applied sciences

MDPI

Article

Ontology Support for Vehicle Routing Problem

Anita Agárdi [1], **László Kovács** [1] **and Tamás Bányai** [2,*]

[1] Institute of Informatics, University of Miskolc, 3515 Miskolc, Hungary
[2] Institute of Logistics, University of Miskolc, 3515 Miskolc, Hungary
* Correspondence: alttamas@uni-miskolc.hu

Abstract: This paper aims to present a generalized ontology model for the Vehicle Routing Problem (VRP) and it gives some out-plant material handling case studies. The Vehicle Routing Problem is a logistics task where customers with a specific need for products are served within the least possible distance traveled by vehicles. The Vehicle Routing Problem has been highly investigated in operations research, computer science, transportation science, and mathematics. As our new approach shows, the VRP can be used to model in-plant and out-plant material handling and out-plant passenger transport. The Vehicle Routing Problem is a complex, multi-component heterogeneous environment, where consistent handling and integrity of components is a more difficult problem. In this alignment (integrity management, automation), our goal was to develop a unified semantic background framework. Our ontology describes the concepts and the relationships between concepts for the investigated domain. The paper presents the construction and application of ontology for a sample framework and presents test runs based on case studies. The paper shows that ontology can be built into the logic of software applications related to logistic problems. The last part of the article focuses on case studies for our ontology model from the field of tank, money, parcel, and perishable food transportation.

Keywords: vehicle routing problem; ontology; logistics

check for updates

Citation: Agárdi, A.; Kovács, L.; Bányai, T. Ontology Support for Vehicle Routing Problem. *Appl. Sci.* **2022**, *12*, 12299. https://doi.org/10.3390/app122312299

Academic Editors: Jarosław Wikarek and Mirco Peron

Received: 22 October 2022
Accepted: 29 November 2022
Published: 1 December 2022

1. Introduction

Transportation tasks have become increasingly complex over the years, so there may be a need for a general system that can model many variants of transportation tasks. Our goal is to present such a system based on the components of the well-known Vehicle Routing Problem (VRP) logistic model. The basic Vehicle Routing Problem consists of the following elements: customers, a single depot, vehicles, and the demands of the customers. The vehicles transport products from the depot to the customers and then return to the depot. The objective function is the minimization of the length of the route taken by the vehicles. In the basic VRP, the following constraints can be formulated: the incoming and outcoming edges of each node must be the same; each customer must be visited once; vehicles have a capacity limit (which is the same in the case of all vehicles); the number of visits to the depot is equal to the number of vehicles.

After the first publication on Vehicle Routing in 1959 by Dantzig and Ramser [1], several types of vehicle routing have been discussed. In the following, the VRPs that contributed to our generalized model are presented. In the case of Capacity Constraint (Capacitated VRP) [2], vehicles have a capacity limit for the products to be transported. Cross-Docking (VRP with Cross-Docking) [3] is a technique in which the products are first collected (picked up) from each node and then delivered from there. For the technique, it is not customary to store the collected products for a long time; the delivery takes place almost immediately. In the case of the Cumulative Capacitated VRP [4], the objective is to minimize the waiting time in the nodes. Electric Vehicles (Electric VRP) [5] means that in the system there are electric vehicles that transport products. In the case of Environmentally Friendly Vehicles (Environmentally Friendly VRP) [6], environmentally friendly vehicles transport the products. In the case of Fuel-Efficient Green Vehicle Routing Problem [7], the objective is not the minimization of the distance traveled but the minimization of the fuel consumption of the vehicles. A Fuzzy Vehicle Routing Problem [8] is a type, where certain factors, e.g., the product demands of the customers, time windows, and route distance, are given by fuzzy numbers. A Heterogeneous Fleet (Heterogeneous Fleet VRP) [9] is one where the types of vehicles in the system are different. In the case of a Homogeneous Fleet (Homogeneous Fleet VRP) [10], the vehicle types in the system are the same. Inter-Depot Routes (Multi-Depot VRP with Inter-Depot Routes) [11] means that there are multiple depots in the system. Each vehicle leaves one of the depots, but after visiting customers, it can return to another depot. In the case of Multiple Depots (VRP with Multiple Depots) [12], there are multiple depots in the system; each vehicle leaves one of the depots and after visiting the customers it must return to the depot where it started the route. Multiple Products (VRP with Multiple Products) [13] means that the system includes several types of products; this means that each customer may have multiple product demands. Multiple Time Windows (VRP with Multiple Time Windows) [14] means that several time intervals were given to serve the demands of the products of the customers; one of the time intervals is chosen to satisfy the demands. In the case of Occasional Drivers (VRP with Occasional Drivers) [15], the company is not restricted to using its own vehicles and can also rent vehicles. Open Route (Open VRP) [16] means that there are multiple depots in the system; each vehicle leaves one of the depots and after visiting the customers it does not have to arrive at one of the depots. Periodic Problem (Periodic VRP) [17] is a case where a period is given in the system. In the case of Perishable Food Products Delivery (VRP with Perishable Food Products Delivery) [18], the expiry date of the products must also be considered during delivery. Pickup and Delivery (VRP with Pickup and Delivery) [19] means a combination of delivery and pickup. For this problem, certain products are shipped from the depots to the customers, while other products are collected from the customers to the depots. In the case of the Risk Constrained VRP [20], a safety factor is also given; the objective is to maximize route safety. The Selective Vehicle Routing Problem [21] means that we only serve the needs of profitable customers. In the case of a Single Depot (VRP with a Single Depot) [22], there is only one depot in the system; vehicles leave this depot and then return to the depot when customers are visited. A Soft Time Window (VRP with Soft Time Window) [23] means that a time window was added to the product demands of each customer, however, in this case, customers can also be visited outside the time window; only then will the solution receive a penalty point. Stochastic (Stochastic VRP) [24] is a problem where the value of some factor of the system (e.g., customer demand, time window) is given with a probability distribution. Time Window (VRP with Time Window) [25] means that a time window has been added to the demands of each customer in the system; customers should be visited within the time window. Traffic Congestion (VRPs with Traffic Congestion) [26] means that during the system, the route time between nodes is not fixed, it varies depending on the traffic. The traveling Salesman Problem [27] is a case when a single salesman (vehicle) visits the cities (customers); the objective is to minimize the distance traveled by the vehicles. Two-Echelon (Two-Echelon VRP) [28] means that the system includes not only depots and customers, but also intermediate distribution points (satellites); the vehicles first transport the products from the depot to the satellites and then from there to the customers.

In the case of the fuzzy multi-level multi-objective fractional decision-making problem (ML-MOFDM) [29] the fractional objectives are given with fuzzy numbers. This problem has multiple objective functions. The supply chain network (SCN) [30] objective is the cost-efficient moving of an item or administration from provider to client. Such a problem is the bi-level multi-objective supply chain model (BL-MOSCM) [30], where the customer demand and supply are uncertain. The assignment problem (AP) [31] is also related to the VRP. In this case, however, tasks must be assigned to people. The ordering must be done in the most ideal way at the lowest cost. The following are the best-known algorithms for the task: linear programming, swarm optimization, and the Hungarian algorithm. In the case of the fuzzy AP [31], the system contains fuzzy data. Another important optimization algorithm used to solve logistics tasks is the bi-level optimization problem (BL-OP) [32]. In the case of an intuitionistic fuzzy multi-objective fractional transportation problem (FIF-MOFTP) [33], the seller's supply, the customer's demand, and the transportation fee are given by fuzzy numbers.

Ontology is a standard tool for knowledge representation. The term comes from philosophy, which means a systematic description of existence. An ontology describes a domain, its concepts, and the relationships between concepts. Its purpose is not only the presentation of information but the automatic processing of the information content. The most common general ontology language is the OWL, developed by the World Web Consortium (W3C). The OWL helps machine interpretation of web content to a greater extent than XML, RDF, and RDF Schema (RDF-S). The OWL also offers an extended vocabulary and formal semantics [34,35].

Ontology defines the concepts necessary for the representation and description of the knowledge area and helps to unify their treatment. Ontologies are used by databases, applications, and users who need to work together in a particular area of knowledge (e.g., medicine, financial management, etc.). Ontologies contain the basic concepts of the knowledge area and their contexts in a computer-interpretable form. Because the Semantic Web requires ontologies with a degree of structure, it must specify the following categories [36]: classes (general concepts) in a wide range of subject areas, relevant relationships between things, and the properties (or attributes) that these things may have.

Ontologies are formulated to make precise, clear, meaningful distinctions between classes, attributes, and relationships between things. The main role of ontology is to unify the concepts: conclusion, specialization, and validation [36].

The main components of the OWL are the classes, properties, and individuals. The classes in OWL contain sets of individuals. Each class has a logical formula that describes whether an individual belongs to a particular class. Classes are often organized in a subclass-parent class hierarchy; this is also called a taxonomy [34]. The OWL properties represent relationships. There are two main property types: object properties and datatype properties. Object properties describe the relationship between two individuals. Object properties link one individual to another. There is another type of property in OWL, it is the annotation property that is used to add metadata to classes, individuals, and object/datatype properties. Properties can have domains and ranges [34]. The individuals represent class instances with concrete property values. In OWL, classes can be described with derivation and restrictions based on other classes [34]. An important feature of ontologies is that they can be processed with a reasoner. It is supported by machines that infer logical operations. One of the main tasks of a reasoner is to decide for each class whether it is a subclass of another class, whether each class can be instantiated, and whether the system contains a contradiction. In ontology, by performing such tests on the classes, the reasoner is the inferred ontology class hierarchy [34].

One of the advantages of applying ontology to the VRP problem is transparency and easy handling. The programmers create the ontology (OWL file) through the Ptotége [37] system. The Protége system provides a graphical interface, we can easily create, modify, and delete ontology elements. It also includes several visualization techniques such as OWLViz [38], VOWL [39], and OntoGraf [40]. If we implement these logics into the program code, we will have a much more difficult task. It is difficult to see through, and visually more difficult to display (perhaps possible with a UML diagram). The generation of the system logic is also more difficult, but with the help of Protége we can graphically generate and edit ontology elements, thus making it easier for the user to write and read the program code, even a non-expert programming user can read and modify the ontology more easily. The ontology can be reused, it can be imported into another ontology, and with another program, you can model a new ontology system and solve a new problem.

In the following, the paper presents the development of a unified description of an ontology-based formalization to describe a wide spectrum of the Vehicle Routing Problems.

The system is implemented in OWL, and the manipulations can be made in standard ontology API, Protége editor for IT expert users, and with a JSON descriptor for logistics expert users. The paper presented a mapping method to convert VRP model into ontology format.

2. Ontology of Transportation Systems

In this section, the most important ontology systems are presented that have been developed over the past years. Figure 1 presents an overview of the ontology systems used in the field of logistics.

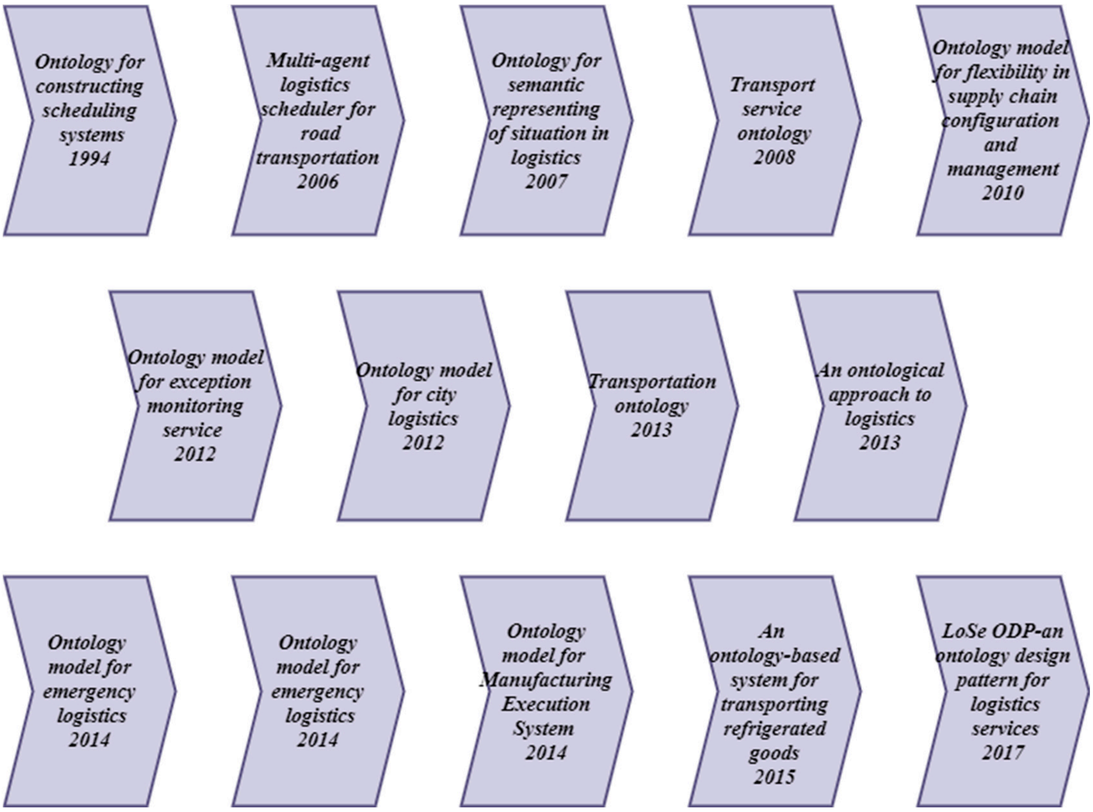

Ontology for constructing scheduling systems 1994
Multi-agent logistics scheduler for road transportation 2006
Ontology for semantic representing of situation in logistics 2007
Transport service ontology 2008
Ontology model for flexibility in supply chain configuration and management 2010
Ontology model for exception monitoring service 2012
Ontology model for city logistics 2012
Transportation ontology 2013
An ontological approach to logistics 2013
Ontology model for emergency logistics 2014
Ontology model for emergency logistics 2014
Ontology model for Manufacturing Execution System 2014
An ontology-based system for transporting refrigerated goods 2015
LoSe ODP-an ontology design pattern for logistics services 2017

Figure 1. Overview of ontology systems used in logistics.

An *ontology for constructing scheduling systems* developed by *Smith and Becker* [41] in 1994 was one of the first works used. The building blocks of the ontology system included the following five classes: 'Demand', 'Activity', 'Resource', 'Product', and 'Constraint'. 'Demand' was the demand for products and services. The 'Product' was associated with 'Demand' and included the following properties: 'Activities', and 'Resources'. 'Resources' supported the implementation of 'Activities'. The following properties were defined for 'Resources': 'Capacity', 'Setup-Duration', and 'Usage-Restrictions'. Properties were also defined for the 'Activities' class: 'Start-Time', 'End-Time', 'Duration' etc.

A *multi-agent logistics scheduler for road transportation* system was reported by *Himoff, Rzevski, and Skobelev* in *2006* [42]. In this paper, a model of intelligent road transport scheduling was presented, which included several functions such as real-time scheduling, scheduling improvement, and scheduling analysis. The scheduler was event-driven, which means that schedules were compiled to consider all events that affect the schedule (e.g., the arrival of a new order, cancellation of an order, truck failure, or lack of travel). When an event occurred, the scheduler rescheduled only the affected part, thereby minimizing the change to resources that had already been allocated. Their system consisted of three main components: 'Basic', 'Operations', and 'Scheduling'.

In an article by *Lian, Park, and Kwon* [43] published in 2007, an *ontology for semantic representation of the situation in logistics* was performed. The following 'Logistics Processes' were covered: 'Outsourcing', 'Manufacture', 'Warehouse', 'Inventory', 'Maintenance', 'Order', 'Packing', 'Load', 'Transportation', 'Unload', 'Distribution', 'Reverse Logistics'. The following 'Logistics Events' elements were defined: 'Material_Arrived', 'Products Exist', 'Warehoused', 'Inventoried', 'Temperature_Regulated', 'Maintaining', 'Stock Check', 'Packed', 'Back Order', 'Loaded', 'Transport', 'Products_Arrived', 'Unload', 'Distributing', 'Back Out'.

Dong, Hussain, and Chang discussed a *transport service ontology* in 2008 [44]. The authors divided the ontology system into layers. The first layer was the root of the hierarchy, which includes all delivery services. The second layer was a specialization of the concept of abstract transport service, comprising four categories: air transport, rail transport, road transport, and shipping services. The third layer was a further specialization of the second layer, in this service the service concepts can be considered concrete or abstract concepts. In the third layer, the abstract concepts had additional specializations in the fourth layer.

In their 2010 publication [45], the authors *Hoxha, Scheuermann, and Bloehdorn* reported an *ontology model for flexibility in supply chain configuration and management*. The approach combined loosely coupled logistics services and semantic technologies to provide a unified representation of different logistics data and service functions. The authors created a framework that used automated, intelligent techniques to discover, rank, and execute efficient assemblies of services. The following main concepts were discussed in connection with the system: 'Process', 'Service', and 'Resource'. 'Process Logistics Process' could be 'Service Logistics Service'. 'Logistics Service' further specialized in 'Service Provider', 'Service Requester', 'Transportation', 'Handling, and 'Storage'.

An *ontology model for exception monitoring service* was presented by *Xu, Wijesooriya, Wang, and Beydoun* in 2012 [46]. The system modeled logistics exceptions. The authors focused on the type of exceptions and the reason for their creation. The main concept of their system was 'Logistics Exceptions'. The following terms were defined below: 'Delivery', 'Warehouse / Storage', 'Order / Forecasting', 'Planning', 'Transportation', and 'Information and Data'. The concept of 'Delivery' was further specified: 'Late Delivery', 'Partial Delivery', 'No / Missed Delivery', 'Wrong Delivery', and 'Delivery Holdings'.

Authors *Anand, Yang, van Duin, and Tavasszy* specifically focused on *an ontology model for city logistics* in their article [47], published in 2012. Their main classes were 'Activity', 'KPI', 'Objective', 'Stakeholder', 'Resource', 'Measure', and 'R&D'. Several subclasses and properties were created for the classes. In their article, they wrote about the following numbers: class 263, object property 56, data property 102, and individual 170.

A *transportation ontology* was developed by the authors *De Oliveira, Bacha, Mnasser, and Abed* in *2013* [48]. The system included the following main classes: 'Calendar', 'City', 'Connection link', 'Connection point', 'Exchange pole', 'Geographic element', 'Geographic element with services', 'Geographic element without services', 'Infrastructure link', 'Journey', 'Journey pattern', 'Operator Institutions', 'Railway element', 'Railway junction', 'Road element', 'Road junction', 'Stop point', 'Stop point in journey pattern', 'Transport line', 'Transport mode', 'Transportation network', 'Vehicle journey', 'Vehicle type', 'Wire element' and 'Wire junction'.

An example of an *ontological approach to general logistics* systems is in the literature. These systems are good because other ontological scientists can supplement the general system with additional classes, relationships, and instances. Such a system was presented by *Daniele and Pires* in 2013 [49]. The following main classes were created: 'Product', 'Package', 'Moveable equipment', 'Transport means', 'Static equipment', 'Facility', and 'Facility structure'.

An *ontology model* has also been developed *for emergency logistics*; such a system was presented by *Zhang, Jiang, Zeng, Ning, and Wanga* in their paper in 2014 [50]. The main components were 'Time', 'Place', 'Equipment', 'Facilities', 'Resource', 'Environment', 'Event', 'Demand', 'Organization', 'Task', and 'Scheme'.

An ontology model for the Manufacturing Execution System was discussed in 2014 by *Fumagalli, Pala, Garetti, and Negri* [51]. The main class of their system was the 'Component' class, which contains 'Storage', 'Transporter', 'Processor', and 'Sensor' classes. Each class was also given datatype properties such as 'ID', 'name', 'type', 'capacity', and so on.

An ontology-based system for transporting refrigerated goods was introduced in 2015 by the authors *Wang, Yi, Zhu, Luo, and Ji* [52], as the supply chain of refrigerated goods is becoming more complex and challenging today. The main classes of ontology were 'Parameter', and 'Diagnostic'. The parameter contained additional subclasses: 'Temperature', 'Humidity', 'Location', 'Time', and 'Product information'.

The LoSe ODP, ontological design pattern sample was published in 2017 by *Glöckner and Ludwig* [53]. The composition of the logistics services of different service providers is a difficult task due to the different formulations, descriptions, and IT systems. Such logistics service building blocks can be easily implemented with a central ontological design pattern. Data from different service providers (services) can be easily accessed, connected, and exchanged within the network. The most important class of the ontological model of the authors was the 'LogisticsService' class, which was related to the 'Constraints' class, the 'Capability' class, and the 'Flow', 'Resource' and 'ServiceLevelAgreements' classes.

3. The Ontology Model for the Generalized Vehicle Routing Problem

In this section, the ontology system of our generalized model of the Vehicle Routing Problem (VRP) is presented. Some of the ontology systems presented in the literature are limited to specific logistics tasks, for example, monitoring service, city logistics, emergency logistics, manufacturing execution system, and refrigerated goods. Some ontology models try to cover the whole logistics process. As the literature lacks a general ontology model of VRP, the goal of our investigation was to develop a novel general model. We needed a system that would help the user choose which components and metrics of our generalized VRP model to use when transporting a given product, to a given node, with a given vehicle. We also saw the need for the system to help decide which products can be shipped with which vehicles to nodes. The advantage of ontologies is that ontology systems can work together, so our system can be supplemented with a logistics model developed by others. The developed system includes the following key concepts: vehicles, products, period, set of values (static, stochastic, fuzzy, forecasted values), and attributes (nodes, vehicles, period, products, costs, functional parameter) detailed in the mathematical model in our paper [54]. Furthermore, metrics are included in the model. In the sample system, ontological concepts allow for rule-based validation. The main components of the ontology model are shown in Figure 2.

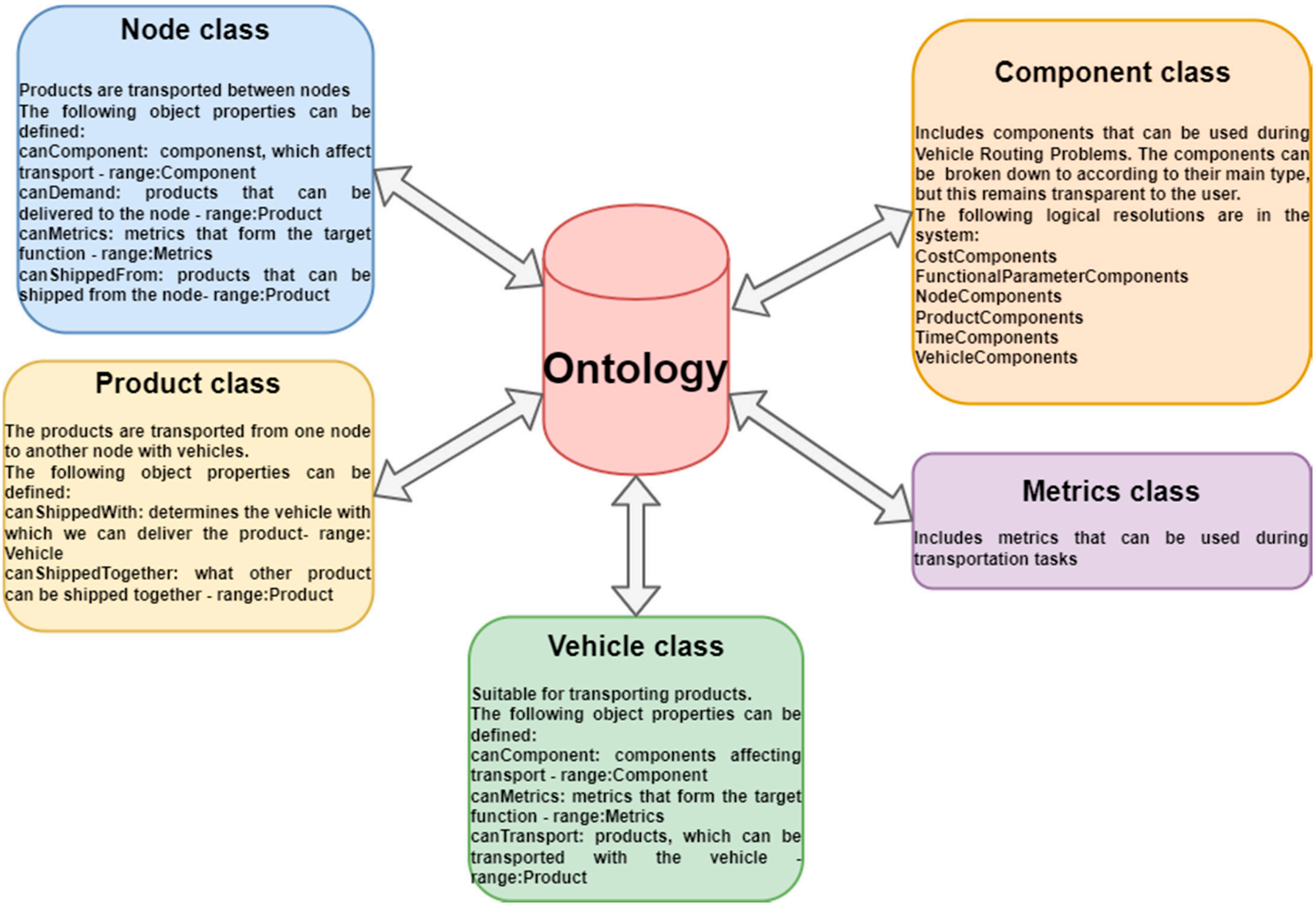
The Classes and Their Properties

In the following, the classes and their hierarchies and properties are presented.

Figures A1 and A2 represent the ontology class hierarchy: Figure A1 shows only the main classes, and Figure A2 also illustrates the subclasses. The figures present how many classes our system contains. These classes will be detailed in the following.

Figure 3 illustrates the VOWL representation of our ontology system. The VOWL representation represents ontologies as a graph and provides a tangled representation for large ontologies.

The 'BaseComponent' class (Figure 4) presents the base components of the system, which are the following: 'Vehicle', 'Product', 'Node', 'Period', and 'Point'. 'Vehicle' transport products from one node to another. Almost all components also depend on the vehicle, such as the distance between nodes, how long it takes to get from one node to another, and so on. 'Product' is also one of the main building blocks of the system because the products are transported from one node to another. The products affect almost every other component of the system, such as what vehicle to transport or what nodes to visit. 'Period' is a time interval. Within this time interval, the nodes need to be visited several times, because they may have several product demands. 'Period' is also a very important building block because almost every component depends on it. For example, the time it takes to travel between two nodes depends on the period because it does not matter whether the products are delivered during rush hour or at night. 'Point' means that during transport tasks, products are transported from one position to another. Neither class has an object property.



Node class

Products are transported between nodes
The following object properties can be defined:
canComponent: componenst, which affect transport - range:Component
canDemand: products that can be delivered to the node - range:Product
canMetrics: metrics that form the target function - range:Metrics
canShippedFrom: products that can be shipped from the node- range:Product

Component class

Includes components that can be used during Vehicle Routing Problems. The components can be broken down to according to their main type, but this remains transparent to the user.
The following logical resolutions are in the system:
CostComponents
FunctionalParameterComponents
NodeComponents
ProductComponents
TimeComponents
VehicleComponents

Product class

The products are transported from one node to another node with vehicles.
The following object properties can be defined:
canShippedWith: determines the vehicle with which we can deliver the product- range: Vehicle
canShippedTogether: what other product can be shipped together - range:Product

Metrics class

Includes metrics that can be used during transportation tasks

Vehicle class

Suitable for transporting products.
The following object properties can be defined:
canComponent: components affecting transport - range:Component
canMetrics: metrics that form the target function - range:Metrics
canTransport: products, which can be transported with the vehicle - range:Product

Figure 2. The ontology system.

The 'Node' class has several subclasses based on out-plant material handling places. The nodes can be classified, such as 'Landfill', 'ResidentialHouse', 'Bank', 'Warehouse', 'Hospital', 'Factory', 'Shop', and 'CustomsPlace'. Some classes may be further specialized, such as 'Warehouse', 'Factory', and 'Shop' classes.

The 'Product' class has also several subclasses, depending on what products are shipped. These are the 'People', 'Food', 'Money', 'ElectricProduct', 'BulkProduct', 'Garbage', 'PostalPackage', 'Toys', 'HorticultureProduct', 'Book', 'Furniture', 'Clothing', 'LiveAnimalProduct', 'Medicine'.

The 'Vehicle' class also has subclasses according to the transportation units, for example, 'Train', 'WaterCraft', VehicularVehicle', and 'AirCraft'. These classes can be also categorized into subclasses. The 'Train' can be 'GeneralPurposeTrain', and 'SpecialPurposeTrain'. The 'WaterCraft' can be 'SeaShip' and 'RiverBoat', the 'VehicularVehicle' can be 'GarbageTransporter', 'MoneyTransporter', 'Truck', 'Bus', 'Ambulance' and 'Car'. The 'AirCraft' can be categorized into the following subclasses: 'PersonAndCargoPlane', 'CargoHelicopter', and 'CargoPlain'.

Another class, which subclass is the 'owl:Thing' is the 'ValueComponent' class. 'ValueComponent' means that in the system the values of each component (such as the product demands of the nodes, the capacity constraints of the vehicles, etc.) can be a different type. The 'ValueComponent' class has the following subclasses: 'Static', 'Stochastic', 'Fuzzy', and 'Forecasted'. 'Static' value means that the value is given exactly (with a single number). The 'Stochastic' value indicates that the probability distribution of the values is given only.

'Fuzzy' value means that the factors are given in fuzzy numbers. Four values are given, between which the component can take the actual value. 'Forecasted' value is when only historical data are given. In this case, the values, and the dates (times) are given. From these past data, future data should be predicted using forecasting techniques.

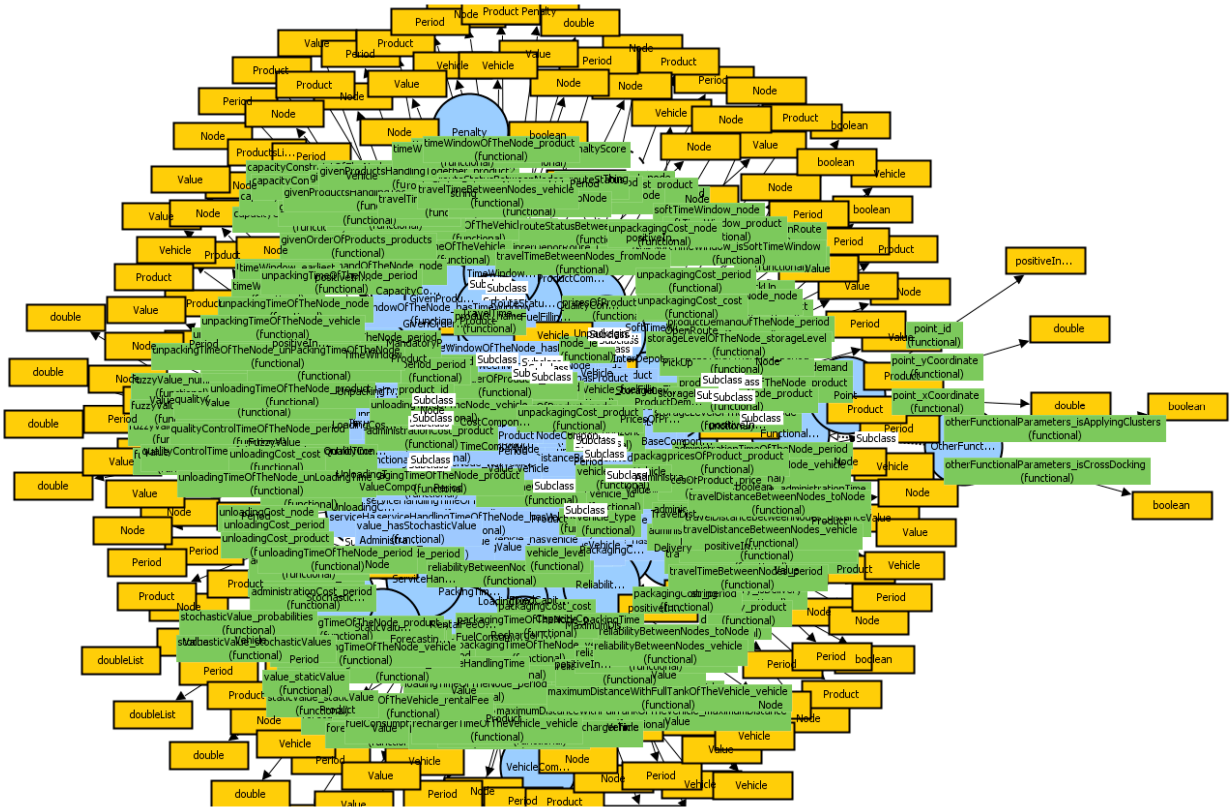


Figure 3. The VOWL representation of our ontology system.

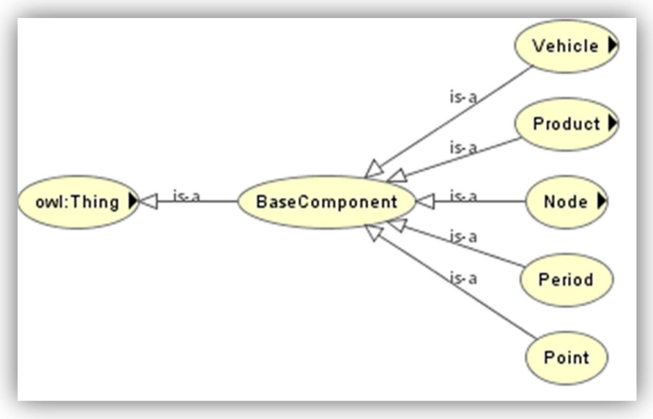


Figure 4. The BaseComponent class.

The 'NodeComponent' (Figure 5) class consists of classes that represent components in connection with two 'Nodes'. The 'NodeComponent' class has the following subclasses: 'TravelTimeBetweenNodes', 'RouteStatusBetweenNodes', 'ReliabilityBetweenNodes', and 'TravelDistanceBetweenNodes'. The 'TravelTimeBetweenNodes' is an important factor, the objective is to deliver the products in the shortest time. Travel time also depends on the period, because in the morning and afternoon it usually takes more time to take a route between two nodes than at night. The 'TravelDistanceBetweenNodes' means the distance between two nodes, which is an important factor because the objective is for vehicles to travel as little distance as possible, then they consume less fuel. The distance between two nodes depends on the type of vehicle and may be different for water or land transport. The 'ReliabilityBetweenNodes' can often be an important factor. The objective is to transport by the most reliable route due to the robberies. The vehicle type also depends on this factor because transport can be not safe for ground transport, but it is possible for air transport. 'RouteStatusBetweenNodes' is important due to the depreciation of the vehicle. The route status factor also depends on the vehicle type, for example, it can be higher between two nodes for ground transport than for water transport.

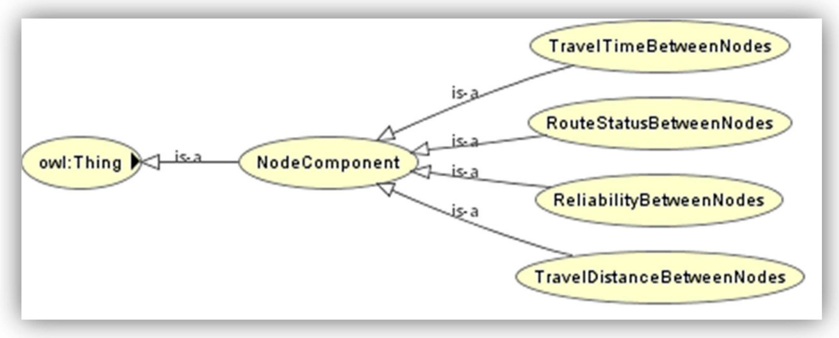


Figure 5. The NodeComponent class.

The 'VehicleComponent' (Figure 6) class describes the components in connection with vehicles. It has the following subclasses: 'RechargerTimeOfTheVehicle', 'RentalFeeOfTheVehicle', 'CapacityConstraintOfTheVehicle', 'FuelConsumptionOfTheVehicle' and 'MaximumDistanceWithFullTankOfTheVehicle'. The 'RechargerTimeOfTheVehicle' class means that each vehicle may have a different recharger time. 'RentalFeeOfTheVehicle' means that each vehicle has a different rental fee. It is worth using your vehicles first, they have no rental fee. The 'CapacityConstraintOfTheVehicle' means that the vehicles have a capacity limit for the products to be transported. A vehicle may not be able to transport a certain product (for example, refrigerated goods require special transport units). 'FuelConsumptionOfTheVehicle' means that each vehicle type may have a different fuel consumption. 'MaximumDistanceWithFullTankOfTheVehicle' presents that each vehicle type has different fuel consumption and can cover a maximum distance with a maximum tank.

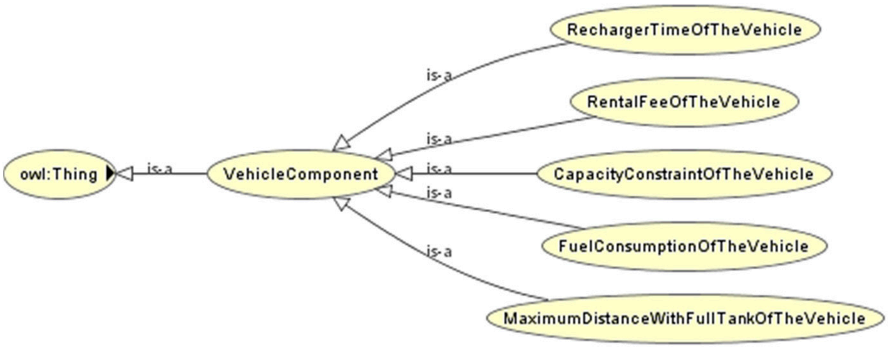

owl:Thing
is-a
VehicleComponent
is-a
RechargerTimeOfTheVehicle
is-a
RentalFeeOfTheVehicle
is-a
CapacityConstraintOfTheVehicle
is-a
FuelConsumptionOfTheVehicle
is-a
MaximumDistanceWithFullTankOfTheVehicle

Figure 6. The VehicleComponent class.

The 'TimeComponent' (Figure 7) class has subclasses in connection with a time component. This class has the following subclasses: 'ServiceHandlingTime', 'PackingTime', 'UnpackingTime', 'LoadingTime', 'UnloagindTime', 'FixedCapitalTime', 'AdministrationTime', 'QualityControlTime', 'TimeWindow' and 'TimeWindowOfTheProduct'. The 'ServiceHandlingTime' means that vehicles can not only transport products to the nodes, but they can also perform some service (such as maintenance). The 'PackingTime' class means that the products are often shipped wrapped; this component shows the packing time. The 'LoadingTime' means that the products to be transported are loaded into the vehicles. The value of a component depends largely on the vehicles and the products. 'FixedCapitalTime' means that the products are available at intermediate locations (satellites) for a while. 'AdministrationTime' means that an administrative activity must be carried out on the arrival and departure of the products. The 'QualityControlTime' has the following meaning: the products are subject to any quality control before departure and upon arrival at the node. The time for this is shown by that factor. The 'TimeWindow' means that the specific product demand of a given node can only be met within a time interval. It has two properties: 'Earliest' and 'Latest'. The 'TimeWindowOfTheNode' means the time windows of the node.

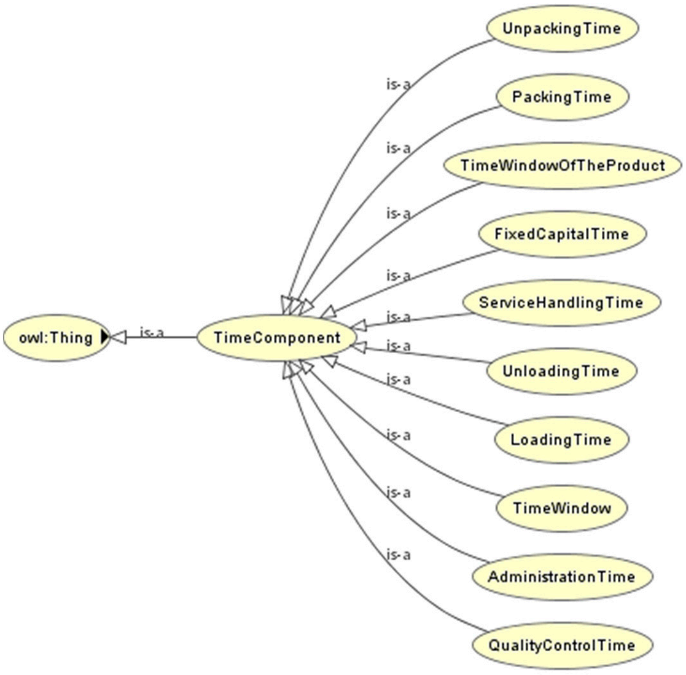


Figure 7. The TimeComponent class.

The 'ProductComponent' (Figure 8) class is responsible for the components in connection with products. The parent class of the 'ProductComponent' class is the 'owl:Thing', and it has the following subclasses: 'CapacityConstraintOfTheNode', 'GivenOrderOfProducts', 'ProductDemandOfTheNode', 'GivenProductsHandlingTogether', 'PricesOfProduct' and 'StorageLevelOfTheNode'. The 'CapacityConstraintOfTheNode' has the following meaning: nodes have a storage capacity limit for each product. The capacity limit cannot be exceeded. The nodes may expect the products in a specific order. It means that after product A, only product B can arrive. For example, the node first processes product A and only then can process product B; if product B arrives first, it should be stored. This is shown by 'GivenOrderOfProducts'. 'GivenProductsHandlingTogether' means whether certain products can be shipped at the same time, with one vehicle.

The 'CostComponent' (Figure 9) class represents the components in connection with costs in our system. The 'owl:Thing' is the parent of the 'CostComponent' class, and it has the following subclasses: 'PackagingCost', 'UnpackagingCost', 'LoadingCost', 'UnloadingCost', 'AdministrativeCost' and 'QualityControlCost'. The 'PackagingCost' has the following meaning: each item is packaged at each node. The cost of this can vary by vehicle and node. The 'UnpackingCost' can be also important because each item is unpackaged at each node. The cost of this can vary by vehicle and node. The 'LoadingCost' has the following meaning: each item is placed on the transport device (vehicle) at each node. The cost of this can vary by vehicle and node. The 'UnloadingCost' means that each item is unloaded from the transport device at each node; the cost of this can vary by vehicle and node. The 'AdministrativeCost' is important because, for each product, there is an administration cost departure from the node or on arrival at the node. The 'QualityControlCost' class can also be important because the products may also be subject to quality control. In the case of each cost component, the objective is to minimize the cost value.

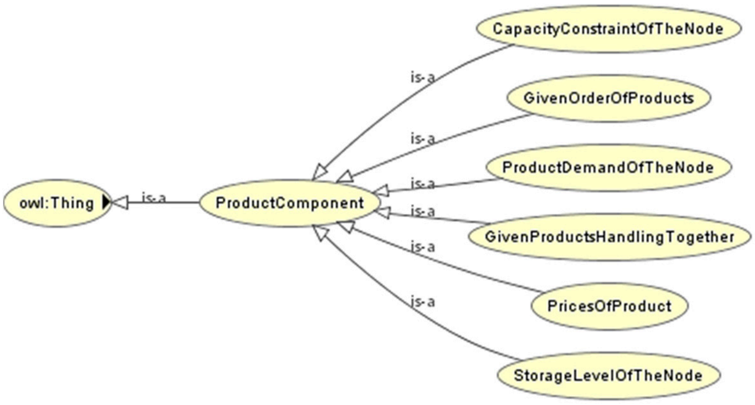


Figure 8. The ProductComponent class.

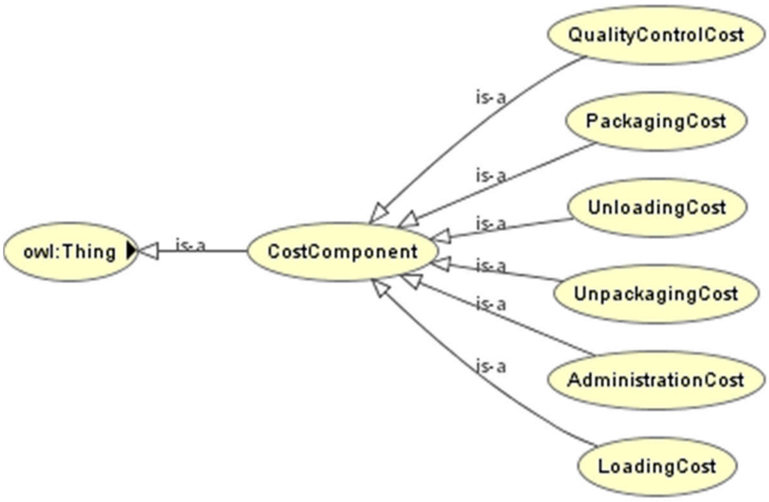


Figure 9. The CostComponent class.

The ‘FunctionalParameterComponent’ class is the subclass of ‘owl:Thing’, and this class has also subclasses, which are the following: ‘OpenRoute’, ‘PickUp’, ‘InterDepotRoute’, ‘SoftTimeWindow’ and ‘Delivery’. The ‘OpenRoute’ means that the vehicles start their route from the higher-level nodes and then visit the lower-lever nodes but do not return to the higher levels. ‘PickUp’ class has the following meaning: it is the reverse of delivery; the products start from the lower-level nodes and the transportation takes place to the higher-level nodes. A combination of pickup and delivery is also possible. An important functional parameter is the ‘InterDepotRoute’. If there is an inter-depot route, vehicles will start from the upper levels but may return to any of the higher-level nodes after visiting the lower-level nodes. In case of the ‘Delivery’, the products are transported from the higher levels to the lower levels. This type depends on the position and the type of products. When ‘SoftTimeWindow’ is applied, we can also serve the product demand of the node outside the time window, but then the solution gets a penalty point.

The ‘Metrics’ class indicates the metrics, from which the objective function can be consisted of. The objective can be the minimization of the following metrics: ‘LengthOfTheRoute’, ‘PackagingCost’, ‘UnpackagingCost’, ‘LoadingCost’, UnloadingCost’, ‘AdministrativeCost’, ‘QualityControlCost’, ‘FuelConsumption’, ‘VehicleRentalFee’ RouteTime’, ‘PackagingTime’, ‘UnpackagingTime’, ‘LoadingTime’, ‘CapitalTime’, ‘PackagingTime’, ‘UnpackagingTime’, ‘LoadingTime’, ‘CapitalTime’, ‘AdministrativeTime’, ‘QualityControlTime’, ‘FuelFillingTime’, ‘WaitingTimeOfTheNodes’, ‘ExceedingTimeWindow’ and ‘Unvisited-

Customers'. For the following metrics, maximization is the objective: 'TransportedValue', 'ReliabilityBetweenNodes', and 'RouteStatusBetweenNodes'.

4. The Application of the Ontology System

The proposed system is a user-friendly interface because instead of ontology-specific queries (DL Query, SPARQL Query), the user only needs to specify a JSON descriptor, which is an easy-to-read format for humans

The optimization of VRPs has been characterized as one of the most important success stories of logistics and supply chain management, providing and facilitating optimal solutions for vehicle fleets in a wide range of real-life applications including in-plant logistics and external supply chain solutions. The optimal solution for VRPs in logistics systems is especially challenging because they are characterized as one of the most challenging combinatorial optimization problems of logistics design and supply chain management. While optimizing the VRPs of real-life logistics applications, the following important questions must be answered:

- Number of vehicles, related material handling equipment (loading and unloading, packaging), and human resources (drivers, operators) required to fulfil the predefined demands of customers.
- The capacity of vehicles and related material handling equipment.
- Scheduling of vehicles, material handling equipment, and human resources.
- Required qualification of human resources.
- Structure, location, and relationship of objects in the supply chain (number of tiers, collection, and distribution centers, cross-docking facilities, suppliers, users, third-party logistics service providers).
- Assignment of objects, locations, resources, and products.
- Definition of delivery strategies.
- Definition of objective functions (cost, capacity utilization, flexibility, efficiency, availability, environmental impact, GHG emission, sustainability) and constraints (capacity, time).
- The use of the chosen collection, distribution, and delivery strategies to reduce costs and improve performance by decision-makers and policymakers in the logistics area and supply chain management.

The system is illustrated in Figure 10, where the model architecture is motivated by the proposal of Li, Hsieh, and Sun [55]. The ontology system is created so, that it can be handled by two types of users. One of the users (the expert user), who has adequate IT and logistics knowledge, does the following: purpose, scope, and requirement identification, then concept collection, ontology creation, and analysis. This user accesses the ontology through the ontology editor, where the user can implement it. The other user is the end-user who has adequate logistical knowledge. They ask questions to the system using a Java program and a JSON descriptor. The query engine layer of the Java program generates SPARQL queries, thereby extracting information from the ontology, and the Java program responds to the user. The ontology layer consists of the RDF/OWL file itself and the reasoner. In the figure, we marked each layer in red and each operation in blue.

In the following, the JSON descriptor is introduced, which allows the user to ask questions to the system.

The ontology sample system (Figure 11) assists in the optimal organization of transport tasks. If the user knows what types of nodes to visit during the transport task, the system helps to tell which products can be transported to or from the node and what constraint and objective function components can be used during the transport task. If the user knows which products the delivery is limited to, the sample system helps to decide what type of vehicles can be used. If the type of vehicle is known during the delivery task, the system tells us in response what products can be delivered with the vehicle, and what constraint components and objective function components can be included in the optimization. If the system input parameter is a node vehicle type, the response is the type of possible products to be transported to and from the node, the vehicles, the possible vehicle routing components, and the possible objective function components. If the system receives nodes and products as input, it returns other products, vehicle types, vehicle routing components, and objective function components that can be transported to and from the nodes. A list of vehicles and products is provided, and the system returns the products that can be shipped to the node, the products that can be shipped from the node, the list of possible other vehicles, the components, and metrics. If a node, a vehicle, and a product are also specified, the system responds with products, vehicles, components, and objective function metrics.

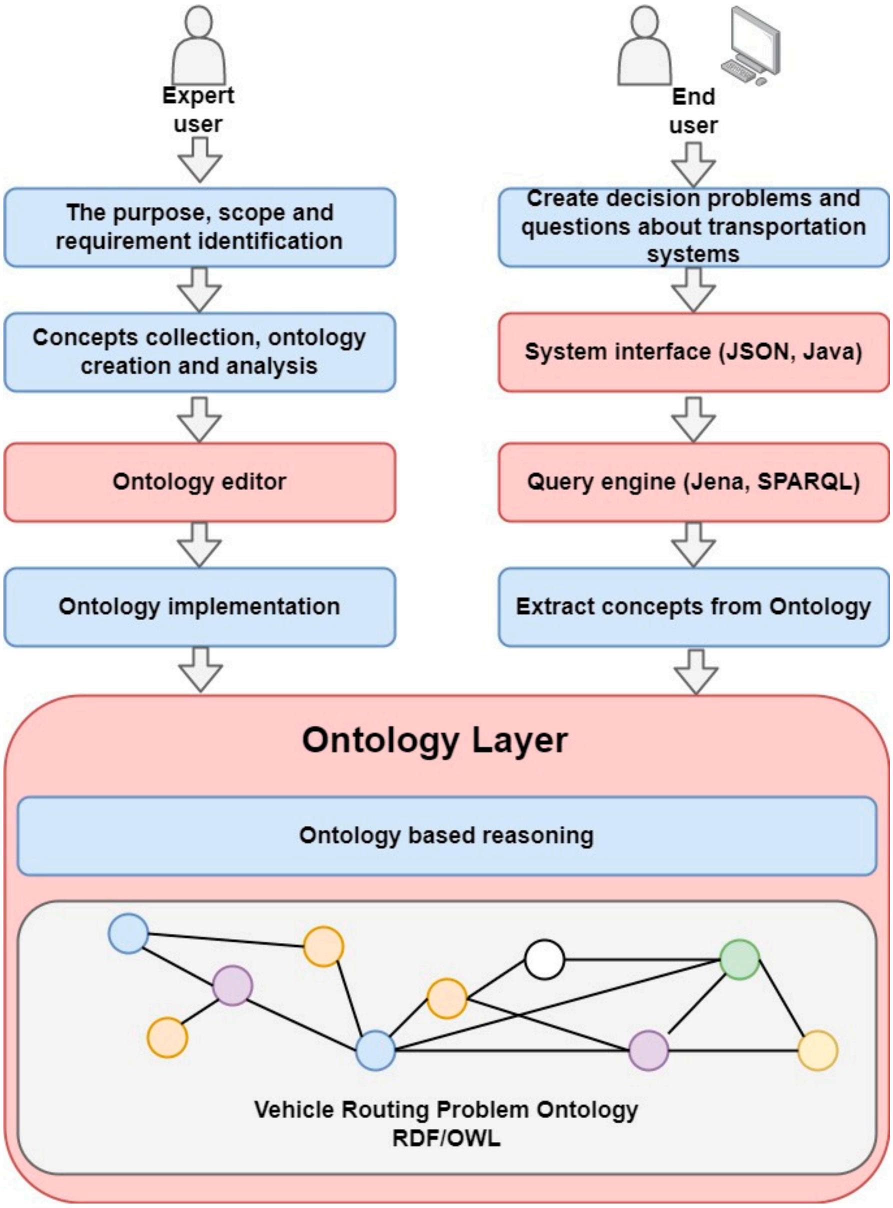

Expert user
The purpose, scope and requirement identification
Concepts collection, ontology creation and analysis
Ontology editor
Ontology implementation
End user
Create decision problems and questions about transportation systems
System interface (JSON, Java)
Query engine (Jena, SPARQL)
Extract concepts from Ontology
Ontology Layer
Ontology based reasoning
Vehicle Routing Problem Ontology RDF/OWL

Figure 10. The architecture of the ontology-based system.

The new assumptions of our research are the followings:

- Automatic integrity test.
- Automatic decision support in the design process.
- Unified domain model, reusability.
- Extensive description of the VRP system.
- None of the publications we found in the field of logistics satisfied the description of our system, they describe only a small part, only a few classes and properties that are common.

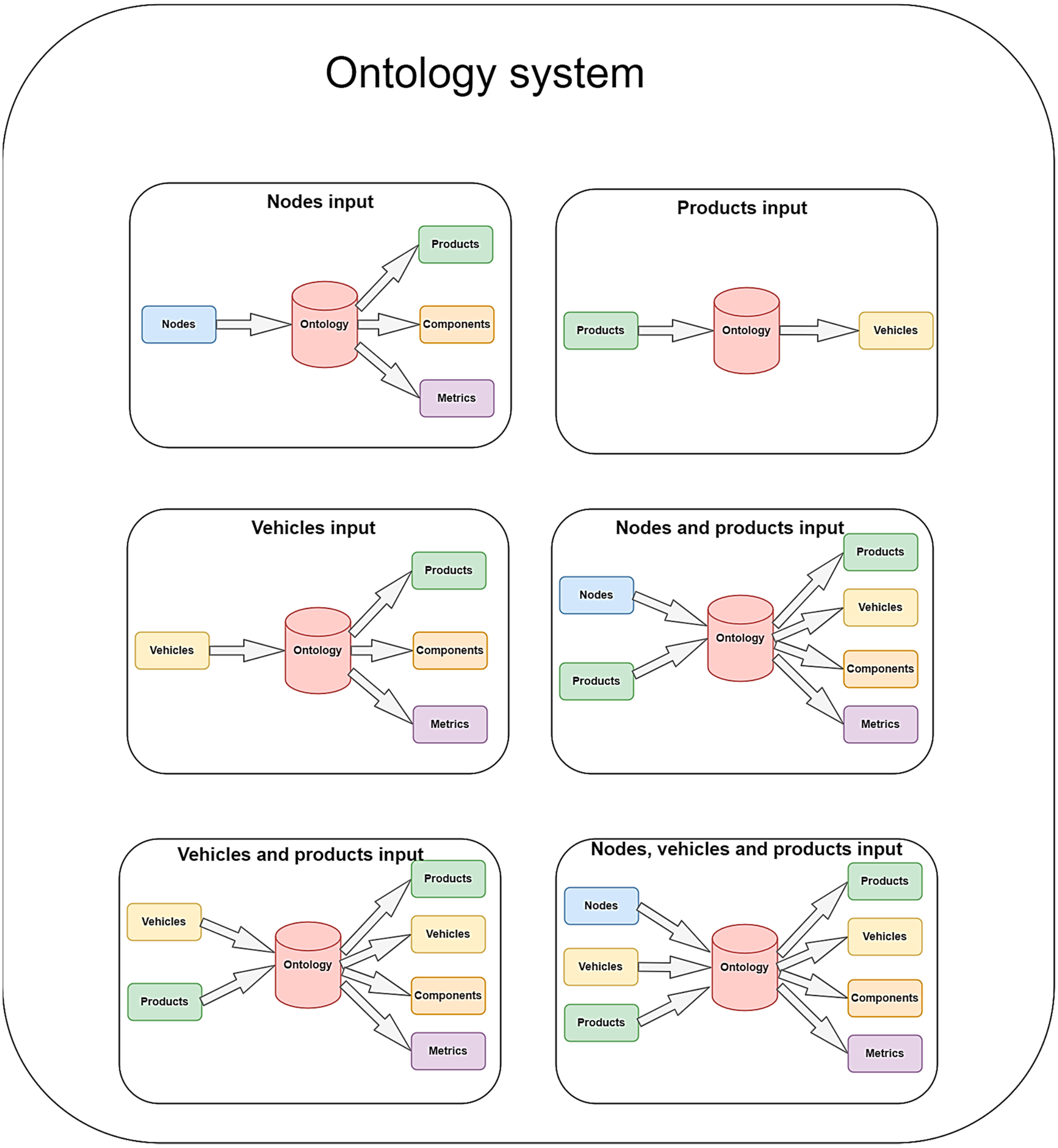


Figure 11. Ontology system input-output values.

The system has the following limitations:

- The research involves finite ontology, it does not cover all special cases.
- The extension of the ontology model is a costly operation.
- For large systems, the reasoning process may be relatively slow.
- If only products are specified in the query, we only get vehicles, we do not receive the component and metrics section, nor the nodes section.
- If nodes are specified in the query, we do not receive the vehicles and products section.
- If vehicles are specified in the query, we do not receive the nodes section.
- This VRP ontology can only work from a given set. This means that queries and responses are limited. We can only perform queries for vehicle types, node types, and product types that are included in the system. As a response, we also receive the

product type, node type, vehicle type, VRP component and metrics that the system contains. This has a great impact on the result, currently we tried to cover a wide range of VRP, in a general way, covering all delivery tasks, but in the future. there will certainly be new delivery tasks, new types of VRP, and these will have to be integrated into the system.

- Adding many new elements (product type, node type, vehicle type, VRP component and metrics) to the system takes a lot of time, even with an ontology editor.

In the literature review, we presented articles investigated logistics ontologies. These ontologies cannot be used for our task. Below, we present the parts and building blocks that can be used for a part of our system (Table 1).

Table 1. The parts and building blocks that can be used for a part of our system.

Publication	Common Part	Limitations
Ontology for constructing scheduling systems by Smith and Becker [41]	The following classes are useful when designing a VRP ontology system: 'Demand', 'Activity', 'Resource', 'Product', and 'Constraint', 'Capacity', 'Setup-Duration', and 'Usage-Restrictions', 'Customer order', 'Movement request', 'Temporal Relations', 'Priority', 'Quantity,' 'Material'. Properties have also been defined for the 'Activities' class: 'Start-Time', 'End-Time', 'Duration'	It contained a couple of elements that are found in our VRP ontology, but these were insignificant and some elements that were included in the authors' ontology were not needed. This ontology is only suitable for scheduling systems.
Multi-agent logistics scheduler for road transportation system by Himoff, Rzevski, and Skobelev [42]	The following classes are useful when designing a VRP ontology system: 'Geo location', 'Order', 'Client', 'Cargo', 'Carrier', 'Route'.	It can only be used for road transportation, only a few classes from the system could be used. Some classes can only be taken from it.
Ontology for semantic representing the situation in logistics by Lian, Park, and Kwon [43]	The following classes are useful when designing a VRP ontology system: 'Time', 'Location', 'Deadline', 'Time end', 'Time begin', 'GPS location', 'Product packed', 'Time duration'.	This ontology does not decide our questions either (which vehicle can be used to deliver the goods, which goods can be transported to each other, which VRP components and objective functions are used during the task, these are missing). And it doesn't include many classes.
Transport service ontology by Dong, Hussain, and Chang [44]	The following classes are useful when designing a VRP ontology system: 'Road transport service', 'Air transport service', 'Rail transport service', 'Shipping service',	This ontology only implemented the transport service, the article presents few ontological elements.
Ontology model for flexibility in supply chain configuration and management by Hoxha, Scheuermann, and Bloehdorn [45]	'Company', 'Service provider', 'Service requester', 'Transportation', 'Storage', 'Delivery flexibility', 'Delivery reliability', 'Delivery quality', 'Lead time',	Only some classes were shown.
Ontology model for exception monitoring service by Xu, Wijesooriya, Wang, and Beydoun [46]	The following classes were useful when designing a VRP ontology system: 'Delivery', 'Warehouse/storage', 'order/forecasting', 'Planning', 'Transportation',	Only some classes were shown.

Table 1. *Cont.*

Publication	Common Part	Limitations
Ontology model for city logistics by Anand, Yang, van Duin, and Tavasszy [47]	The following classes could be added to our system as extensions, which belong to the objective class: 'Emission_reduction', 'Other_nuisance_reduction', 'Congestion_reduction', 'Safety_objective', 'Health_objective', 'Liveability_objective'. Azok amelyek a 'Resource' osztályhoz tartoznak: 'Equipment', 'Infrastructure', 'Fuel', 'Goods', 'Personnel', 'Software', 'Warehouse', 'Depot'. The following subclasses from the 'Activity' class could be used in the system: 'Receiving', 'Storing', 'Ordering', 'Loading', 'Unloading', 'Breaking', 'Validating',	Many classes are common to or have similar meanings as in our system, but this system contains many classes (almost 300), so many classes would be redundant for our system.
Transportation ontology by De Oliveira, Bacha, Mnasser, and Abed [48]	The 'Transport mode' class, which can be used, had subclasses: 'Train', 'Metro', 'Tram', 'Bus'. A 'Geographic element' osztály alosztályai: 'Road element', 'Wire element', 'Railway element'.	The ontology is only focused on passenger transport. Some classes are common to our system.
Ontological approach to general logistics by Daniele and Pires [49]	'Product' class with the following properties: - Type - State - is Refeer - is Dangerous - is Oversized 'Package' class with the following properties: - Type - Quantity - Volume 'Moveable equipment' class has the following properties: - Type - ID - Volume - Quantity 'Transport means', 'Static equipment', 'Facility', 'Facility structure' classes would also be usable.	There are some classes and their attributes that are common to our system, but our system contains many more classes.
Ontology model for the manufacturing execution system by Fumagalli, Pala, Garetti, and Negri [51]	The following classes are useful when designing a VRP ontology system: 'Compartment' 'Storage', 'Transporter', 'Product', 'Customer order'	The authors presented only a few classes, and only a few of them are applicable to our system.
Ontology for transporting refrigerated goods by Wang, Yi, Zhu, Luo, and Ji [52]	The following classes are useful when designing a VRP ontology system: 'Temperature', 'Humidity', 'Location', 'Time', 'Product information'.	This ontology is only applicable for refrigerated goods
LoSE ODP, ontological design pattern by Glöckner, and Ludwig [53]	The following classes are useful when designing a VRP ontology system: 'LogisticsService', 'Capability', 'Cost', 'Quantity', 'Quality', 'Location', 'Time', 'Information', 'Good', 'Control', 'Resource'.	It does not contain many classes as described, only a few classes can be used in our system.

Reasoning Engine, Logic

The inference module in the ontology is a code, a piece of software. It infers logical consequences from facts and axioms. It supports decision-making and performs an integrity test.

In the following, we will generally show how to generate answers to user questions using the reasoning engine of our ontology system.

Key components of the example ontology: $NODE = \{node_1, \ldots, node_n\}$, transporting $PRODUCT = \{product_1, \ldots, product_m\}$ with $VEHICLE = \{vehicle_1, \ldots, vehicle_k\}$:

Question 1.:

What other products can be shipped from $NODE = \{node_1, \ldots, node_n\}$ with $VEHICLE = \{vehicle_1, \ldots, vehicle_k\}$ if $PRODUCT = \{product_1, \ldots, product_m\}$ are also transported?

Answer 1:

$$\begin{gathered} \{canShippedFrom(node_1, x) \vee \ldots \vee canShippedFrom(node_n, x)\}) \\ \bigwedge\{canShippedTogether(product_1, x) \vee \ldots \vee canShippedTogether(product_m, x)\} \\ \bigwedge\{canTransport(vehicle_1, x) \vee \ldots \vee canTransport(vehicle_k, x)\} \end{gathered} \tag{1}$$

Question 2.:

What other products can be shipped to $NODE = \{node_1, \ldots, node_n\}$ with $VEHICLE = \{vehicle_1, \ldots, vehicle_k\}$ if $CT = \{product_1, \ldots, product_m\}$ } are also transported?

Answer 2.:

$$\begin{gathered} \{canDemand(node_1, x) \vee \ldots \vee canDemand(node_n, x)\} \\ \bigwedge\{canShippedTogether(product_1, x) \vee \ldots \vee canShippedTogether(product_m, x)\} \\ \bigwedge\{canTransport(vehicle_1, x) \vee \ldots \vee canTransport(vehicle_k, x)\} \end{gathered} \tag{2}$$

Question 3.:

What vehicles can be used to transport $PRODUCT = \{product_1, \ldots, product_m\}$ to $NODE = \{node_1, \ldots, node_n\}$?

Answer 3.:

$$canShippedWith(product_1, x) \vee \ldots \vee canShippedWith(product_m, x) \tag{3}$$

Question 4.:

What Vehicle Routing components can be used when visiting $NODE = \{node_1, \ldots, node_n\}$ with $VEHICLE = \{vehicle_1, \ldots, vehicle_k\}$ if $PRODUCT = \{product_1, \ldots, product_m\}$ are transported?

Answer 4.:

$$\begin{gathered} \{canComponent(node_1,\ x) \vee \ldots \vee canComponent(node_n,\ x)\} \\ \vee\{canComponent(vehicle_1,\ x) \vee \ldots \vee canComponent(vehicle_k,\ x)\} \end{gathered} \tag{4}$$

Question 5.:

What Vehicle Routing metrics can be used when visiting $NODE = \{node_1, \ldots, node_n\}$ with $VEHICLE = \{vehicle_1, \ldots, vehicle_k\}$ if $PRODUCT = \{product_1, \ldots, product_m\}$ are transported?

Answer 5.:

$$\begin{gathered} \{canMetrics(node_1,\ x) \vee \ldots \vee canMetrics(node_n,\ x)\} \\ \vee\{canMetrics(vehicle_1,\ x) \vee \ldots \vee canMetrics(vehicle_k,\ x)\} \end{gathered} \tag{5}$$

With the help of the inference engine, we can also check consistency. In our system, consistency check contains the following: the vehicles do not exceed their capacity limit, which products can be transported together, which products can be transported by a specific vehicle type.

5. Applications of the General Vehicle Routing Model in Case of Out-Plant Material Handling

In this section, some case studies of our ontology model are presented. The most important out-plant transport case studies are detailed, which are: the transport of short-term food, transport of refrigerated products, tank transport, transport of parcels, and money transportation. During the tests, we provide examples for entering both Node, Vehicle, and Product inputs. The most important inputs were given during the given case study, and the most important outputs issued by the program are listed here. The system was validated by human validation: we checked which of its elements were still possible for the given task.

5.1. Transportation of Perishable Food

In this delivery type, using the tiered system (depot-satellite-customer), only one depot and several customers are distinguished. There is no need for satellites because the bakeries are located near the shops.

Among the relationships between the nodes, the travel time between the locations, the travel distance between the locations, and the route status between the locations can be considered. There are several types of vehicles that can deliver baked goods, but the company has a single type of vehicle. In the vehicle attributes, the following components can be considered: the capacity constraints of the vehicles, and fuel consumption. Owned or borrowed vehicles and the rental fee component can also be considered.

Among the temporality attributes, the service handling time of the locations is not considered because no service is performed. When the packing time of the location is considered, it only makes sense in the depot (bread, bakery). The loading time of the locations only makes sense in the depot. The unloading time of the location makes sense for customers. The fixed capital time of the locations is not considered, there is no fixed capital. The administration time of the locations can be considered for the nodes, as can the quality control time of the locations. Fuel filling time can also be considered, but this is not a significant factor during transport. The time window, on the other hand, is even more important because both the depot and the shops (customers) have opening hours. There can be multiple time windows, as the store (customers) may not always accept goods.

Among the attributes of products and services, the capacity constraint of each location per the product type, and their demand for products by product type can be considered. The storage level of the locations may be worth considering. It is also worth considering the prices of the product.

Among the metrics, minimization of the length of the route is an important factor; maximization of profit can also be an important factor, where maximization of sold values, minimization of loading costs, minimization of administrative costs, minimization quality control cost, minimization fuel consumption, and minimization rental fee can be considered. Route status maximization (minimization of the idle time of vehicles). Time minimization can be also an important metric, it can contain the followings: minimization of route time, minimization of packaging time, minimization of loading time, minimization of administrative time, minimization of quality control time, and minimization of waiting time for nodes. Among the penalty points, the following can be considered: exceeding the number of suppliers, exceeding the number of suppliers.

Each component depends on the temporality (period). The period can be given, for example, a week and the stores must be served 7 times a week. But we can even bring a product to the store several times a day. The attributes may be different during each period, e.g., the store's bakery needs will be different during the week than on weekends, and opening hours will be different.

The first test run included the following parameters: products (fruit, vegetable). The result was the following vehicle list: truck, refrigerator train, vehicular vehicle, and sea ship. The validation of the first test run was the following: truck, refrigerator train, vehicular vehicle, and sea ship are correct. There were no parameters that were missed by the system.

The second test run included the following parameters: products (dairy products). The result was the following vehicle list: vehicular vehicle. The validation of the second test run was the following: vehicular vehicle is correct. The following elements were missed by the test run: Perishable product train, tanker train, liquid transporter ship, refrigerator ship, liquid transporter ship.

The third test run included the following parameters: products (meat). The result was the following vehicle list: truck, refrigerator train, and vehicular vehicle. The validation of the third test run was the following: truck, refrigerator train, and vehicular vehicle classes were correct. There was no other class that we should have received during the test run (because these classes also have their subclasses).

5.2. Tank Transport

During tank transport, liquid products are transported in containers. Such liquid products may be water, gasoline, milk, oil, etc. This requires special vehicles. In this type of transport, usually, only transport one type of product can be transported at a time. The route status between nodes is important. Packing means pouring into the container, unpacking means pouring out of the container. Delivery and pickup of liquid material are also possible. The following metrics can be used during tank transport task: length of the route, transported value, fuel consumption, vehicle rental fee, route status, route time, packaging time, unpacking time, administrative time, fuel filling time, waiting time of the nodes, exceeding time window, a penalty point for missed customers.

The test run included the following parameters: vehicles (tanker). The result was the following product list: tanker liquid, bulk product. The result also contained the following component list: route status, administrative cost, vehicle rental fee, fuel filling time, time window, administrative time, reliability between nodes, fuel consumption, delivery, inter-depot route, open route etc. It also had the following metrics: route status, exceeding the number of suppliers, loading time, quality control time, administrative cost, capital time, loading cost, length of the route, transported value, quality control cost, and vehicle rental fee.

The validation of the test result was the following: The program result was correct, there were no other classes that the query should have included.

5.3. Transport of Parcels

The transportation of parcels is also a completely different case. The number of levels can be important here, especially for foreign parcels. From the post office, the parcels are sent to a central sorter, then to other sorters, then finally to a post office close to the recipient, and finally from there to the recipient. A multi-level satellite system was considered in this case.

Among the attributes between nodes, the followings are important: travel time between the locations, travel distance between locations, and route status between the locations.

Different types of vehicles (land, water, air) can also be used. It also depends on the levels, and what kind of vehicles is used. Attributes for vehicles are also important; here too we need to consider several attributes. Several types of vehicles can be used, so these attributes are very important: capacity constraint of each vehicle, fuel consumption of vehicles and recharger time can also be calculated because electric vehicles can also transport mail items. Whether the vehicle is owned or borrowed can also be an important factor. The rental fee per vehicle type is also an important factor when using rented vehicles. Fuel consumption of vehicles can also be a useful factor.

The packing time of the locations is low; letters are only bagged, just as the unpacking time of the locations is low. However, there is a loading time for the locations, as well as an unloading time of the locations. The fuel filling time of the vehicles can be considered. There is also a time window; each sending post office has opening hours, which can vary depending on the size of the post office. Each sorting location may also have opening hours.

A multiple-time window can also be used; each post can close e.g., due to a lunch break in the village post offices.

In terms of the functional parameter, the soft time window can also be used, e.g., the post office receives letters and parcels even after it is open, but then the postmen must pay overtime, so the objective function gets a penalty point. The open route can also be used, especially in the case of transport by train, boat, or plane, and in the case of road transport, round-trip transport is common.

Among the metrics, the minimization of the route is an important factor. Profit maximization is not usually used, so minimizing sold values, minimizing packaging costs, minimizing unpacking costs, minimizing loading costs, minimizing unloading costs, minimizing administrative costs, and minimizing quality control costs do not make sense. Minimization of fuel consumption and minimization of the vehicle rental fee can be considered. Maximization of road safety is not usually considered but maximizing route status can be considered. From the time minimization factor, the minimization of route time can be considered, but the minimization of packaging time and minimization of unpacking time factors does not make sense. minimization of loading time can be useful, but minimization of capital time has no sense. Minimization of administrative time and minimization of quality control time is not considered. Of the penalty points, exceeding the number of suppliers and unvisited customers can be important, because it is important that all customers are served by a given vehicle.

The test run included the following parameters: products (postal package). The result was the following vehicle list: truck, train, vehicular vehicle, and airplane.

The validation of the test run was the following: the test result was correct; these classes contain many subclasses covering a wide range of vehicle types.

5.4. Money Transportation

The transportation of money is a special case of vehicle routing; it may be different from the previous cases. Here, too, the tiered system (depot-satellite-customer) can be applied. The customers are the bank, post office, gas station, shop, etc.

Among the relationships between nodes, the travel time between the locations, the travel distance between the locations, the reliability between the locations, and possibly the status of the route, are important. Reliability between the locations is a very important factor, as it is not advisable for money to be robbed.

Among the attributes of the node, the type of the node (depot, satellite, and customer) is an important factor; no other factor plays a role.

Among the attributes of vehicles, is the capacity constraint, which can be important, the vehicles can only transport a certain amount of money. Fuel consumption can also be important. Electric vehicles may not carry the money, so the recharger time does not matter. It may also be important whether the vehicle is owned or borrowed and the rental fee per vehicle type. Fuel consumption of vehicles is also a significant attribute.

The attributes of the temporalities can be the packing time of the locations and the unpacking time of the locations if the distribution is made from intermediate locations. The time window plays an important role due to the opening hours of certain places (e.g., post offices and shops have opening hours).

Among the attributes of products, the capacity constraint of the locations can be important, as well as the product demand of the locations, because it can be given how much money is needed in each position. The number of products (money) in the warehouse (storage level of the locations) can be an important factor as possibly the type of products (money).

Among the functional parameters, the inter-depot route, i.e., the movement between repositories (upper levels), can be used. Furthermore, here is the delivery, but it can also be a pickup, we collect the old worn-out money for destruction, or the excess money is also handed over for storage security. A soft time window can also be used.

Among the metrics, the minimization of the route is an important metric. Some of the factors of maximization of the profit may also be worth considering, such as minimization packaging costs and minimization unpacking costs, minimization unloading costs and minimization unloading costs, minimization administrative costs, minimization quality control costs, minimization fuel consumption, minimization rental fee which may consist of minimization vehicle rental fee. Maximization of road safety is a very important factor, but so is the maximization of route status. Minimization of route time, minimization of packaging time and unpacking time, minimization of loading time, minimization of administrative time, minimization of quality control time and minimization of waiting time for nodes can be important metrics. Penalty points can include exceeding the number of suppliers and unvisited customers.

The first test run included the following parameters: products (money). The result was the following vehicle list: money transporter. The validation of the following: the test run was correct; the money needs to be transported by a special vehicle

The second test run included the following parameters: vehicle (money transporter). The result was the following product list: money. It contained the following component list: route status, administrative cost, vehicle rental fee, fuel filling time, time window, administrative time, reliability between nodes, fuel consumption, delivery, inter-depot route, and open route. It also contained the following metrics: route status, administrative cost, vehicle rental fee, fuel filling time, exceeding time window, administrative time, reliability between nodes, and fuel consumption. The validation of the second test run was the following: From the product point of view, only money should be transported with the money transporter, so this was correct. The components and metrics were correct, but the following could be added to the result: maximum distance with full tank (in connection with vehicle), recharger time (in connection with vehicle).

The third test result included the following parameters: node (bank). The result was the following product list: money. It had the following component list: route status, administrative cost, vehicle rental fee, fuel filling time, time window, administrative time, reliability between nodes, fuel consumption, delivery, inter-depot route, and open route. It had the following metrics: route status, administrative cost, vehicle rental fee, fuel filling time, exceeding time window, administrative time, reliability between nodes, and fuel consumption. The validation of the third test run was the following: From the product point of view, only money should be transported so this was correct. The components and metrics were correct, but the following could be added to the result: maximum distance with full tank (in connection with vehicle), recharger time (in connection with vehicle).

5.5. Improvement of Our Ontology System

In this subchapter, we present possible improvements for the presented ontology system, the improvement of which is part of our plan. Our system can be improved with the following way:

- Expansion of our ontology with more special cases.
- Expansion of the query set (now the system gives vehicles, we do not receive the component and metrics section, nor the nodes section).
- Building a more accurate validation option into the system. In the system queries and responses are limited to for a small range. The range should be expanded.

6. Conclusions

The Vehicle Routing Problem is a highly investigated problem domain, the first related paper was published in 1959 by Dantzig and Ramster, and we can see nowadays in the literature very complex VRP systems containing several components. Over the years, different authors presented only single problem-specific models, so there may be a need to design a general transportation system. In this paper, we have presented a general model that is suitable for the design of VRP components for out-plant material handling tasks, which products can be delivered to which nodes by which vehicles and which products can be delivered together. The paper presents the development of a unified description of the ontology-based formalism to describe a wide spectrum of VRPs. The system is implemented in OWL, and the manipulations can be made in standard ontology API, Protége editor for IT expert users, and with a JSON descriptor for logistics expert users. The paper presented a mapping method to convert the VRP model into ontology format. Our system contains the following main components: 'Node', 'Product', 'Vehicle', 'Component', and 'Metrics'. The 'Component' class means general VRP components, for example, components can be the travel time between the nodes, the capacity constraint of the vehicles, inter-depot route, pick-up, delivery etc. 'Metrics' here denotes the objective function components, such as the minimization of the length of the route, minimization of the packing time etc. We also have proposed a user interface for the system, with the help of which even users who are not ontology experts can easily make queries to the system. After the analysis of our ontology system, we presented logistics applications. We presented, that the ontology can be built into the logic of the software applications related to logistic problems. The paper includes the following case studies: transport of short-term food, transport of refrigerated products, tank transport, transport of parcels, and money transportation. The developed ontology system is not only for the expert user (who knows how to create ontologies and can formulate queries with ontology query languages such as SPARQL) but also for any end-user (without special ontology knowledge) who can use simple queries to formulate to get the required information.

Author Contributions: Conceptualization, A.A. and L.K.; data curation, A.A.; formal analysis, A.A. and L.K.; methodology, A.A.; software, A.A.; supervision, L.K. and T.B.; visualization, A.A.; writing—original draft, A.A., L.K. and T.B.; writing—review and editing, A.A., L.K. and T.B. All authors have read and agreed to the published version of the manuscript.

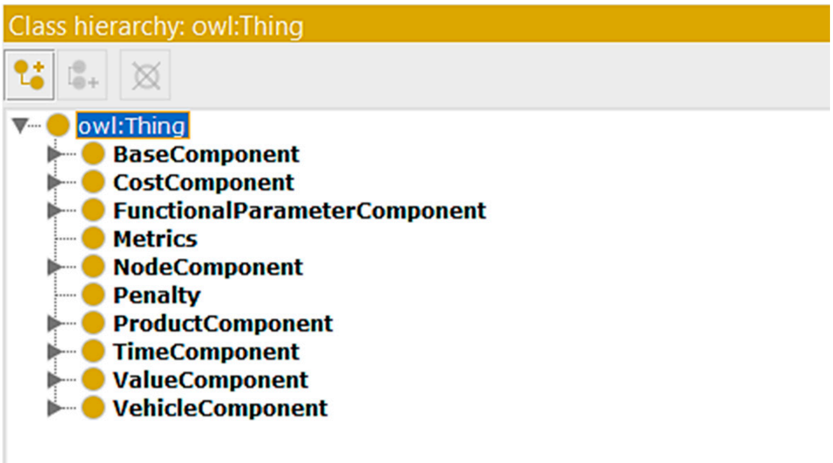
Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

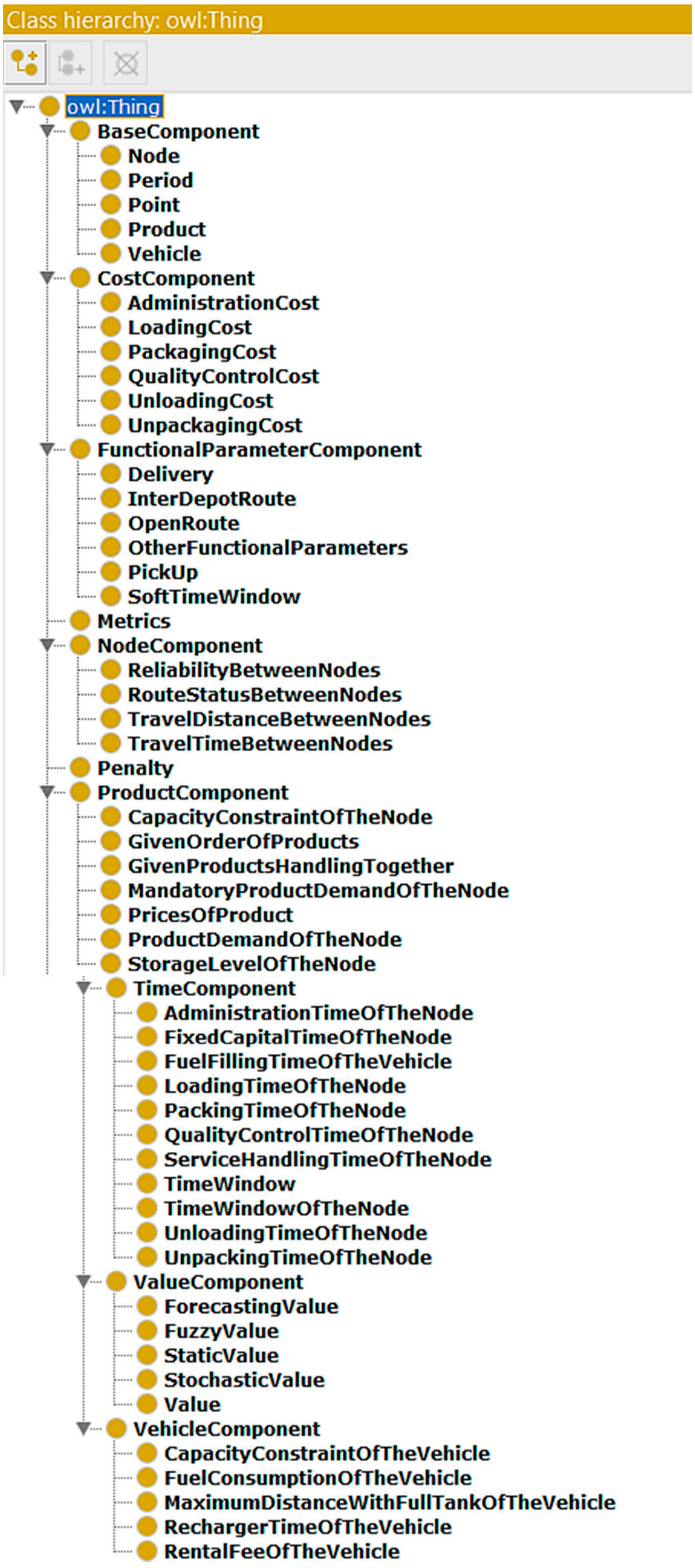
Appendix A



Class hierarchy: owl:Thing

- owl:Thing
  - BaseComponent
  - CostComponent
  - FunctionalParameterComponent
  - Metrics
  - NodeComponent
  - Penalty
  - ProductComponent
  - TimeComponent
  - ValueComponent
  - VehicleComponent

Figure A1. The ontology class hierarchy.



Class hierarchy: owl:Thing

- owl:Thing
  - BaseComponent
    - Node
    - Period
    - Point
    - Product
    - Vehicle
  - CostComponent
    - AdministrationCost
    - LoadingCost
    - PackagingCost
    - QualityControlCost
    - UnloadingCost
    - UnpackagingCost
  - FunctionalParameterComponent
    - Delivery
    - InterDepotRoute
    - OpenRoute
    - OtherFunctionalParameters
    - PickUp
    - SoftTimeWindow
  - Metrics
  - NodeComponent
    - ReliabilityBetweenNodes
    - RouteStatusBetweenNodes
    - TravelDistanceBetweenNodes
    - TravelTimeBetweenNodes
  - Penalty
  - ProductComponent
    - CapacityConstraintOfTheNode
    - GivenOrderOfProducts
    - GivenProductsHandlingTogether
    - MandatoryProductDemandOfTheNode
    - PricesOfProduct
    - ProductDemandOfTheNode
    - StorageLevelOfTheNode
    - TimeComponent
      - AdministrationTimeOfTheNode
      - FixedCapitalTimeOfTheNode
      - FuelFillingTimeOfTheVehicle
      - LoadingTimeOfTheNode
      - PackingTimeOfTheNode
      - QualityControlTimeOfTheNode
      - ServiceHandlingTimeOfTheNode
      - TimeWindow
      - TimeWindowOfTheNode
      - UnloadingTimeOfTheNode
      - UnpackingTimeOfTheNode
    - ValueComponent
      - ForecastingValue
      - FuzzyValue
      - StaticValue
      - StochasticValue
      - Value
    - VehicleComponent
      - CapacityConstraintOfTheVehicle
      - FuelConsumptionOfTheVehicle
      - MaximumDistanceWithFullTankOfTheVehicle
      - RechargerTimeOfTheVehicle
      - RentalFeeOfTheVehicle

Figure A2. The ontology class hierarchy.

References

1. Dantzig, G.B.; Ramser, J.H. The truck dispatching problem. *Manag. Sci.* **1959**, *6*, 80–91. [CrossRef]
2. Lysgaard, J.; Letchford, A.N.; Eglese, R.W. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Math. Program.* **2004**, *100*, 423–445. [CrossRef]
3. Vincent, F.Y.; Jewpanya, P.; Redi, A.P. Open vehicle routing problem with cross-docking. *Comput. Ind. Eng.* **2016**, *94*, 6–17.
4. Ribeiro, G.M.; Laporte, G. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Comput. Oper. Res.* **2012**, *39*, 728–735. [CrossRef]
5. Montoya, A.; Guéret, C.; Mendoza, J.E.; Villegas, J.G. The electric vehicle routing problem with nonlinear charging function. *Transp. Res. Part B Methodol.* **2017**, *103*, 87–110. [CrossRef]
6. Soysal, M.; Bloemhof-Ruwaard, J.M.; Bektaş, T. The time-dependent two-echelon capacitated vehicle routing problem with environmental considerations. *Int. J. Prod. Econ.* **2015**, *164*, 366–378. [CrossRef]
7. Poonthalir, G.; Nadarajan, R. A fuel efficient green vehicle routing problem with varying speed constraint (F-GVRP). *Expert Syst. Appl.* **2018**, *100*, 131–144. [CrossRef]
8. Gupta, R.; Singh, B.; Pandey, D. Fuzzy vehicle routing problem with uncertainty in service time. *Int. J. Contemp. Math. Sci.* **2010**, *5*, 497–507.
9. Gendreau, M.; Laporte, G.; Musaraganyi, C.; Taillard, É.D. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Comput. Oper. Res.* **1999**, *26*, 1153–1173. [CrossRef]
10. Dondo, R.; Cerdá, J. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *Eur. J. Oper. Res.* **2007**, *176*, 1478–1507. [CrossRef]
11. Setak, M.; Jalili Bolhassani, S.; Karimi, H.; Ghorbani, B. Capacitated Multi-depot Vehicle Routing Problem with Inter-depot Routes. *Adv. Ind. Eng.* **2014**, *48*, 11–18.
12. Nagy, G.; Salhi, S. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *Eur. J. Oper. Res.* **2005**, *162*, 126–141. [CrossRef]
13. Kabcome, P.; Mouktonglang, T. Vehicle routing problem for multiple product types, compartments, and trips with soft time windows. *Int. J. Math. Math. Sci.* **2015**, *2015*, 126754. [CrossRef]
14. Belhaiza, S.; Hansen, P.; Laporte, G. A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. *Comput. Oper. Res.* **2014**, *52*, 269–281. [CrossRef]
15. Archetti, C.; Savelsbergh, M.; Speranza, M.G. The vehicle routing problem with occasional drivers. *Eur. J. Oper. Res.* **2016**, *254*, 472–480. [CrossRef]
16. Li, F.; Golden, B.; Wasil, E. The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. *Comput. Oper. Res.* **2007**, *34*, 2918–2930. [CrossRef]
17. Angelelli, E.; Speranza, M.G. The periodic vehicle routing problem with intermediate facilities. *Eur. J. Oper. Res.* **2002**, *137*, 233–247. [CrossRef]
18. Hsu, C.I.; Hung, S.F.; Li, H.C. Vehicle routing problem with time-windows for perishable food delivery. *J. Food Eng.* **2007**, *80*, 465–475. [CrossRef]
19. Lin, C.K.Y. A vehicle routing problem with pickup and delivery time windows, and coordination of transportable resources. *Comput. Oper. Res.* **2011**, *38*, 1596–1609. [CrossRef]
20. Talarico, L.; Springael, J.; Sörensen, K.; Talarico, F. A large neighbourhood metaheuristic for the risk-constrained cash-in-transit vehicle routing problem. *Comput. Oper. Res.* **2017**, *78*, 547–556. [CrossRef]
21. Allahviranloo, M.; Chow, J.Y.; Recker, W.W. Selective vehicle routing problems under uncertainty without recourse. *Transp. Res. Part E Logist. Transp. Rev.* **2014**, *62*, 68–88. [CrossRef]
22. Skrlec, D.; Filipec, M.; Krajcar, S. A heuristic modification of genetic algorithm used for solving the single depot capacited vehicle routing problem. In Proceedings of the Intelligent Information Systems, IIS'97, Grand Bahama Island, Bahamas, 8–10 December 1997; pp. 184–188.
23. Figliozzi, M.A. An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. *Transp. Res. Part C Emerg. Technol.* **2010**, *18*, 668–679. [CrossRef]
24. Stewart, W.R., Jr.; Golden, B.L. Stochastic vehicle routing: A comprehensive approach. *Eur. J. Oper. Res.* **1983**, *14*, 371–385. [CrossRef]
25. Desrochers, M.; Desrosiers, J.; Solomon, M. A new optimization algorithm for the vehicle routing problem with time windows. *Oper. Res.* **1992**, *40*, 342–354. [CrossRef]
26. Xiao, Y.; Konak, A. The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion. *Transp. Res. Part E Logist. Transp. Rev.* **2016**, *88*, 146–166. [CrossRef]
27. Černý, V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *J. Optim. Theory Appl.* **1985**, *45*, 41–51. [CrossRef]
28. Perboli, G.; Tadei, R. New families of valid inequalities for the two-echelon vehicle routing problem. *Electron. Notes Discret. Math.* **2010**, *36*, 639–646. [CrossRef]
29. El Sayed, M.A.; Baky, I.A.; Singh, P. A modified TOPSIS approach for solving stochastic fuzzy multi-level multi-objective fractional decision making problem. *Opsearch* **2020**, *57*, 1374–1403. [CrossRef]

30. El Sayed, M.A.; Farahat, F.A.; Elsisy, M.A. A novel interactive approach for solving uncertain bi-level multi-objective supply chain model. *Comput. Ind. Eng.* **2022**, *169*, 108225. [CrossRef]
31. Elsisy, M.A.; Elsaadany, A.S.; El Sayed, M.A. Using interval operations in the Hungarian method to solve the fuzzy assignment problem and its application in the rehabilitation problem of valuable buildings in Egypt. *Complexity* **2020**, *2020*, 9207650. [CrossRef]
32. Elsisy, M.A.; El Sayed, M.A.; Abo-Elnaga, Y. A novel algorithm for generating Pareto frontier of bi-level multi-objective rough nonlinear programming problem. *Ain Shams Eng. J.* **2021**, *12*, 2125–2133. [CrossRef]
33. El Sayed, M.A.; Abo-Sinna, M.A. A novel approach for fully intuitionistic fuzzy multi-objective fractional transportation problem. *Alex. Eng. J.* **2021**, *60*, 1447–1463. [CrossRef]
34. Horridge, M.; Jupp, S.; Moulton, G.; Rector, A.; Stevens, R.; Wroe, C.; Jupp, S.; Moulton, G.; Drummond, N.; Brandt, S. *A Practical Guide to Building OWL Ontologies Using Protege 4 and CO-ODE Tools, 1.3 ed.*; The University of Manchester: Manchester, UK, 2011.
35. OWL Overview. Available online: http://www.w3c.hu/forditasok/OWL/REC-owl-features-20040210.html (accessed on 1 September 2022).
36. OWL Use Cases and Requirements. Available online: http://www.w3c.hu/forditasok/OWL/REC-webont-req-20040210.html (accessed on 1 September 2022).
37. Protége. Available online: https://protege.stanford.edu/ (accessed on 1 September 2022).
38. OWLViz. Available online: https://github.com/protegeproject/owlviz (accessed on 1 September 2022).
39. VOWL. Available online: http://vowl.visualdataweb.org/ (accessed on 1 September 2022).
40. OntoGraf. Available online: https://protegewiki.stanford.edu/wiki/OntoGraf (accessed on 1 September 2022).
41. Smith, S.F.; Becker, M.A. An ontology for constructing scheduling systems. In Proceedings of the Working Notes of 1997 AAAI Symposium on Ontological Engineering, Stanford, CA, USA, 24–25 March 1997; AAAI Press: Washington, DC, USA, 1997; pp. 120–127.
42. Himoff, J.; Rzevski, G.; Skobelev, P. Magenta technology multi-agent logistics i-Scheduler for road transportation. In Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, Hakodate, Japan, 8–12 May 2006; pp. 1514–1521.
43. Lian, P.; Park, D.W.; Kwon, H.C. Design of logistics ontology for semantic representing of situation in logistics. In Proceedings of the Second Workshop on Digital Media and its Application in Museum & Heritages (DMAMH 2007), Chongqing, China, 10–12 December 2007; IEEE: Manhattan, NY, USA, 2007; pp. 432–437.
44. Dong, H.; Hussain, F.K.; Chang, E. Transport service ontology and its application in the field of semantic search. In Proceedings of the 2008 IEEE International Conference on Service Operations and Logistics, and Informatics, Beijing, China, 12–15 October 2008; IEEE: Manhattan, NY, USA, 2008; Volume 1, pp. 820–824.
45. Hoxha, J.; Scheuermann, A.; Bloehdorn, S. An approach to formal and semantic representation of logistics services. In Proceedings of the Workshop on Artificial Intelligence and Logistics (AILog), 19th European Conference on Artificial Intelligence (ECAI 2010), Lisbon, Portugal, 16–20 August 2010; pp. 73–78.
46. Xu, D.; Wijesooriya, C.; Wang, Y.G.; Beydoun, G. Outbound logistics exception monitoring: A multi-perspective ontologies' approach with intelligent agents. *Expert Syst. Appl.* **2011**, *38*, 13604–13611. [CrossRef]
47. Anand, N.; Yang, M.; van Duin, J.R.; Tavasszy, L. GenCLOn: An ontology for city logistics. *Expert Syst. Appl.* **2012**, *39*, 11944–11960. [CrossRef]
48. De Oliveira, K.M.; Bacha, F.; Mnasser, H.; Abed, M. Transportation ontology definition and application for the content personalization of user interfaces. *Expert Syst. Appl.* **2013**, *40*, 3145–3159. [CrossRef]
49. Daniele, L.; Pires, L.F. An ontological approach to logistics. In *Enterprise Interoperability, Research and Applications in the Service-Oriented Ecosystem, IWEI 2013*; John Wiley & Sons: Hoboken, NJ, USA, 2014; Volume 13, pp. 199–213.
50. Zhang, L.; Jiang, D.; Zeng, Y.; Ning, Y.; Wang, Q. Exploring Ontology-driven Modeling Approach for Multi-agent Cooperation in Emergency Logistics. *JCP* **2014**, *9*, 285–294. [CrossRef]
51. Fumagalli, L.; Pala, S.; Garetti, M.; Negri, E. Ontology-based modeling of manufacturing and logistics systems for a new MES architecture. In *IFIP International Conference on Advances in Production Management Systems*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 192–200.
52. Wang, Y.; Yi, J.; Zhu, X.; Luo, J.; Ji, B. Developing an ontology-based cold chain logistics monitoring and decision system. *J. Sens.* **2015**, *2015*, 231706. [CrossRef]
53. Glöckner, M.; Ludwig, A. LoSe ODP-an ontology design pattern for logistics services. *Adv. Ontol. Des. Patterns* **2017**, *32*, 131.
54. Agárdi, A.; Kovács, L.; Bányai, T. Mathematical Model for the Generalized VRP Model. *Sustainability* **2022**, *14*, 11639. [CrossRef]
55. Li, S.T.; Hsieh, H.C.; Sun, I.W. An Ontology-based Knowledge Management System for the Metal Industry. *WWW (Alternate Paper Tracks)*. 2003. Available online: http://www.ra.ethz.ch/cdstore/www2003/papers/alternate/P620/P620-LI.PDF (accessed on 1 September 2022).