metrics**
Attribute Richness	Absolute Root Cardinality	Average Population	Class Connectivity
Inheritance Richness	Absolute Leaf Cardinality	Class Richness	Class Fullness
Relationship Richness	Absolute Sibling Cardinality		Class Importance
Attribute-Class Ratio	Absolute Depth		Class Inheritance Richness
Equivalence Ratio	Average Depth		Class Readability
Axiom Class Ratio	Maximal Depth		Class Relationship Richness
Inverse Relations Ratio	Absolute Breadth		Class Children
Class Relation Ratio	Average Breadth		Class Instances
	Maximal Breadth		Class Properties
	Ratio of Leaf Fan-outness		
	Ratio of Sibling Fan-outness		
	Tangledness		
	Total Number of Paths		
	Average Number of Paths		

4. SHARED AUTONOMOUS VEHICLES ONTOLOGY

With the recent innovation in autonomous vehicles, a reduction in road congestion might be expected [17]. However, with the deployment of autonomous vehicles, many aspects that can increase road congestion instead of reducing it tend to be often forgotten [18]. Some of these aspects are the lack of possibilities to acquire an autonomous vehicle [19], or the use of autonomous vehicles leaving them near their owners which will end up creating congestion in corporate places.

One way to avoid the creation of unnecessary congestion is the use of shared autonomous vehicles. Adopting the use of shared autonomous vehicles could decrease local congestion since other people from different areas may summon those vehicles for their use, thus decreasing the number of vehicles in a certain area, which will reduce congestion since the vehicles will always be circulating instead of being parked.

The ontology we have in hand tries to demonstrate the utility of shared autonomous vehicles and show the impact of this solution not only in an urban scenario but also from a governmental point of view. This ontology is composed of two sub-ontologies.

The first one describes the ecosystem efficiency, while the second one describes the social impact in society by shared autonomous vehicles [20]. These sub-ontologies' diagrams can be seen in Fig. 1 and Fig. 2 and are concentrated on the reality impacts of the deployment of such a service.

Some of the concepts, like the sharing concepts (Mobility-as-a-Service, carsharing, ridesharing and ridepooling) or traffic optimization, electrical vehicles/low emissions, and vehicle improvements in general, are appliable either to autonomous or non-autonomous cars and reflect an overall trend that can be perceived in non-autonomous cars.

Table 3

OQuaRE Metrics [5, 6, 9]

Metrics	Description
ANOnto	Mean Number of Annotation Properties per Class
AROnto	Number of Restrictions of the Ontology per Class
CBOnto	Number of Direct Ancestors of Classes Divided by the Number of Classes Minus Subclasses of Thing
CBOnto2	Mean Number of Direct Ancestors per Class
CROnto	Mean Number of Individuals per Class
DITOnto	Length of the Longest Path from Thing to a Leaf Class
INROnto	Mean Number of Subclasses Per Class
LCOMOnto	Mean Length of All the Paths from Leaf Classes to Thing
NACOnto	Mean Number of Superclasses per Leaf Class
NOCOnto	Number of the Direct Subclasses Divided by the Number of Classes Minus the Number of Leaf Classes
NOMOnto	Mean Number of Object and Data Property Usages per Class
POnto	Mean Number of Direct Ancestors per Class
PROnto	Number of Subclasses of Relationships Divided by the Number of Subclasses of Relationships and Properties
RROnto	Number of Usages of Object and Data Properties Divided by the Number of Subclasses of Relations and Properties
RFCOnto	Number of Usages of Object and Data Properties and Superclasses Divided by the Number of Classes
TMOnto	Mean Number of Classes with More Than 1 Direct Ancestor
TMOnto2	Mean Number of Direct Ancestors of Classes with More Than 1 Direct Ancestor
WMCOnto	Mean Length of the Path from Thing to a Leaf Class
WMCOnto2	Mean Number of Paths from Thing to a Leaf Class per Leaf Class

To complement the information regarding each class, properties were added to the most general class. These properties describe not only the attributes but also the restrictions of each class. The defined properties can be seen in Fig. 3.

5. EVALUATION RESULTS

In this section, we present the results obtained from the three different frameworks used to evaluate the ontology we had in hand.

5.1. Ontometrics Results

As presented in the previous section, this framework presents four different types of metrics calculated to understand the quality of the ontology elements, the design of the ontology and its structure.

The calculation results are presented in the Tables 4 and 5. In the Class Metrics, only the metrics related to the Number of Children and the Class Inheritance Richness presented calculated values, with the results from this last metric fluctuating between 0 and 40. The rest of the metrics present a calculation with 0 having been omitted.

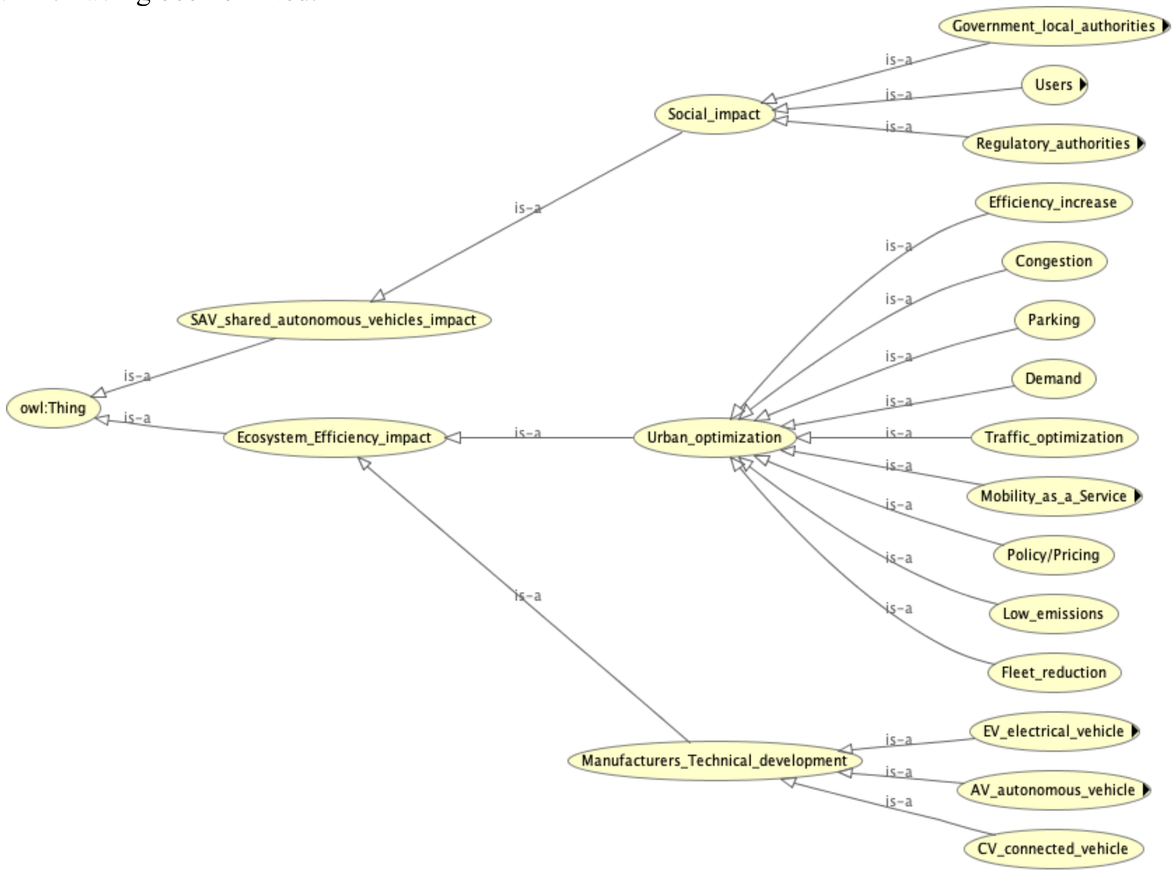


Fig. 1. Shared Autonomous Vehicles Ecosystem Efficiency

Table 4
Schema Metrics from OntoMetrics

Metrics	Metric Value
Attribute Richness	0.05
Inheritance Richness	1.075
Relationship Richness	0.338462
Axiom/Class Ratio	4.275
Class/Relation Class	0.615385

5.2. OOPS Results

OOPS was designed to help developers find possible mistakes in their ontologies. Thus, the results are different from the ones shown in the previous sub-section. Instead of calculating several metrics, pitfalls were looked for and returned to help the developer make their work more complete. The detected pitfalls are presented in Table 6.

5.3. OQuaRE Results

OQuaRE calculates different metrics to ensure that an ontology is rich for the set of characteristics mentioned in the previous section. After calculating the different metrics, this framework applies to scale functions to turn the metrics' values into quality scores between 1 and 5. The metrics' values and the corresponding quality scores are presented in Table 7.

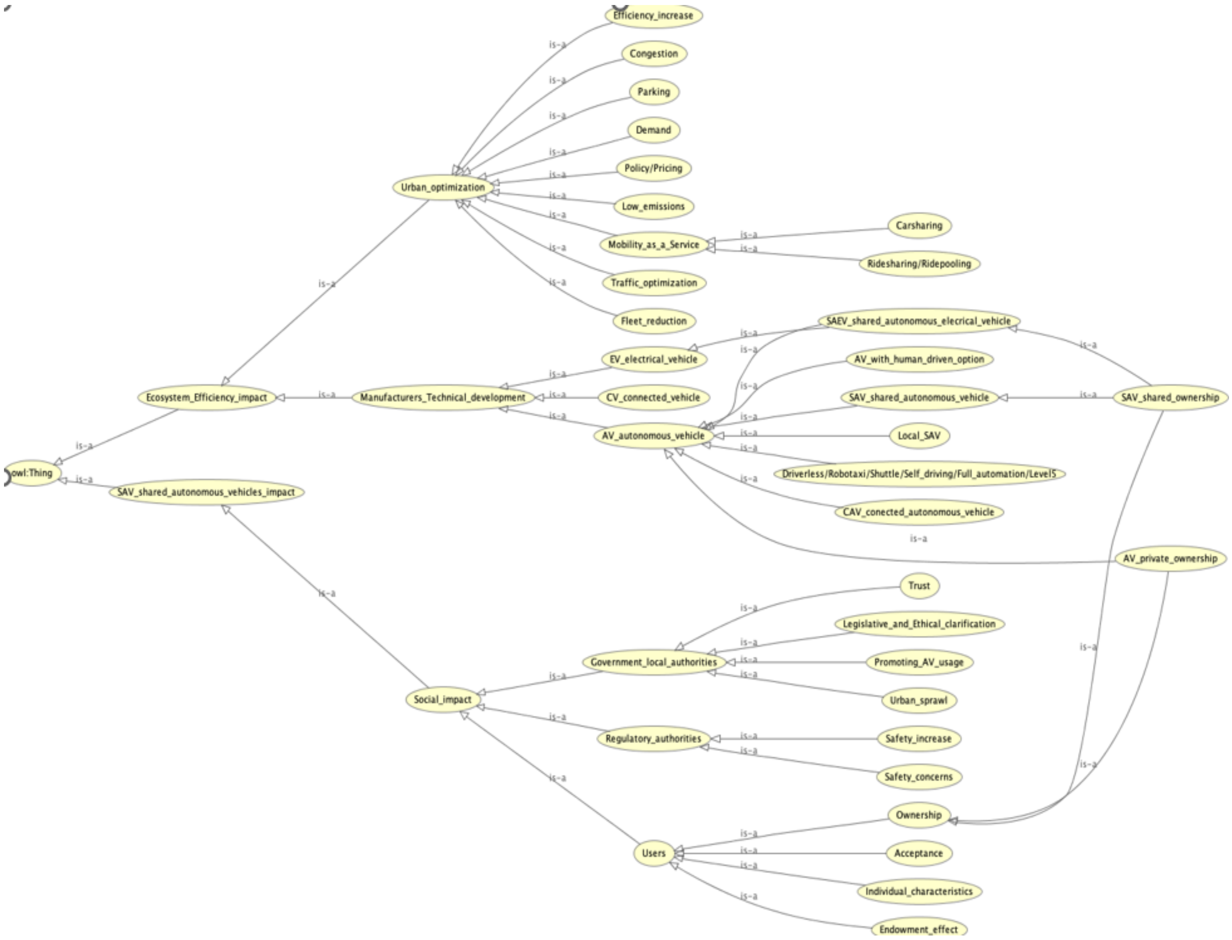
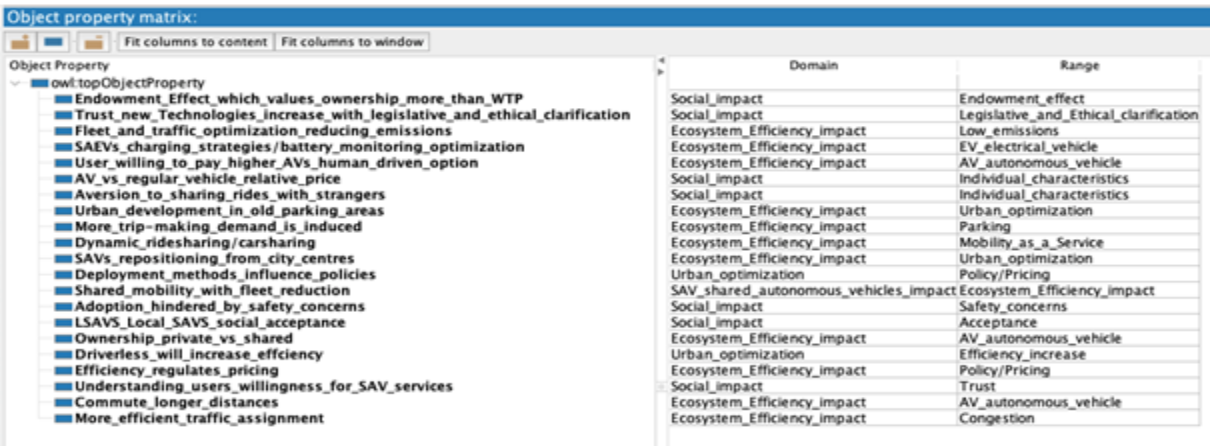


Fig. 2. Shared Autonomous Vehicles Social Impact


Object property matrix:

Fit columns to content | Fit columns to window

Object Property	Domain	Range
owl:topObjectProperty		
Endowment_Effect_which_values_ownership_more_than_WTP	Social_impact	Endowment_effect
Trust_new_Technologies_increase_with_legislative_and_ethical_clarification	Social_impact	Legislative_and_Ethical_clarification
Fleet_and_traffic_optimization_reducing_emissions	Ecosystem_Efficiency_impact	Low_emissions
SAEVs_charging_strategies/battery_monitoring_optimization	Ecosystem_Efficiency_impact	EV_electrical_vehicle
User_willing_to_pay_higher_AVs_human_driven_option	Ecosystem_Efficiency_impact	AV_autonomous_vehicle
AV_vs_regular_vehicle_relative_price	Social_impact	Individual_characteristics
Aversion_to_sharing_rides_with_strangers	Social_impact	Individual_characteristics
Urban_development_in_old_parking_areas	Ecosystem_Efficiency_impact	Urban_optimization
More_trip-making_demand_is_induced	Ecosystem_Efficiency_impact	Parking
Dynamic_ridesharing/carsharing	Ecosystem_Efficiency_impact	Mobility_as_a_Service
SAVs_repositioning_from_city_centres	Ecosystem_Efficiency_impact	Urban_optimization
Deployment_methods_influence_policies	Urban_optimization	Policy/Pricing
Shared_mobility_with_fleet_reduction	SAV_shared_autonomous_vehicles_impact	Ecosystem_Efficiency_impact
Adoption_hindered_by_safety_concerns	Social_impact	Safety_concerns
LSAVS_Local_SAVS_social_acceptance	Social_impact	Acceptance
Ownership_private_vs_shared	Ecosystem_Efficiency_impact	AV_autonomous_vehicle
Driverless_will_increase_effciency	Urban_optimization	Efficiency_increase
Efficiency_regulates_pricing	Ecosystem_Efficiency_impact	Policy/Pricing
Understanding_users_willingness_for_SAV_services	Social_impact	Trust
Commute_longer_distances	Ecosystem_Efficiency_impact	AV_autonomous_vehicle
More_efficient_traffic_assignment	Ecosystem_Efficiency_impact	Congestion

Fig. 3. Shared Autonomous Vehicles Concept Properties

6. LESSONS LEARNED

After evaluating different methods, an analysis of the results had to be done to understand the quality of the ontology. As can be seen from the results displayed in the tables above, some common mistakes (pitfalls) are present in the ontology but are not critical mistakes that can harm the overall quality of the developed work. Most of the mistakes detected were considered minor and had low consequences on the work.

Table 5

Graph Metrics from OntoMetrics

Metric	Metric Value
Absolute Root Cardinality	1
Absolute Leaf Cardinality	26
Absolute Sibling Cardinality	40
Absolute Depth	186
Average Depth	4.133333
Maximal Depth	6
Absolute Breadth	45
Average Breadth	2.8125
Maximal Breadth	9
Ratio of Leaf Fan-Outness	0.65
Ratio of Sibling Fan-Outness	1.0
Tangledness	0.075
Total Number of Paths	45
Average Number of Paths	7.5

Table 6

Pitfalls detected by OOPS

Pitfalls	Number of Occurrences	Importance Level
P07: Merging Different Concepts in the Same Class	1	Minor
P08: Missing Annotations	62	Minor
P13: Inverse Relationships not Explicitly Declared	20	Minor
P22: Using Different Naming Conventions in the Ontology	Ontology in general	Minor
P41: No License Declared	Ontology in general	Important

Regarding the results obtained from the metric calculations, from OquaRE, it is possible to see that the quality scores present high values, and the overall quality score of 4.105 out of 5, which is a positive result according to the scaling, by which a score of 5 represents “Exceeds Requirements” [9]. From OntoMetrics, the evaluation presented the calculation which indicates the ontology has good levels of understandability and possibly accuracy. From these metrics, we were able to infer such qualities due to the different depths and breadths of the ontology, which are not too extensive and could lead to confusion, as well as through the different values of richness, especially inheritance, which is usually a good indicator of the type of knowledge presented, either covering one single topic in depth or covering a great variety of topics with less depth but while connecting different topics.

With an evaluation focused only on the semantics and structure of the work developed, it is possible to gain an understanding of how well-elaborated the work is around specific concepts. Assessing the quality of the work developed gives ontologists an insight into their work by knowing if specific concepts are well developed or not. It can also alert new ontologists to possible pitfalls they might have committed.

Table 7
Metrics from OQuaRE

Metrics	Metric Value	Quality Score
ANOnto	0.0	1
AROnto	10.512195	5
CBOnto	1.073170	5
CBOnto2	1.073170	5
CROnto	0.0	1
DITOnto	6.0	3
INROnto	1.073170	5
LCOMOnto	3.866667	4
NACOnto	1.115385	5
NOCOnto	2.933333	5
NOMOnto	1.682927	5
POnto	2.756098	5
PROnto	0.610619	4
RROnto	0.075	5
RFCOnto	2.333333	3
TMOnto	1.024390	5
TMOnto2	0.389380	2
WMCOnto	4.461538	5
WMCOnto2	1.153846	5

7. CONCLUSIONS

An evaluation was performed on a work developed around the topic of shared autonomous vehicles to demonstrate the utility of evaluating ontologies. Overall, the evaluation presented very positive results. Despite being of high quality, the evaluated work presented some pitfalls that could be easily rectified to increase the quality of the work.

The initial ontology was based on scientific articles focusing on shared autonomous vehicles, evaluated with Cruize (GM) in San Francisco [11].

In fact, real (and commercial) trials are already underway, especially in the US [14] and in China [15]. In the US, these trials started in Phoenix and were then extended to San Francisco. It was already announced by Waymo (Google) that Los Angeles is the next step [16], which means that this type of mobility service is being deployed step by step.

Even though the present evaluation presented good results, this work has limitations. Since the topic of shared autonomous vehicles is a rapidly increasing topic with new findings being reported with a high frequency, this work may become outdated relatively quickly.

In future work, comparisons and analyses should be made with the current state of shared autonomous vehicles in order the ensure the precision of the ontology, which means that scientific works that ground the built of the ontology should be updated with new findings, which, in theory, can change the ontology at least partially, if not in full, which implies the need to evaluate the updated ontology again.

References

1. Verma, A. An abstract framework for ontology evaluation. In: *2016 International Conference on Data Science and Engineering (ICDSE)*. 2016. P. 1-6. DOI: 10.1109/ICDSE.2016.7823945.

2. Hlomani, H. & Stacey D.A. Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. *Semantic Web Journal*. 2014. P. 1-5.
3. Ensan, F. & Du, W. A semantic metrics suite for evaluating modular ontologies. *Inf. Syst*. 2013. Vol. 38(5). P. 745-770.
4. García, J. & García-Peñalvo, J. & Therón, R. A survey on ontology metrics. In: Lytras M.D. & Ordonez De Pablos, P. & Ziderman, A. & Roulstone, A. & Maurer, H. & Imber, J.B. (eds.) Knowledge Management, Information Systems, E-Learning, and Sustainability Research. WSKS 2010. *Communications in Computer and Information Science*. Springer. Berlin, Heidelberg. 2010. P. 22-27.
5. Duque-Ramos, A. & Boeker, M. & Jansen, L. & Schulz, S. & Iniesta, M. & Fernndez-Breis, J. Evaluating the good ontology design guideline (GoodOD) with the ontology quality requirements and evaluation method and metrics (OQuaRE). *PLoS ONE*. 2014. Vol. 9. No. e104463.
6. Duque-Ramos, A. & Fernndez-Breis, J. & Iniesta, M. & Dumontier, M. & Egaa Aranguren, M. & Schulz, S. & Aussenac-Gilles, N. & Stevens, R. Evaluation of the OQuaRE framework for ontology quality. *Expert Systems with Applications*. 2013. Vol. 40. P. 2696-2703.
7. Lantow, B. OntoMetrics: Putting Metrics into Use for Ontology Evaluation. In: *KEOD*. 2016. P. 186-191.
8. Poveda-Villalón, M. & Gómez-Pérez, A. & SuárezFigueroa, M.C. OOPS! (OntOlogy Pitfall Scanner!): an on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems*. 2014. Vol. 10(2). P. 7-34. DOI: 10.4018/ijswis.2014040102.
9. Quesada-Martínez, M. & Duque-Ramos, A. & Iniesta-Moreno, M. & FernándezBreis, J.T. Preliminary analysis of the OBO foundry ontologies and their evolution using OQuaRE. *Stud. Health Technol. Inform*. 2017. Vol. 235(1). P. 426-430.
10. Gruber, T. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies.* 1995. Vol. 43. P. 907-928.
11. Sousa Boieiro, P. & Silva, M.M. & Santoro, F.M. & Reis Pereira, A.R. Shared autonomous vehicles ontology. In: *Digital Technologies and Transformation in Business, Industry and Organizations.* Cham: Springer International Publishing. 2022. P. 51-78.
12. Martinez, L.M. & Viegas, J.M. Assessing the impacts of deploying a shared self-driving urban mobility system: An agent-based model applied to the city of Lisbon, Portugal. *International Journal of Transportation Science and Technology.* 2017. Vol. 6(1). P. 13-27.
13. Sousa Carneiro, P.D. & Reis Pereira, A. & Silva, M.M.D. A model for the deployment of shared autonomous vehicles in urban areas based on the research literature. *International Journal of Transport Development and Integration.* 2023. Vol. 7(3). P. 199-213.
14. Chatman, D.G. & Moran, M. *Autonomous vehicles in the United States: Understanding why and how cities and regions are responding*. 2019. UC Berkeley. Report No. UC-ITS-2019-13.
15. Pizzuto, L. & Thomas, C. & Wang, A. & Wu, T. *How China will help fuel the revolution in autonomous vehicles*. McKinsey & Company. January, 2019.
16. Waymo. *Arriving soon in the City of Angels*. Available at: https://waymo.com/waymo-one-los-angeles/.
17. Fan, Q. van Essen, J.T. & Correia, G.H. Optimising fleet sizing and management of shared automated vehicle (SAV) services: A mixed-integer programming approach integrating endogenous demand, congestion effects, and accept/reject mechanism impacts. *Transportation Research Part C: Emerging Technologies*. 2023. Vol. 157. No. 104398.
18. Metz, D. Developing policy for urban autonomous vehicles: Impact on congestion. *Urban Science.* 2018. Vol. 2(2). No. 33.
19. Delle Site, P. Pricing of connected and autonomous vehicles in mixed-traffic networks. *Transportation Research Record.* 2021. Vol. 2675(5). P. 178-192.
20. Narayanan, S. & Chaniotakis, E. & Antoniou, C. Shared autonomous vehicle services: A comprehensive review. *Transportation Research Part C: Emerging Technologies.* 2020. Vol. 111. P. 255-293.

Received 11.10.2022; accepted in revised form 09.03.2024