applied sciences

MDPI

Article

Design and Implementation of an Ontology for Semantic Labeling and Testing: Automotive Global Ontology (AGO)

Itziar Urbieta *, Marcos Nieto, Mikel García and Oihana Otaegui

Vicomtech, Parque Científico y Tecnológico de Gipuzkoa, Mikeletegi Pasealekua, 57, 20009 Donostia-San Sebastian, Spain; mnieto@vicomtech.org (M.N.); mgarcia@vicomtech.org (M.G.); ootaegui@vicomtech.org (O.O.)
* Correspondence: iurbieta@vicomtech.org

Abstract: Modern Artificial Intelligence (AI) methods can produce a large quantity of accurate and richly described data, in domains such as surveillance or automation. As a result, the need to organize data at a large scale in a semantic structure has arisen for long-term data maintenance and consumption. Ontologies and graph databases have gained popularity as mechanisms to satisfy this need. Ontologies provide the means to formally structure descriptive and semantic relations of a domain. Graph databases allow efficient and well-adapted storage, manipulation, and consumption of these linked data resources. However, at present, there is no a universally defined strategy for building AI-oriented ontologies for the automotive sector. One of the key challenges is the lack of a global standardized vocabulary. Most private initiatives and large open datasets for Advanced Driver Assistance Systems (ADASs) and Autonomous Driving (AD) development include their own definitions of terms, with incompatible taxonomies and structures, thus resulting in a well-known lack of interoperability. This paper presents the Automotive Global Ontology (AGO) as a Knowledge Organization System (KOS) using a graph database (Neo4j). Two different use cases for the AGO domain ontology are presented to showcase its capabilities in terms of semantic labeling and scenario-based testing. The ontology and related material have been made public for their subsequent use by the industry and academic communities.

Keywords: semantics; ontology; scenario-based testing; AD; ADAS; labeling; graph database; Neo4j

Citation: Urbieta, I.; Nieto, M.; García, M.; Otaegui, O. Design and Implementation of an Ontology for Semantic Labeling and Testing: Automotive Global Ontology (AGO). *Appl. Sci.* **2021**, *11*, 7782. https://doi.org/10.3390/app11177782

Academic Editors: Miguel Clavijo, Felipe Jiménez and Jose Eugenio Naranjo

Received: 3 August 2021
Accepted: 22 August 2021
Published: 24 August 2021

1. Introduction

Labeling, in the context of Artificial Intelligence (AI), is the process of adding descriptive information to data; for example, collections of images, data series, or sensor measurements. Labeling is a major bottleneck for machine learning (ML) progress, because the current state of the art regarding deep learning (DL) implies creating massive training datasets composed of data samples (e.g., images and point cloud scans). These samples are labeled with descriptions that are learnt by the DL model (e.g., labels with the class name, or the bounding box or shape of objects). As a consequence, better models are typically obtained using larger and carefully crafted datasets, which encompass, as much as possible, all the potential data variability of the domain of interest (e.g., all possible configurations of complex driving scenes).

As a result, business models for labeling have emerged, in which label producers offer services to create large datasets from raw data recordings. As an example, from 2018 to 2020, almost 20 very large open datasets were released for the development of Autonomous Driving (AD) technologies (see nuScenes (https://www.nuscenes.org/ accessed on 14 March 2021), Lyft Level 5 (https://level5.lyft.com/dataset/ accessed on 14 March 2021), H3D Honda Dataset (https://usa.honda-ri.com/H3D accessed on 14 March 2021), Waymo (https://waymo.com/open/ accessed on 14 March 2021), Audi A2-D2 (https://www.a2d2.audi/a2d2/en.html accessed on 14 March 2021), Berkeley Deep-Dive (BDD) (https://bdd-data.berkeley.edu/ accessed on 14 March 2021), Apolloscape (http://apolloscape.auto/

accessed on 14 March 2021), or Mapillary Vistas (https://www.mapillary.com/dataset/vistas accessed on 14 March 2021)). These approaches solve the short-term need for labels in DL training stages, but create new challenges for long-term operation: evolution of label types, class taxonomies, and hierarchies; fusion or comparison of labels from heterogeneous sources; scene management and searching via complex queries, etc. Other problems include the use of human languages and synonyms, different taxonomies (hierarchies of classes), and different types of attributes and properties. In turn, this leads to the problem of having to develop a new model for each dataset. Hence, dataset labeling approaches require representation of the data linked to ontology-based semantics [1].

Analogously, scenario-based testing requires the ability to unambiguously describe scenes from an imperative perspective, i.e., to command data generation or implementation in testing methods such as simulation environments or field testing. Scenario representations must refer to terms that correspond to classes defined in a Knowledge Organization System (KOS) to guarantee repeatability of the simulation and traceability of the results. Large databases of scenarios must be explored via queries, to enable extraction of particular data subsets of interest for their consumption for testing specific AD functions. Semantic connection between elements of a scenario description can lead to semantic querying, i.e., the ability to find hidden or non-explicit information using rules or reasoning-enabled query mechanisms.

Furthermore, data science has gained importance in recent years due to the steady increase in the quantity of collected data. Data has evolved to become crucial for strategic decision making or situation awareness-based systems. Similarly, data-driven DL approaches are emerging in the automotive industry. These developments require higher-level semantics to enhance the performance, functionality, and scope of trained models, enabling reasoning mechanisms with action recognition or scene understanding.

In this paper we present our approach for the construction of the *Automotive Global Ontology (AGO)* using graph databases to enable semantic services for the automotive domain. The novelty of our approach in the proposed methodology is the construction of such an ontology from diverse, heterogeneous data sources and taxonomies (e.g., from large existing Advanced Driver Assistance Systems (ADASs) and AD open datasets). The proposed method ensures a global list of concepts is created, which is linked to the existing source concepts but also generalizes well for future datasets. AGO addresses the need to provide meaning to labels and scenario descriptions by including high-level semantics in the knowledge base. Thus, the format of data is detached from the meaning, and standard practices can be safely used, such as the Video Content Description (VCD) (http://vcd.vicomtech.org accessed on 23 August 2021) language or the upcoming ASAM OpenLabel (https://www.asam.net/active-projects/ accessed on 14 June 2021) standard. The VCD structure permits labeling an entire scene in a single file, including actions, attributes, and relations, in the form of RDF triples [2], which makes it ideal for the considered use cases.

This paper is organized as follows: existing semantic resources in the field are studied in Section 2; Section 3 presents the terminology defined for this work, and Section 4 introduces the AGO domain model. Section 5 details the methodology and the ontology construction process. Section 6 exemplifies AGO utilization for the two defined use cases, with the produced results presented in Section 7.

2. State of the Art

Different works in the transportation field have identified the importance of domain-knowledge structures for different purposes: to assess traffic scenes in real time applications [3], to provide automatic support for design and analysis of performance monitoring systems for Public Transport Systems [4], or to infer knowledge to aid test management [5].

Several ontologies in the transportation domain can be found in the literature, e.g., the Ontology of Traffic Networks (OTN) [6], which is summarized as a direct encoding of

geographic data files (GDF) in OWL, and the Transport Disruption ontology [7], which provides a formal framework for modeling travel planning-related events.

In recent years, the focus has been on knowledge-based approaches representing scenarios with the purpose of promoting the scenario-based evaluation of ADASs and AD. Ontologies have become a key component for formalizing this knowledge. An event-based scenario description for testing was presented [8] based on the three abstraction levels for scenario description: functional, logical, and concrete scenarios, as described in [9] and adopted by the Pegasus project (https://www.pegasusprojekt.de/en/ accessed on 5 April 2021). The strategy was enhanced with a procedure that enabled qualitative description for the generation of more concrete scenarios. This scenario-based testing strategy entails the development of a structured knowledge and formal scenario representation language for driving simulation environments, such as, OpenSCENARIO (https://www.asam.net/project-detail/asam-openscenario-v1x/ accessed on 4 June 2021), CommonRoad (https://commonroad.in.tum.de/ accessed on 4 June 2021), or the Safety Pool's scenario description language (https://www.safetypool.ai/ accessed on 20 June 2021).

Despite these advances, there is currently a lack of an open knowledge base in the automotive domain that covers the needs of the testing and labeling applications. Therefore, one of the aims of this paper is to present AGO. This was built with the purpose of formalizing the terminology used for representing automotive scenarios and providing the required knowledge layer to support semantic-labeling tools.

A small number of studies have been published related to ontology engineering methodologies that present design and construction principles. However, to the best of our knowledge, there is not yet a standardized approach or any formal requirement other than the ontology languages defined by the W3C group [10,11] to define an ontology. In the following, a selection of existing works is discussed. METHONTOLOGY [12], On-To-Knowledge Methodology (OTKM) [13], and DILIGENT [14] constitute the basis for many subsequent proposals. For instance, NeOn [2] emphasizes the reuse of existing resources for building a collaborative ontology rather than starting from scratch, such as in METHONTOLOGY [12].

In relation to the reusable resources related to the automotive domain, in 2018 a survey of existing ontologies for transportation was carried out [15] in which several approaches were studied and compared. Among these, the ontology for road traffic management [16] presents bidirectional axioms ("doesAction" and "isActionDoneBy") that relate classes to driving actions. Other examples also consider attributes by introducing additional axioms into the ontology.

UPON [17] was published in 2005 as a proposal that takes advantage of the Unified Software Development Process and the Unified Modeling Language (UML). The proposed methodology is based on the semantic languages created by the World Wide Web (W3C), RDF, and OWL (see Section 5.2. for further information). These XML-based syntaxes allow the representation of knowledge as triples, i.e., a three-entity statement in the form of subject–predicate–object expressions. This atomic structure forms a directed graph; hence, ontologies can be defined as graphs, in which each class is a node (vertex), and is connected to other classes via properties or relations (edges). Some studies have considered the use of graph databases to implement RDF stores [18] or to build an ontology for an automated vehicle's context model [19].

However, there is not yet a de jure standard for the construction or design of ontologies in the automotive domain. Furthermore, the proposed methodology bases the construction and representation of the ontologies in graph databases. Among the analyzed references, to the best of our knowledge, sound developments based on graphs do not exist. Thus, is no dominant, standardized methodology based on these databases. In this work, we used Neo4j as the graph database to host the ontology and the Cypher query language to interoperate with it. This database deploys the ontology as a database resource, fostering its utilization to address the new challenges of the labeling industry, such as global networking

(e.g., using the Bolt network protocol for client–server communication), Big Data, advanced algorithms (e.g., pathfinding), and visualization applications.

3. Terminology

In this work, the following definitions were adopted:

- Action: A class understood to be a situation with a semantic meaning, happening in the scene typically related to Objects, which are either the subjects or the objects of the Action. They occur during a specific time interval (frames). It is necessary to distinguish between intransitive and transitive actions because they are semantically different.
    - Intransitive action: Express status of Objects, and thus can be expressed as adjectives or verbs in the present continuous form: "the car is parked". In this example, the object is not specified, but the "Car" is known to be the subject and "Parked" is the predicate of the sentence.
    - Transitive action: Can be naturally treated as triples, where there is a subject, a predicate, and an object. For example, "a child is running in the park", where "Child" is the subject, "Park" is the object, and "Running" is the predicate or the Action.
- Attribute: A quality or feature of a class element or axiom of the ontology.
- Axiom (Relationship): Statements that are asserted to be true in the domain being described [20]. They structure the ontology and provide semantic information. They are represented as relationships in the graph database.
- Class: Concept of the domain represented as a node in the graph database.
- Context: A class for elements that describe the general situation and circumstances of the scenario. Contextualizing can involve any aspect that helps the user or application define the surroundings and general conditions of the scenario.
- Event: A class to represent anything that happens in an instant of time (frame). Therefore, any instantaneous change of state caused by an Object can be defined as an Event. These changes usually cause a new occurrence and, depending on the duration, this can be defined as a new Event or an Action.
- Individual: Instance of an ontology class. In the case of automotive scenarios, a named class should be assigned to each individual of any scene. Hence, individuals of a class are defined by a unique identifier (UID) and a unique name that is specific to each analyzed case.
- Object: A class that represents anything tangible, i.e., a person or thing. They are the main elements of the ontology and can be related to attributes or actions.
- Ontology: Formal description of concepts (Class) and their relations (Axioms) according to a common understanding of experts in the domain. The definition of these elements can be completed with properties or restrictions.
- Scenario: A quantitative and qualitative description of the situation (e.g., traffic environment), as the sequence of Actions and Events performed by Objects.

4. The AGO Domain Model

AGO aims to cover the main elements required to support semantic labeling and the description of automotive scenarios for testing environments. Hence, the core concepts defined in the ontology correspond to those used to structure the information in VCD and OpenLABEL. Its high-level structure can be seen in Figure 1. The main superclasses of the ontology are *Object*, *Context*, *Action*, and *Event*. These elements form the first level of classes and all other classes are derived from them. Furthermore, in VCD and OpenLABEL, the *Relation* element is required to structure the domain knowledge and semantically enrich the ontology. The RDF language model [10] defines several axioms for describing properties and relationships among named terms: "rdfs: domain", "rdfs: range", "rdfs: subClassOf".

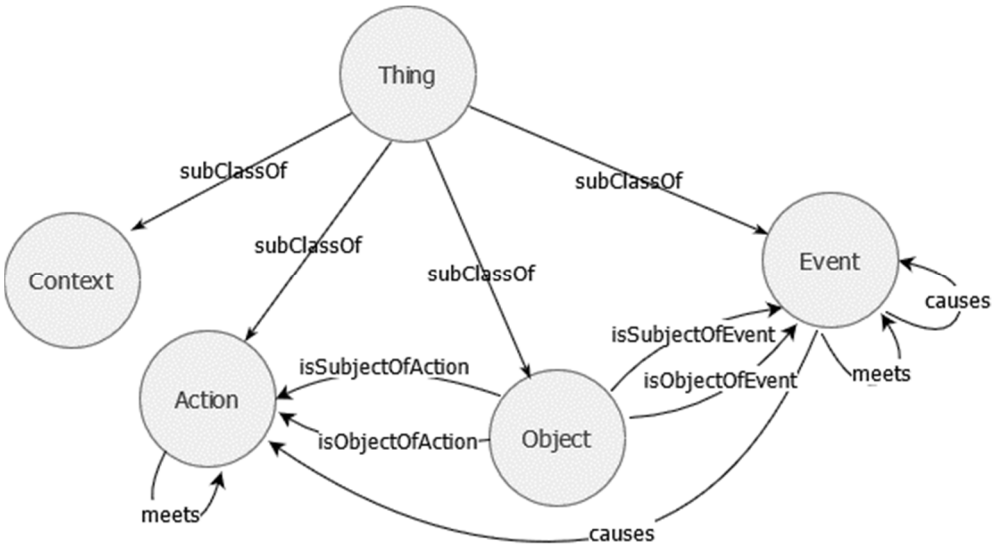

Figure 1. Schema of the domain model of AGO showing core concepts and main relationship types.

Additionally, some non-standard relations are proposed in AGO for the purpose of covering the needs of the description of automotive scenarios.

- "isSubjectOfAction" and "isObjectOfAction": these axioms are defined for actions expressed with *transitive verbs*. Transitive actions can be naturally treated as triples from a language perspective. Nevertheless, transitive triples do not allow relating the action with other nodes of the ontology (because the action in a transitive triple is the predicate or relation between subject and object, and thus it is not a class). Therefore, transitive actions are unwrapped as two related RDF triples, where the action is a class and the relations are "isSubjectOfAction" and "isObjectOfAction". In the case of the *intransitive actions*, a unique triple is generated relating the element that performs the action and the Action itself.
- "isSubjectOfEvent" and "isObjectOfEvent": as for the Actions, the Events can be also distinguished as either transitive or intransitive. Therefore, the same type of relations is defined in AGO for these elements. One describes "*who*" performs the Event and the other "*who/what*" is affected by it.
- "causes": Events are occurrences that happen in a time instant and usually trigger another Event or Action. Therefore, this axiom is adopted to relate Events with Actions and represent this effect.

To provide the ontology with the capability of representing spatio-temporal information by relating the different classes, two additions were performed. First, Allen's temporal relations [21] were adopted (e.g., "meets" as a relation used to define the timeline of the scenario by relating Action and Events in temporal order).

Second, for defining the spatial relations among the elements (e.g., required to construct a complete description of the road network):

- "isPartOf": describes a spatial relation between the different Objects (i.e., "Lane"—"isPartOf" -> "Road")
- "isConnectedTo": this relation is defined to relate all the spatial objects describing the network of objects. Concatenating these static objects (i.e., "Road"—"isConnectedTo" -> "Intersection") is required to formally represent the road network in OpenDRIVE (https://www.asam.net/standards/detail/opendrive/ accessed on 13 May 2021) format.

Furthermore, spatial relations among objects in the scenario can be further specified by "behindOf", "inFrontOf", "leftOf", "rightOf", and "middleOf".

For the representation of the knowledge, in the OWL Reference documentation, the W3C states that an OWL ontology is an RDF graph, which is in turn a set of RDF triples [11] (*subject–predicate -> object*). Therefore, some notations from the data-modeling vocabulary defined by the RDF Schema and the OWL language principles were adopted for the purpose of this work [10]:

- Classes are identified by an Internationalized Resource Identifier (IRI). In addition, each Class is represented by a lexically meaningful Uniform Resource Identifier (URI) that is unique for each entity.
- The RDF/XML document includes an ontology header with the defined base URI.
- The set of all individuals is defined by the class extension of "owl:Thing".

All listed characteristics are automatically considered by Protégé when including new Class entities in the hierarchy. To complete the description of the elements, a *description* and a *label* are included as annotations, which results in a class representation, as depicted in Figure 2.

```
<!-- http://vcd.vicomtech.org/ontology/automotive#Car -->

<owl:Class rdf:about=
"http://vcd.vicomtech.org/ontology/automotive#Car">
    <rdfs:subClassOf rdf:resource=
    "http://vcd.vicomtech.org/ontology/automotive#SmallVehicle"/>
    <rdfs:comment xml:lang="en">A road vehicle with an engine,
    four wheels, and seats for a small number of people.
    </rdfs:comment>
    <rdfs:label xml:lang="en">Car</rdfs:label>
</owl:Class>
```

Figure 2. Example of an AGO Class definition in OWL.

5. Methodology: AGO Construction

This section presents the knowledge acquired during the process of constructing AGO. Figure 3 shows the pipeline of the followed methodology for the construction of the automotive domain ontology.

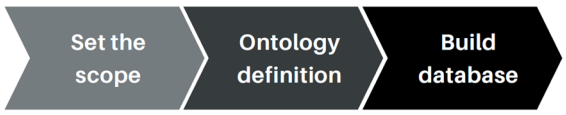

Figure 3. Pipeline of the methodology for ontology building.

5.1. First Phase: Definition of the Scope and Knowledge Acquisition

The first stage is an analysis phase and aims to establish the concrete scope of the use case. This determines which objects are included in the taxonomy of classes. This step requires having a deep knowledge of the domain of interest and can be the most expensive phase in terms of person-hours. In this case, several large-scale open datasets for the development of AD technologies and the Safety Pool taxonomy were analyzed for acquiring knowledge and defining the elements of the ontology. The number of elements is summarized in Table 1. The goal was to define generic classes that cover as many elements of the analyzed resources as possible to enhance interoperability, while maintaining a sufficient level of detail for the applications. Thus, the desired output was a domain-specific hierarchically structured knowledge graph.

Table 1. Summary of the analyzed datasets for defining the taxonomy of the ontology.

Dataset	Year	Object Classes	Action Classes
nuScenes	2019	23	5
H3D Honda	2019	8	-
LyftLevel5	2019	9	18
Waymo	2019	4	-
Apolloscape	2019	65	-
Berkeley Deep-Drive	2020	40	-
Audi A2-D2	2020	52	-
Mapillary-Vistas	2020	66	-
SafetyPool (Taxonomy)	2020	126	-

In general, these datasets do not define classes in a hierarchy but often constitute a one-level list. In addition, only two of the listed datasets cover a few classes related to maneuvers (Action classes in the AGO domain). However, actions are presented as special attributes of the corresponding objects. Due to the manner in which they are defined, these datasets do not consider the need of making a distinction between Action and Events. Moreover, they do not present semantically relevant relations among these elements and the objects. Taking into account that most of these datasets do not consider the traffic maneuvers, the "*ALKS Regulation UN R157*" (https://undocs.org/ECE/TRANS/WP.29/2020/81 accessed on 20 July 2021) was included in AGO so that the ontology gathers the critical scenarios listed in the regulation as Actions, along with some additional Object classes and properties. As a result, AGO is an ontology that offers flexibility for present and future labeling needs, supported by the established mechanism to connect Objects, Actions, and Events through Relations. AGO also includes, in its current form, a rich set of object classes, which are easily extensible, and an equivalently wide number of actions and maneuvers that can serve as a basis for multiple applications in the domain (e.g., test coverage analysis, online data recording, and digital twins).

5.2. Second Phase: Build the Taxonomy of Classes

5.2.1. Technologies and Tool

The Semantic Web focuses on enabling machines with the capabilities of providing formal structured information using the encoded knowledge from ontologies [22]. There are several languages that enable formalizing and encoding knowledge. For the purpose of this work, the chosen ontology representation language was the Web Ontology Language (OWL) [11]. OWL was developed as a vocabulary extension of the Resource Description Framework (RDF) specification. This linking structure forms a directed, labeled graph, in which the edges represent the named link between two resources, represented by the graph nodes [5].

There are several tools specifically designed to create and manage ontologies: Protégé, Protégé Web, Fluent Editor (FE), OWLGrEd, etc. For this approach, the ontology was defined with Protégé (https://protege.stanford.edu/ accessed on 11 March 2021) because this tool enables intuitive construction of complete ontologies. Similarly, the ontology can be exported into different ontology representation languages, including RDF and OWL. This tool also provides the user with a Taxonomy Tree viewer (see Figure 4) and other visualization tools, such as OWLViz, which generates a diagram of the selected elements in different layouts.

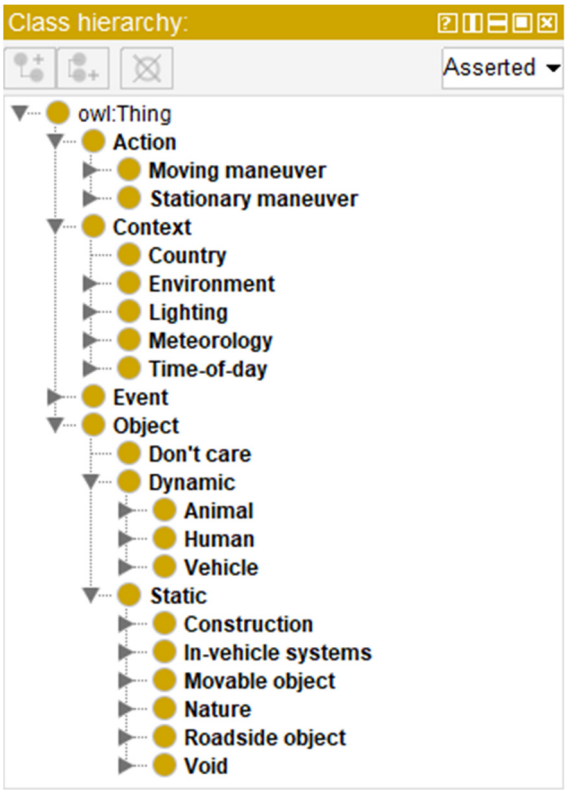

Figure 4. Taxonomy tree in Protégé showing the main hierarchy of classes in the ontology.

This tool also provides the means to include Annotation properties, which comprise ontology metadata and general properties of the classes. For the metadata, the *versionInfo* (owl:versionInfo) is defined as a built-in annotation property of OWL [11]}. Furthermore, some of the *Dublin Core Metadata Initiative* (DCMI) [23] terms are adopted: *Contributor*, *creator*, *description*, *title*, *date submitted*, *license*, and *publisher*. In addition, to complete the description of the classes, a brief *description* and a *label* are included as defined by the RDF schema [10]: rdfs:comment and rdfs:label.

5.2.2. Ontology Construction

The AGO domain ontology was constructed with the common classes identified among the datasets in the first phase, and the analysis was undertaken with the objective of identifying the main terms and the related concepts. Thus, this phase involved the following tasks:

- Identification of the common classes among the datasets;
- Define upper-level classes for logically structuring the hierarchy;
- Include every class defined in the datasets (manage synonyms).

The defined ontology provides a shared understanding of common concepts (Classes) among the main automotive open datasets. Therefore, it can be considered to be a domain ontology. The general classification of the classes was conducted according to the element types defined in the AGO domain model description (Section 4). Hence, the upper level consists of four core concepts, as depicted in Figure 1. In addition, it was constructed in a top-manner down using the *Containment Relationship Type* [24] ("rdfs:subClassOf") and according to the hierarchy defined on the datasets. Consequently, most of the included classes appear in one or more of the analyzed datasets. After this process, the hierarchy of classes evolves from the simpler tree or plain lists defined in the source datasets into a more complex graph structure in which each class may have several parent nodes.

5.2.3. Knowledge Representation Language and Knowledge Graph Structure

The proposed structure of AGO is a directed graph because this configuration provides the means of representing high-order semantic relations as RDF triples with basic elements of this non-SQL datasets. In terms of the triples, the *subject* and *object* are represented by "Class" labelled nodes. Data properties (owl:DataProperty) are also included as nodes under the "dataProperty" label. These serve as attributes to the classes according to the OWL specification. To semantically cover a traffic scenario, several complex relationships are required to link different elements. The user-defined axioms presented in Section 4 are included as object properties (owl:ObjectProperty) which, according to the OWL (RDF/XML) Structural Specification [20], is the construct defined to connect pairs of individuals with user-defined relationships. Therefore, the graph is populated with them as nodes with the "objectProperty" label to represent the *predicate* of the triples. Both property type nodes are related using the "DOMAIN" and "RANGE" edges. This explanation is illustrated in the graph snippet of Figure 5.

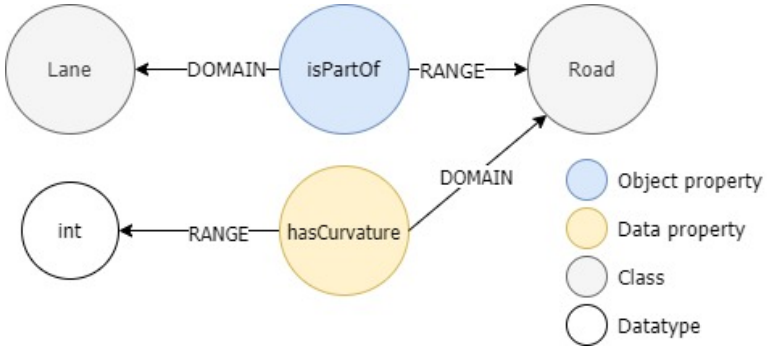

Figure 5. Object property and data property triple examples represented as a graph.

5.3. Third Phase: Database Building in Neo4j

The third stage implies building the knowledge-graph database and making the ontology available for its use via programmatic interfaces. For this purpose, a database to store the data was selected. In this case, Neo4j (https://neo4j.com/neo4j-graph-database/ accessed on 24 July 2021) *neo4j-community-4.2.0*) was chosen as the database for storing the ontology. To represent the ontology as a graph in Neo4j following the standards, the neosemantics plugin (n10s) (https://github.com/neo4j-labs/neosemantics accessed on 4 August 2021) is required. This enables importing and exporting the ontology graph from and into OWL files for further use. Using *n10s Release 4.1.0*, a Cypher query can be used to import the OWL ontology from a local directory or a URL. In addition, the query can be passed with some specific parameters that determine the names given to the elements in the database. The specific query used to structure the ontology is presented in Figure 6.

```
CALL n10s.onto.import.fetch({filepath}, 'RDF/XML',
{classLabel: 'AGO', subClassOfRel: 'subClassOf',
dataTypePropertyLabel: 'dataProperty',
objectPropertyLabel: 'objectProperty',
subPropertyOfRel: 'subPropertyOf'
```

Figure 6. Main cypher query used to import the ontology.

The imported data are further processed by including specific Neo4j labels to classify the nodes according to the core concepts. Grouping nodes with tags related to the first level of classes in the hierarchy (i.e., Action, Object, etc.) helps optimization of the queries to the ontology because the consultation is undertaken for a smaller subset of nodes. Consequently, the ontology-based applications developed in later works should be optimized in terms of ontology consultation.

The resulting graphs' main characteristics are summarized in Table 2. AGO has a total of 523 nodes and 1365 relationships. Some of the class nodes have related properties to provide the user with complete semantic information about each element. These are known as property keys in Neo4j.

Table 2. Main characteristics of the ontology.

Ontology Name	AGO
Version	1.1.0
Neo4j-labels	Context, Object, Action, Event, dataProperty, objectProperty, metadata
Number of Classes (Nodes)	390
Number of Objects	296
Number of Actions	42
Number of Events	24
Number of Contexts	33
Number of dataProperties	57
Number of objectProperties	64
Number of Relations	1367
Relationship types	subClassOf, subPropertyOf, DOMAIN, RANGE
Number of subClassOf Relations	398
Number of subPropertyOf Relations	119
Number of DOMAIN Relations	364
Number of RANGE Relations	486
Property Keys	label, name, comment, URI, type

When including the elements in Neo4j, the features of each element are added as node properties. Table 3 summarizes the included property keys with the related OWL syntax equivalence.

Table 3. Property key summary and OWL syntax equivalence.

Property Key	Description	OWL Syntax
Uid	A unique identifier is included by default to each element.	-
Label	Human-friendly label. Defined for all elements.	rdfs: label
name	The name of each class is given in CamelCase. Defined for all the elements	-
comment	A brief description of the defined element nature. The user should review the comment to understand how the elements are understood in the automotive domain.	rdfs: comment
URI	URIs are used in OWL as object identifiers. Their structure is formalized by the ontology language to be a valid URL where the complete ontology is available and there is a unique name for each class. In the case of AGO, the URI looks like: http://vcd.vicomtech.org/ontology/automotive/# + {name}	rdf:about
Type	Represents the type of data that each dataProperty element should have, which is related to the properties of the classes. For example, the "*color*" attribute should be a string whereas "*height*" should be an integer.	Datatypes

The taxonomy tree for the main classes in Protégé and OpenLABEL differs because the relations are not defined as a Class type in the graph representation. The user-defined axioms are defined as object property nodes and connected with the required elements with the "DOMAIN" and "RANGE" relationships to build the RDF triples. Hence, the subject of the triple is represented by the domain and the object by the range terms. Figure 5 represents the "Lane"–"isPartOf" -> "Road" object property triple as a graph. In addition, Figure 5 also depicts how the "curvature" attribute is related to the "Road" class following the data property syntax definition.

6. Use Cases

This section summarizes two different but related use cases in the automotive domain. The first proposes the utilization of the ontology to guide the creation of configuration files for semantic labeling applications. The second relates to the creation of a database of graphical scenario representations from real data labels and expert knowledge.

6.1. Semantic Labeling

Labeling is the process of creating descriptions of the content of some data. For images or other sensorial data, labels are typically spatio-temporal entities that determine the presence of objects or actions in the reality captured by the sensors [1]. With the emergence of DL, labeling has become a major activity for automakers and providers of electronics. With DL, a sufficiently large and rich dataset containing sensorial data and labels can be used to train models that learn from the dataset and predict labels on previously unseen data. This ability has triggered the creation of many ADAS and AD functions.

As a consequence, datasets have become a critical asset for players in the automotive market. Labels are frequently defined using function-specific or customized taxonomies and lacking a global or universal hierarchy. Datasets are usually non-compatible and difficult to merge due to semantic inconsistencies between the used terms.

In this sense, AGO, as a domain ontology, can serve as the core of a data translation function that maps relations between (otherwise non-interoperable) datasets. During the development of the work, all the datasets were represented graphically, and it was verified that the labels of the elements can be mapped to their respective synonyms in the ontology. This translation function requires the definition of advanced search queries. The result may be used to feed training models that need to be adapted for new tests that use different datasets from those used for training. A diagram of the process is presented in Figure 7.

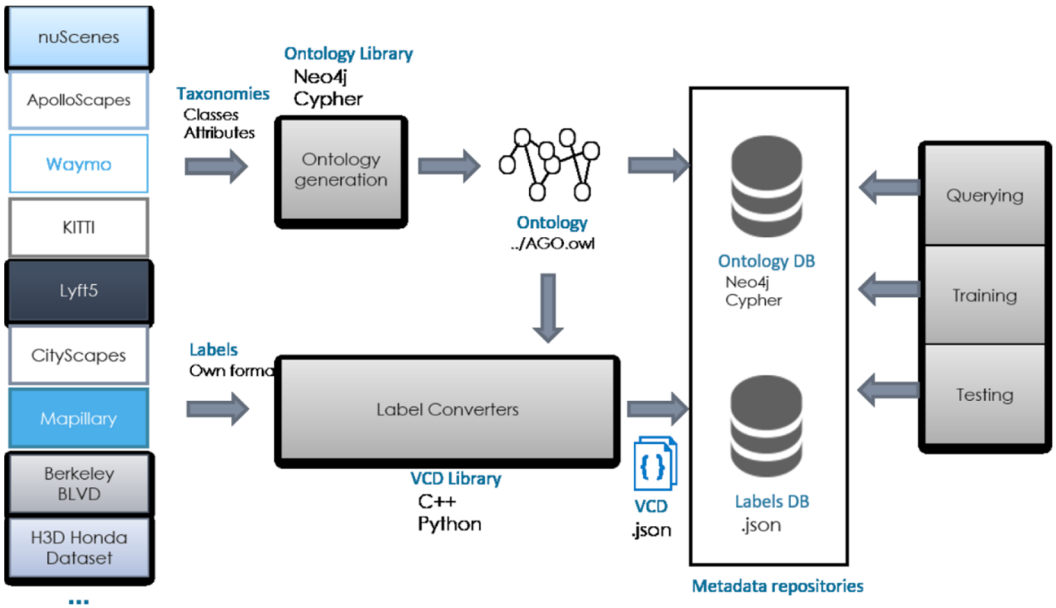

Figure 7. Using AGO ontology for translating of heterogeneous labels (e.g., from different datasets).

As an example of the above-mentioned process, Figure 8 depicts a graphical representation of the classes defined in the "Waymo" dataset mapped to their equivalent AGO classes. The equivalence among elements is defined by the "owl:sameAs" relationship, which means that one can be replaced by the other without altering the meaning and vice versa.

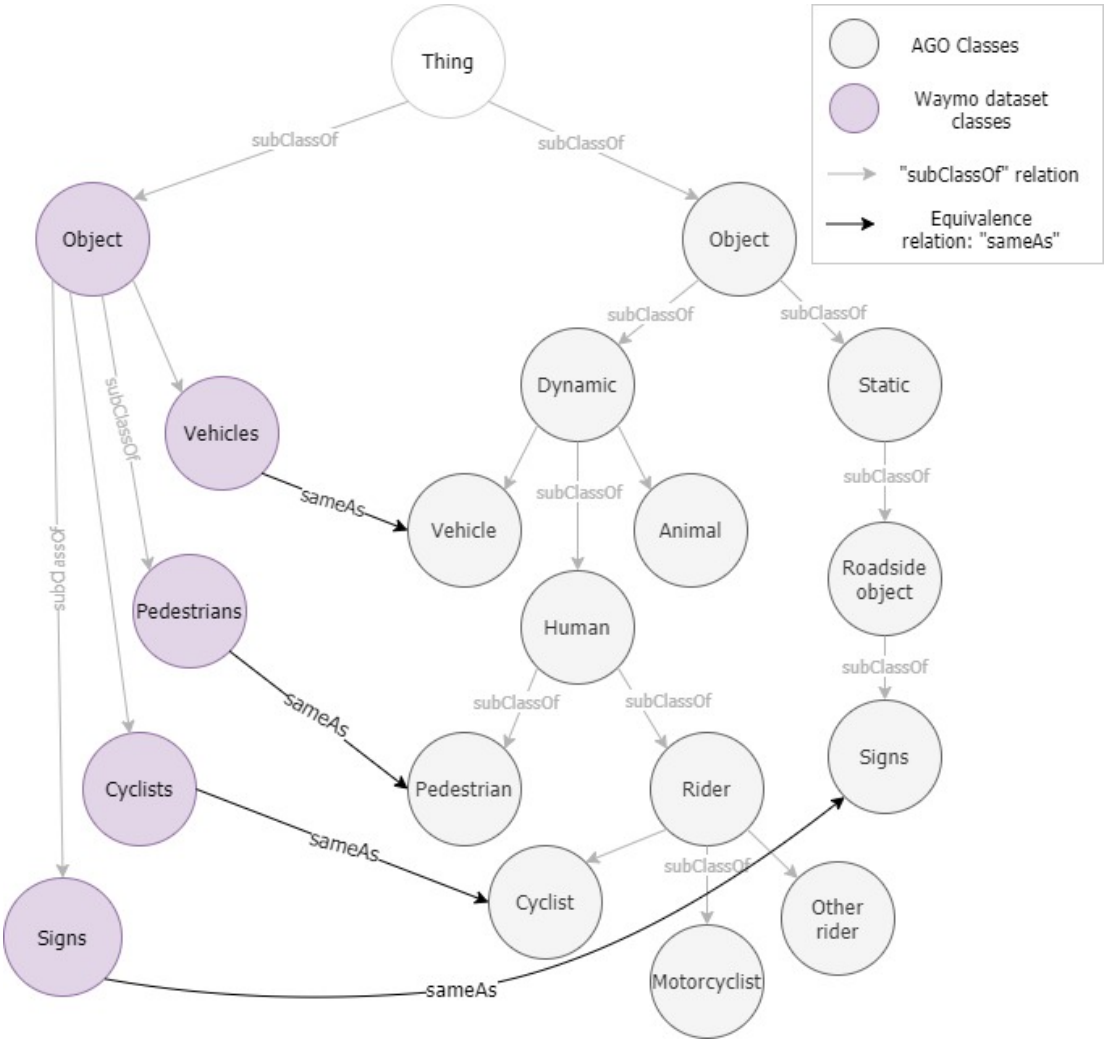

Figure 8. Waymo dataset mapping to AGO example. Matched classes represent the common terms among the different datasets and, thus, the translation can be undertaken by identifying these equivalences.

This type of application allows identifying the relations among different datasets and translating the terms to the needed terminology. The translation can be performed automatically or manually, with user interfaces showing a graph visualization and navigation capabilities.

Furthermore, the ontology can be part of a knowledge management system and used before the labels are created, at the annotation or labeling stage (see Figure 9). Specifically, the ontology can serve as a database with a formal definition of a unified understanding of the terminology related with the use case. In this manner, the database can be used for generating structure or configuration files that can guide a labeling application or process to produce labels not only in the expected format (e.g., OpenLABEL), but also semantically compatible with the ontology as depicted in Figure 9. Also related with these files, both the structure and the terms defined using natural language can be validated with a content-checker application. Thus, even if there are different users working with the tool or defining several use cases, the terms are chosen from the ontology and remain

the same for every use case. For instance, in the case of web-based annotation tools [25], its functionality could be boosted by including relations among the annotated objects. In addition, the previous annotations and configuration files may be checked according to the vocabulary defined in the ontology, enhancing interoperability by ensuring a common understanding of the domain concepts.

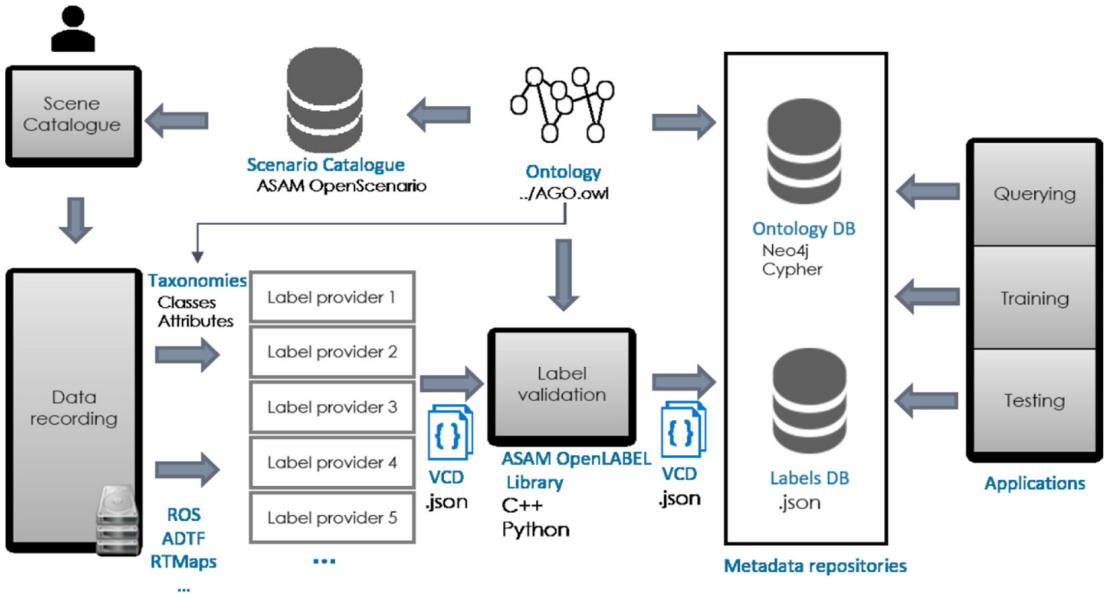

Figure 9. Using AGO to specify taxonomies and configuration files before the labeling stage to produce semantically compliant content.

6.2. Graphical Scenario Representation

Scenario representation is also a key aspect of ADAS/AD development and testing. Rich and realistic scenario representations can lead to the generation of simulated environments that can be utilized in virtual testing procedures (e.g., Hardware-in-the-Loop simulations). Scenarios may include description of the participants of a road or driving scene, including their interactions, spatio-temporal relations, etc.

Scenarios can be generated from expert knowledge, i.e., from high-level descriptions of how the situation should be, or from real data, i.e., from semantic labels obtained from annotation processes.

In this use case, both approaches were implemented, creating synthetic scenarios from the ALKS regulation, and real scenarios from the KITTI dataset (http://www.cvlibs.net/datasets/kitti/eval_tracking.php accessed on 4 June 2021). These scenarios were created by first using the VCD toolkit to create the RDF entries in the OpenLABEL format, which is flexible enough to host high-level actions and relations (for the ALKS scenario), and to abstract high-level information from detailed labels (for the KITTI annotations).

Second, these VCD payloads were imported into Neo4j to build a scenario database. In the graph, each scenario is represented by a primary node with the following metadata:

- cnl_text: textual description of the scenario in a Controlled Natural Language (CNL);
- date_db: date and time of the latest update of the node;
- scenario_uid: is composed by the information source and a numerical id;
- schema_version: the VCD version used to represent the imported information.

The working example presented in this section is represented by the schematic diagram in Figure 10. Additionally, the scenario is graphically depicted in Figure 11. The depicted example includes some individuals and these nodes have the following information included as node properties:

- frame_intervals: the start and end frames for each node;

- name: the name of the individual in CamelCase given as the class name plus a number used to list the individuals with the same type;
- type: the name of the corresponding ontology class.

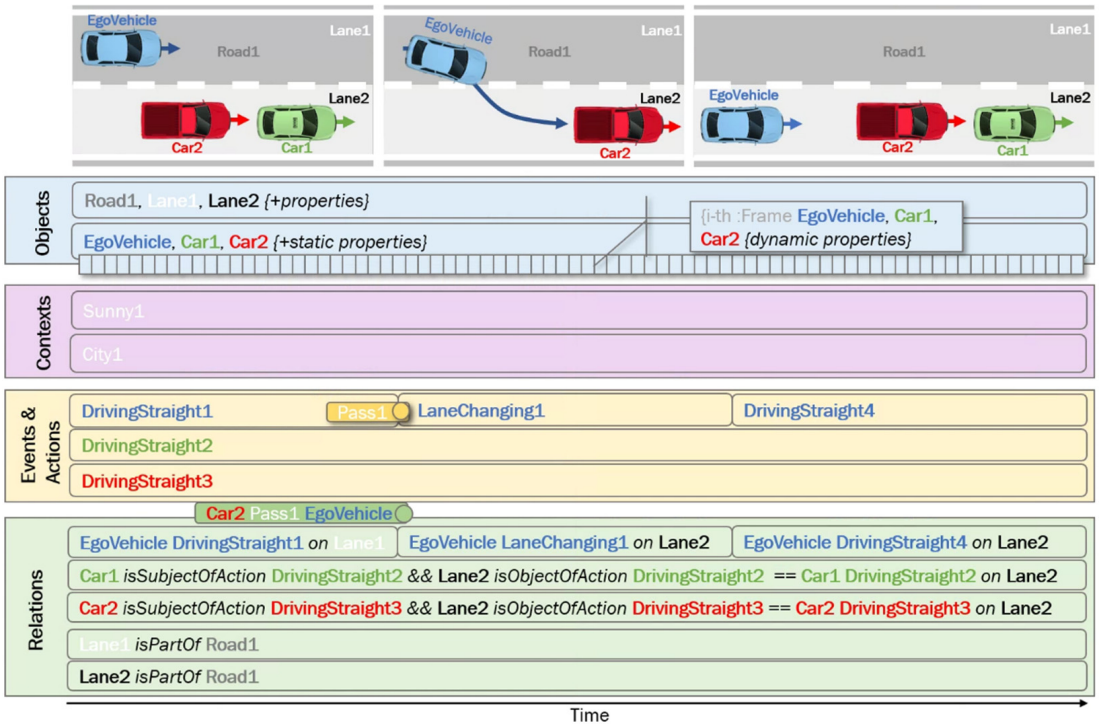

Figure 10. Diagram of the KITTI example scenario elements and relations illustrated over time.

Continuing with the scenario representation illustrated in Figure 11, the upper nodes represent the core static elements that correspond to the upper items of the VCD JSON schema. Hence, the metadata information is included as the node properties of the center node of the representation.

The individuals are presented with the semantically meaningful relations among them. These relations can be easily translated into the form of RDF triples, taking into account the pre-defined "subClassOf" containment relationship.

In addition, considering the AGO domain axioms, the list of the extracted triples for the example in Figure 11 can be extended. Taking all these triples into account, a translation into a NL textual description can be easily undertaken. Both the list of triples and the NL textual description are presented in Table 4. At this stage of the pipeline, the resulting data correspond to a *functional scenario* [9], which can be extended to obtain a *logical scenario* by considering the data properties defined in the ontology and included with the "hasAttribute" relation to the scenario representation.

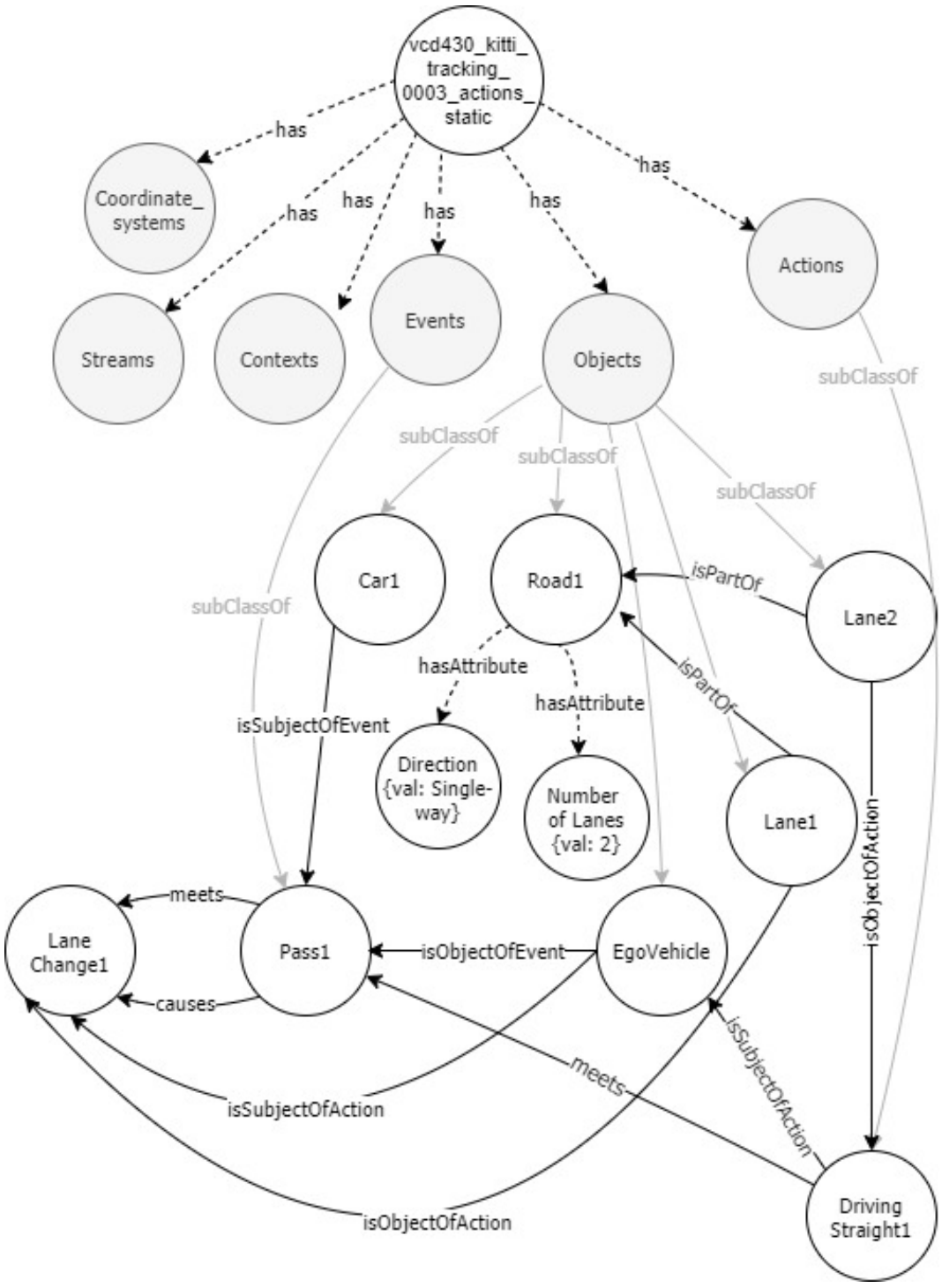

Figure 11. Part of the representation as a graph of the KITTI dataset example scenarios' static elements.

Table 4. RDF triples converted into NL textual description for the KITTI dataset example scenario.

Scenario UID	vcd430_kitti_tracking_0000_actions_static
RDF Triples ("subClasssOf")	"EgoVehicle" -subClassOf-> "Object" "Car2" -subClassOf-> "Object" "Road1" -subClassOf-> "Object" "Lane1" -subClassOf-> "Object" "Lane2" -subClassOf-> "Object" "Pass1" -subClassOf-> "Event" "DrivingStraight1" -subClassOf-> "Action" "Lane Changing1" -subClassOf-> "Action"
RDF Triples (User-defined axioms)	"Lane1" -isPartOf-> "Road1" "Lane2" -isPartOf-> "Road1" "EgoVehicle" -isSubjectOfAction-> "DrivingStraight1" "Lane2" -isObjectOfAction-> "DrivingStraight1" "Car2" -isSubjectOfEvent-> "Pass1" "EgoVehicle" -isObjectOfEvent-> "Pass1" "EgoVehicle" -isSubjectOfAction-> "LaneChanging1" "Lane1" -isObjectOfEvent-> "LaneChanging1" "Pass1 -causes-> "LaneChanging1" "DrivingStraight1" -meets-> "Pass1" -meets-> "LaneChanging1"
NL Description	The ego-vehicle is driving straight in lane1, which is part of a single-way two-lane road. When another car passes the ego-vehicle, then it starts lane changing into the other lane of the same road, lane2.

7. Results

The example ontology produced following the proposed methodology is available online (https://vcd.vicomtech.org/ontology/automotive accessed on 23 August 2021) in RDF format. The automotive ontology is composed of 390 class elements classified into three main groups using Neo4j labels: "Object", "Context", "Action" and "Event". The definition of each element is completed with annotation properties. In addition, the file contains 1367 relationships, of which 398 represent hierarchical relations among the classes (and so defining a graph hierarchy). AGO can be used as a top-level ontology and reused as a starting point to build new domain or application ontologies. Furthermore, SWRL (Semantic Web Rule Language (https://www.w3.org/Submission/SWRL/ accessed on 10 May 2021)) rules are included in the ontology file to extend the axioms of the scenarios by inferring knowledge. They can be also used to validate that the inclusion of the individuals is correct. Nevertheless, these rules are not imported into Neo4j and, therefore, their usability is not further extended in this paper.

The scripts used to build the ontology and scenario databases in Neo4j and other additional material, such as ALKS and KITTI functional scenario files, can be found at the GitHub repository (https://github.com/Vicomtech/video-content-description-VCD/tree/master/ontologies accessed on 23 August 2021).

Comparison with Existing Ontologies

Different available ontologies related to the automotive domain were analyzed (summary presented in Table 5) to identify the gaps that need to be addressed for semantic labelling and scenario representation. Starting with the *Transport Disruption Ontology* [7] mentioned in Section 2, an exhaustive list of hierarchically classified events that may cause disturbances in traffic scenarios is presented. "Agent" is defined to be the subject of the events; however, the only objects defined as its subclasses are "Person", "Group", and "Organization". The scope of the Transport Disruption Ontology differs from the interests of AGO; consequently, this ontology does not cover the whole range of actions and objects related to traffic scenarios. Nevertheless, it may be used to extend the classes by reusing

existing developments. Further, the listed "event" classes have related temporal objects based on the OWL-Time ontology, which allows the distinctions to be made among occurrences that happen instantaneously or during a time range. This means that, although they have somehow identified the need to make a distinction between Action and Events, this is implemented implicitly rather than explicitly. Hence, some object properties defined in relation to the "time" classes work as in the same manner as Allen's temporal relations.

Table 5. Comparison of AGO with existing ontologies.

Name	Description	Access	Spatio-Temporal Relations	Action Support	Event Support
Automotive Global Ontology (AGO)	Automotive domain ontology for traffic scenario representation and semantic labeling applications.	Open	Yes	Yes	Yes
The Transport Disruption Ontology	A formal framework for modeling travel- and transport-related events that have a disruptive impact on an agent's planned travel.	Open	No	No	No
Ontology for scenarios for the assessment of AVs	An ontology for describing scenarios in the context of the assessment of the performance of an AV.	Private	Yes	Yes	Yes
TTI Core	A machine understandable knowledge base for autonomous driving vehicles constructed using three core ontologies: map, control, and car ontology.	Under License	No	No	No

Most of the published approaches cover completely different scopes within the automotive domain; therefore, they lack many classes and properties to fulfill the requirements of scenario representation and semantic labelling. One example is *The Automotive Ontology (AUTO)* [26] created by the W3C Automotive Ontology Community Group, which only covers classes related to popular cars, buses, and motorcycles. Continuing with *TTI Core* [27], this is a layered approach presented as three core ontologies for safe autonomous driving: car, control, and map ontologies. Each of these covers a minimum portion of the domain-related classes and, despite not covering space–time relationships, it presents elements in order to relate objects to map elements

In contrast, the ontology for scenarios for the assessment of autonomous vehicles [28] is the method that most resembles AGO because it makes a clear distinction between Action and Event classes. However, the relationships defined to relate classes are not clear because the ontology is not publicly accessible and the information about it is scarce.

8. Discussion

Most existing ontology building approaches in the automotive domain do not use graph-based tools as part of their pipeline. The presented methodology is based on Neo4j, a graph database that provides flexibility to easily modify and update the ontology with new information. This is a key feature when developing ontology-based approaches, because new scenarios will generate new individuals and, in turn, these elements should be updated for each case. In addition, graph-based data representation models are especially effective for the expression of highly related data, such as hierarchical classification or mappings between concepts. A graph database is also an interesting tool to provide interoperability to the ontology, because it is compatible with numerous services and languages. Therefore, it is easy to use with different applications.

In this work, the area of interest was the automotive applications of semantic labeling and scenario-based testing. Hence, a formal description of the ontology that covers as many driving scenes as possible was presented. The pipeline is based on the taxonomical structure of classes presented in the main automotive datasets, with expressivity and

semantic load added via the inclusion of new relations. As result a reusable top-level automotive domain ontology, named AGO, was defined.

The methodology is defined with accessible tools and steps that do not require a significant technical background. The objective was to provide the means to construct and take advantage of the ontologies of as many user profiles as possible.

The knowledge regarding the construction of a domain ontology presented in this paper, and the two selected use cases, has been made available to the ASAM standardization group (https://www.asam.net/ accessed on 5 June 2021) for the development of the OpenLABEL and OpenXOntology standardization projects (https://www.asam.net/project-detail/asam-openxontology/ accessed on 5 June 2021) (to appear 2021–2022), to contribute to the automotive industry and the scientific community.

Author Contributions: Investigation, M.N., I.U.; writing—original draft preparation, M.N., I.U.; writing—review and editing, M.N., I.U.; data preparation M.N., M.G., I.U.; supervision M.N., O.O.; project administration M.N., O.O.; funding acquisition M.N., O.O. All authors have read and agreed to the published version of the manuscript.

Funding: This work has received funding from the European Union's H2020 research and innovation program (grant agreement no 824309, project HEADSTART).

Data Availability Statement: The final version of the defined ontology is available online and can be accessed through the URL (https://vcd.vicomtech.org/ontology/automotive) or from the GitHub repository (https://github.com/Vicomtech/video-content-description-VCD/tree/master/ontologies). The files necessary for the construction of the ontology and the scenario-database have also been included.

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

References

1. Nieto, M.; Senderos, O.; Otaegui, O. Boosting AI applications: Labeling format for complex datasets. *SoftwareX* **2021**, *13*, 100653. [CrossRef]
2. Suárez-Figueroa, M.C.; Gómez-Pérez, A.; Fernández-López, M. The NeOn Methodology framework: A scenario-based methodology for ontology development. *Appl. Ontol.* **2015**, *10*, 107–145. [CrossRef]
3. 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]
4. Benvenuti, F.; Diamantini, C.; Potena, D.; Storti, E. An ontology-based framework to support performance monitoring in public transport systems. *Transp. Res. Part C Emerg. Technol.* **2017**, *81*, 188–208. [CrossRef]
5. RDF Working Group. Resource Description Framework (RDF). 2014. Available online: https://www.w3.org/RDF/ (accessed on 23 July 2021).
6. Lorenz, B.; Ohlbach, H.; Yang, L. Ontology of Traffic Networks (OTN). REWERSE Proj. Publ. 2005. Available online: http://rewerse.net/publications/rewerse-description/REWERSE-DEL-2005-A1-D4.html (accessed on 25 July 2021).
7. Corsar, D.; Markovic, M.; Edwards, P.; Nelson, J.D. The transport disruption ontology. In *The Semantic Web—ISWC 2015. Lecture Notes in Computer Science*; Arenas, M., Ed.; Springer: Cham, Switzerland, 2015; Volume 9367, pp. 329–336. [CrossRef]
8. Menzel, T.; Bagschik, G.; Isensee, L.; Schomburg, A.; Maurer, M. From functional to logical scenarios: Detailing a keyword-based scenario description for execution in a simulation environment. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 2383–2390. [CrossRef]
9. Menzel, T.; Bagschik, G.; Maurer, A.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]
10. Brickley, D.; Guha, R.V. RDF Schema 1.1. 2014. Available online: https://www.w3.org/TR/rdf-schema/ (accessed on 10 March 2021).
11. Bechhofer, S.; van Harmelen, F.; Hendler, J.; Horrocks, I.; McGuinness, D.L.; Patel-Schneider, P.F.; Stein, L.A. OWL Web Ontology Language Reference. W3C Recommendation. 2004. Available online: https://www.w3.org/TR/owl2-overview/ (accessed on 5 May 2021).
12. Fernandez, M.; Gomez-Perez, A.; Juristo, N. METHONTOLOGY: From ontological art towards ontological engineering. In Proceedings of the AAAI97 Spring Symposium, Stanford, CA, USA, 24–26 March 1997; pp. 33–40.
13. Sure, Y.; Staab, S.; Studer, R. Handbook on ontologies. In *On-To-Knowledge Methodology (OTKM)*; Springer: Berlin/Heidelberg, Germany, 2013.

14. Pinto, H.S.; Staab, S.; Tempich, C. DILIGENT: Towards a fine-grained methodology for Distributed, Loosely-controlled and evolving Engineering of oNTologies. In Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), Valencia, Spain, 22–27 August 2004; pp. 393–397.
15. Katsumi, M.; Fox, M. Ontologies for transportation research: A survey. *Transp. Res. Part C Emerg. Technol.* **2018**, *89*, 53–82. [CrossRef]
16. Bermejo, A.J.; Villadangos, J.; Astrain, J.J.; Cordoba, A. Ontology based road traffic management. In *Intelligent Distributed Computing VI Studies in Computational Intelligence*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 446, pp. 103–108. [CrossRef]
17. De Nicola, A.; Missikoff, M.; Navigli, R. A Proposal for a Unified Process for Ontology Building: UPON. *Lect. Notes Comput. Sci.* **2005**, *3588*, 655–664. [CrossRef]
18. Bellini, P.; Bruno, I.; Nesi, P.; Rauch, N. Graph databases methodology and tool supporting index/store versioning. *J. Vis. Lang. Comput.* **2015**, *31*, 222–229. [CrossRef]
19. Ulbrich, S.; Nothdurft, T.; Maurer, M.; Hecker, P. Graph-based context representation, environment modeling and information aggregation for automated driving. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014. [CrossRef]
20. Bock, C.; Fokoue, A.; Haase, P.; Hoekstra, R.; Horrocks, I.; Ruttenberg, A.; Scattler, U.; Smith, M. OWL 2 Web Ontology Language. Structural Specification and Functional-Style Syntax (Second Edition). W3C. 2012. Available online: https://www.w3.org/TR/owl2-syntax/ (accessed on 10 March 2021).
21. Batsakis, S.; Petrakis, E.G.; Tachmazidis, I.; Antoniou, G. Temporal representation and reasoning in OWL 2. *Semant. Web* **2017**, *8*, 981–1000. [CrossRef]
22. Karbe, T. State of the Art for Automotive Ontology. CRYSTAL Project. Deliverable Patent Number D308.010, 5 June 2014.
23. DCMI Usage Board. DCMI Metadata Terms. 2020. Available online: http://dublincore.org/specifications/dublin-core/dcmi-terms/2020-01-20/ (accessed on 28 June 2021).
24. Wang, S.; Tanaka, K.; Zhou, S.; Ling, T.W.; Guan, J.; Yang, D.; Grandi, F.; Mangina, E.; Song, I.-Y.; Mayr, H.C. Conceptual modeling for advanced application domains: ER 2004 Workshops CoMoGIS, CoMWIM, ECDM, CoMoA, DGOV, and ECOMO. In *Lecture Notes in Computer Science*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2005; pp. 402–414. Available online: https://books.google.es/books?id=NkgBCAAAQBAJ (accessed on 15 May 2021).
25. Mujika, A.; Fanlo, A.D.; Tamayo, I.; Senderos, O.; Barandiaran, J.; Aranjuelo, N.; Nieto, M.; Otaegui, O. Web-based Video-Assisted Point Cloud Annotation for ADAS validation. In Proceedings of the 24th International Conference on 3D Web Technology, Los Angeles, CA, USA, 26–28 July 2019. [CrossRef]
26. Hepp, M.; Kuzinski, D.; Trypuz, R.; Szczepański, K. Markup for Autos. 2021. Available online: https://schema.org/docs/automotive.html (accessed on 7 May 2021).
27. Zhao, L.; Ichise, R.; Mita, S.; Sasiki, Y. Core ontologies for safe autonomous driving. In Proceedings of the International Semantic Web Conference (Posters & Demos), Bethlehem, PA, USA, 11–15 October 2015.
28. De Gelder, E.; Paardekooper, J.P.; Saberi, A.K.; Elrofai, H.; Ploeg, J.; Friedmann, L.; De Schutter, B. Ontology for scenarios for the assessment of automated vehicles. *arXiv* **2020**, arXiv:2001.11507.