



## Modelling of Agent-Based Vehicle Routing Problem Using Unified Modelling Language

Nour Abdullatif<sup>1\*</sup>, Sally Kassem<sup>1,2</sup>

<sup>1</sup> Smart Engineering Systems Center, Industrial Engineering, Nile University, Giza 12613, Egypt

<sup>2</sup> Faculty of Computers and Artificial Intelligence, Cairo University, Giza 12588, Egypt

Corresponding Author Email: [n.sameh@nu.edu.eg](mailto:n.sameh@nu.edu.eg)

<https://doi.org/10.18280/jesa.530604>

### ABSTRACT

**Received:** 8 June 2020

**Accepted:** 26 November 2020

**Keywords:**

*agent-based modelling, UML modelling, VRP*

The Vehicle Routing Problem (VRP) is among the most studied optimization problems in the field of supply chain management. Typically, VRP requires dispatching a fleet of vehicles from a central depot to deliver demand to pre-determined spatially dispersed customers, with the objective of minimizing the total routing cost, and the constraint of not exceeding vehicles' capacities. Agent Based Modelling (ABM) assists industries in the use of technology to support their decision-making process. This paper proposes a model of an Agent Based Vehicle Routing Problem System. The system under study is modelled using the Unified Modelling Language 2.0 (UML 2.0). The aim of the proposed model is to exploit the clear visualization provided by UML and the detailed view of the Agent-based modelling, in order to propose a new modeling perspective for the classic VRP. The paper covers the System initiation phase, in addition to, the functional, behavioral, and structural models.

## 1. INTRODUCTION

The Unified Modeling Language (UML) has a remarkable impact on how software systems are constructed since its standardization in 1997 [1, 2]. The role of modeling in specifying and documenting complicated software systems is currently recognized, and software engineering's industrial strategy is on its course to becoming reality. With the adoption of UML, a fresh generation of methods have also appeared. Those methods use UML to illustrate the system's architecture and requirements engineering. UML is being used to aid various industries in modelling. For instance, returned goods has commonly been seen for years as a burden for both vendors and customers. Many organizations and companies recognize reverse logistics; the motion of products back up the supply chain. Major environmental concerns have been the motive to launch the reverse logistics domain. Moreover, companies have found that reverse logistics plays a major role in attaining "green supply chains" [3]. Building a practical reverse logistics management system is expected to realize high efficiency reverse logistics operations. In addition, it can also create a modern alternative for information management. UML has been helpful in developing the information system for reverse logistics, for example, UML modeling of capacity for life-cycle of products [4, 5].

The classical Vehicle Routing Problem (VRP) is the problem of determining several routes for a fleet of vehicles that are placed at one or several depots. The vehicles are required to serve pre-determined customers that are geographically dispersed. The objective of the VRP is to deliver to all customers with minimum cost. Dantzig et al. [6] proposed the first mathematical programming formulation and algorithmic approach to represent and solve the problem. They also described VRP with a real-world application concerning the delivery of gasoline to service stations. Clarke and Wright

proposed an effective greedy heuristic that improved on the Dantzig-Ramser approach [7]. After these two papers, many models and algorithms are proposed for the optimal and approximate solutions of different variants of the VRP [8-12].

VRP has several applications in various industries. VRP is used to aid the ambulance service, disaster relief, and the blood program to efficiently respond in the shortest time to critical situations [13, 14]. A survey on the applications of the problem on the different elements of the "transport chain" within and outside the port is introduced [15]. Moreover, the paper depicts the increasing need to a heightened efficiency at the different portals. This is to be done while taking into consideration the complexity of managing a terminal's logistics operations. Terminal's interactions with other terminals and ports are noted as well.

Agent-based Modeling (ABM) is an important technique of modeling systems composed of independent communicating agents [16]. An increasing number of Agent-based applications in diverse areas have been made possible through computer developments. Applications greatly vary. For instance, ABM is used in modelling stock market mapping [16, 17] and supply chains [18-20]. In addition, anticipating the spread of disease outbreaks [21] and the danger of using biological beings as weapons in wars [22]. Agent-Based concepts are also typically implemented in microscopic modeling of systems where common activities are represented by independent decision-makers, mostly humans [23]. An Agent-based model for sustainable logistic ideas in courier services is provided by Meyer [24].

Agent-based VRP has been addressed by Komenda et al. [25]. The authors illustrate Contract-Net Protocol (CNP) based approach. For every task, the best Vehicle Agent is selected according to insertion estimation satisfying capacity constraints. This strategy contains no backtracking and in case of allocation failure, due to capacity constraint of Vehicle

Agents, the whole process restarts with a higher number of Vehicles. Moreover, capacity backtracking strategy is based on the CNP strategy, but with backtracking in case of allocation failure. In case no Vehicle Agent can undertake a new task because of the capacity constraint, the best Vehicle Agent is selected regardless of capacity limitations.

Several research articles discuss UML Modelling within the context of supply chain management. Other research articles address Agent-Based VRP. However, to the best of our knowledge, no research has been done to investigate modeling VRP using UML. Furthermore, no research has been done on modelling Agent-based VRP using UML. Due to the importance of Agent-based VRP and UML visualization capabilities, this work presents UML diagrams to model Agent-based VRP. The aim of the paper is to aid in the visualization of VRP from an Agent-based perspective, instead of the optimization of the solution. Hence, the modeling approach adopted to represent the problem defined in this work is the UML approach, and not the classical mathematical programming modelling approach.

The rest of the paper is organized as follows; Section 2 provides problem statement, Section 3 illustrates the different UML models, while Section 4 concludes the paper.

## 2. PROBLEM STATEMENT

The vehicle routing problem modeled in this work is defined as the problem of finding the minimum distance to travel using a fleet of identical vehicles to satisfy demands of a group of customers. The fleet of vehicles is located at a central depot, where customers' demands are available. A vehicle starts its trip from the central depot, visits its assigned customers for delivery of demands, and end the trip back at the depot. Each vehicle has a limited capacity that cannot be violated. The objective is to minimize the total distance travelled by all vehicles while serving all customers.

The VRP falls into the class of NP-hard problems as mentioned by Kassem et al. [9]. As a result, it is challenging to find a solution for practical size VRP instances in reasonable time.

The Agent-based VRP concept relies on dividing the problem into sections and having an agent responsible of each section. In this paper, there are four sections to the problem. The "demand points location and demand collection" section, where an agent called "Demand Points Agent" is responsible for. The "sortation and evaluation of different tasks", i.e., deliveries, and this is the responsibility of the "Task Agent". The "allocation of different demand points to clusters, or partitions" is another section and the "Allocation Agent" is responsible for. Finally, the "Vehicle Agent" is responsible for the section "determining the optimum route". The paper does not rely on a mathematical representation of the problem. This is because the objective of this work is to model VRP from a different perspective that relies on visualization. The aim is to aid in software implementation rather than the typical route optimization associated with VRP.

Figure 1 shows an Integration Definition0 (IDEF0) to illustrate the problem and depict a clearer view of the purpose of the model. The inputs are the quantity of Demand Points locations and their corresponding Demand and the minimum Number of Vehicles that can be utilized. On the other hand, the Dynamic Route is the output. The route is considered dynamic because it might change according to the route

congestion. The controls that direct the activities in the process are the maximum vehicles' capacity, Tasks Sorting Strategies, and Allocations Strategies. The Vehicles' capacity is determined by the vehicles, while the Tasks Sorting Strategies, and Allocations Strategies are determined according to the depot's strategy, which is one of the following: Most Demand First (MDF), Least Demand First (LDF), or First in First Out (FIFO). Finally, the mechanisms required to complete the process are Equipment and Technology.

The system begins with the Demand Points Agent determining customers' locations and their corresponding demands. The Task Agent (TA) sorts demand points according to the depot's strategy, namely, Most Demand First (MDF), Least Demand First (LDF), or First in First Out (FIFO). Then, the Allocation Agent (AA) clusters the demand points into independent clusters according to depot's strategy, as well, such that each cluster is served with one vehicle. The Allocation Agent clusters orders through two phases. In the first phase, a feasible solution is obtained using one of two strategies, which are Contract-Net Protocol (CNP) or Capacity Backtracking (CB). Then, the AA starts improving the feasible solution using one of three strategies. Delegate Worst (DW), Delegate All (DA), or Reallocate All (RA), the Vehicle Agent determines the optimal dynamic routes, or the sequence of visiting customers per cluster with the associated vehicle. After vehicles are dispatched from the depot, feedback from drivers may require routes change to avoid traffic congestions. The Task Agent determines the orders that will be served (delivered) immediately, using the strategy Iterative Processing (ITER), and the other orders that will wait until consolidated with other orders, later, using the Batch Processing (NORM) strategy. This decision is taken by the source depot according to the customer's request, such that, immediate service implies additional cost to customers. If the order has the privilege of immediate shipping, then the depot serves the corresponding demand point immediately, otherwise, the order is collected and sorted according to the depot's strategy through the Task Agent. Then, the order is sent to an Allocation Agent (AA) that will cluster orders. Finally, the Vehicle Agent (VA) will output the optimal route per vehicle. More details about agents and associated roles are available in the paper [25].

Figure 2 illustrates a context diagram to identify and explain the system boundaries, sometimes called a level 0 data-flow diagram. The diagram defines data flows between the system and outside entities. The figure shows a shipping company that operates a fleet of vehicles. The shipping company sends data of the available capacities of vehicles to the Agent Based VRP System (ABVRPS). The ABVRPS has information regarding the amount of demand to be delivered. Accordingly, the ABVRPS determines the minimum number of vehicles required to satisfy the available demand. A Geographic Information System, referred to as Maps, receives demand point locations and their corresponding demand amounts. Accordingly, Maps provides ABVRPS with real time geographic data about possible routes to reach demand points; this includes distances, traffic congestion conditions, and alternative routes. Project Manager is the end user of ABVRPS. The project Manager is provided with a summary report of the input of concern and the system output. Input included in the report are demand point locations with the corresponding demand. Output included in the report are the optimal routes for vehicles and the total distance travelled. Reports generated for the Project Manager might be in the form of tables or

illustrative figures where demand points are shown on a map along with the routes traversed. These reports have managerial

implications in regards with control and improvement.

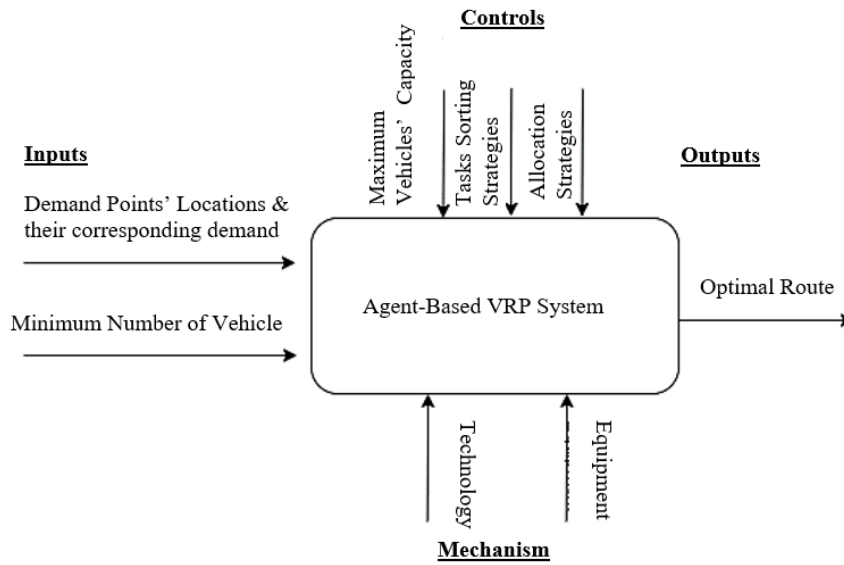


Figure 1. IDEF0 diagram

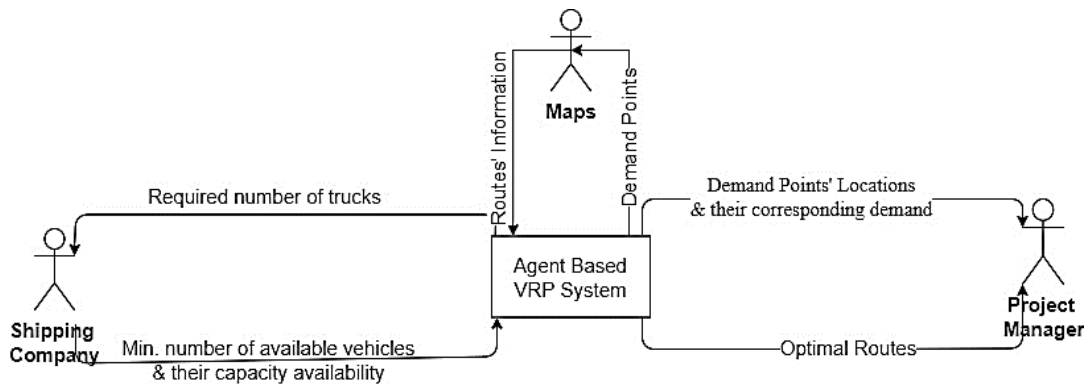


Figure 2. Context diagram

### 3. UML MODELS

#### 3.1 Functional modelling

##### 3.1.1 Use case model

A use case shows the interactions between users and the Agent-based VRP system and describes functionalities of a VRP system. A use case diagram is illustrated in Figure 3. The diagram consists of a set of use cases, actors, and relationships between actors and use cases.

The purpose of use case diagram is to show a context of an Agent-based VRP (ABVRP) system. There are three types of Agents: Task Agent (TA), Allocation Agent (AA), and Vehicle Agent (VA). All three agents have an inheritance relationship with the actor Agents. Demand Points actor represents the customers with their location and demanded quantity. The actors represent the external systems related with ABVRP.

Use Cases are functions in the system. For example, Demand Points use the Report an Order use case to place an order. The use case includes a use case called Collect Demand which is responsible for adding the order to a list of orders for processing. Collect Demand has two use cases that extends it: Order Tasks and Process Demand. Order Tasks use case is

triggered when the system is set to order tasks, or orders, according to a certain priority rule. Otherwise, Process Demand use case is triggered.

Identifying business actors is a crucial step to properly define the system and build the use case model. Table 1 summarizes and documents the actors in the system. Some Actors have a synonym that illustrates what that actor represents in the proposed model. Other actors do not have a synonym as they do not represent other entities in the system and are part of the novel proposed system.

In the use case diagram, relationships between use cases are represented by using stereotypical <<include>> and <<extend>>. For example, Most Demand First (MDF), Least Demand First (LDF), and First in First Out (FIFO) use cases and Collect Demand use cases share the functionality of the Process Demand use case. After an order is requested, Collect Demand use case can be extended to the Order Tasks use case as the Demand Point may ask for tasks ordering or not. All use cases of the system are documented in Table 2.

The activity diagram in Figure 4 illustrates the sequence of activities and the participating actors. The system is initiated when a demand point sends a notification with its demand to a Task Agent (TA). The TA receives orders from different demand points, then checks the orders to be consolidated, and

the orders that to be served immediately. This is decided by the owner of the ABVRP system who is the source depot. In case of needing to batch for consolidation, the order is collected and sorted according to the depot strategy. Then the order is sent to an Allocation Agent (AA). The AA will start allocating demand points to partitions in a cost-efficient manner. The Vehicle Agent (VA) will then conduct a plan based on the orders sorted by the TA and the partition made by the VA.

The processing and ordering of orders are done by the Task Agent. There are two processing strategies. First, to batch the orders together to consolidate the demand. The orders can be processed once received by the Task Agent as well. In case of batching the orders, a priority rule can be set for their sequence of processing. They could be processed by giving a priority to the orders with least demanded quantity, most demand, or first order in will be processed first.

The Allocation Agent (AA) clusters the orders into partitions. There are two strategies for clustering: Contract-Net Protocol (CNP) and Capacity Backtracking (CB). Contract-Net Protocol strategy assigns orders to vehicles based on the insertion estimation in accordance with the vehicles' maximum capacity. However, if the vehicles' capacity constraint hindered the allocation process, additional

vehicle would be added to increase the total vehicles' capacity. The process of CNP is then re-initialized.

Capacity Backtracking allows re-allocation of orders before adding additional vehicles. The AA will keep on re-allocating orders in different vehicles aiming to fit all orders with the initial number of vehicles. CB is stopped after a predefined maximum number of iterations. The AA starts the improvement stage based on one of three strategies: Delegating the worst tasks to different partition, Delegating All tasks in a partition to other partitions, or re-allocating all partitions. The decision of delegation or re-allocation is made if the savings are higher than the insertion costs.

### 3.1.2 Structural modeling

Class diagrams describe a static view of any system in terms of classes and relationships among them [26]. Figure 5 shows a class diagram of the system under study.

The model consists of two abstract classes: Agents and Vehicle. In Figure 5, Task Agent, Allocation Agent, and Vehicle Agent are classes that inherit from Agents Class. While only Vehicle Agent inherits from Vehicles as well. There are other concrete classes like Demand Points, Tasks, and Partitions. A Partition is composed of several Tasks.

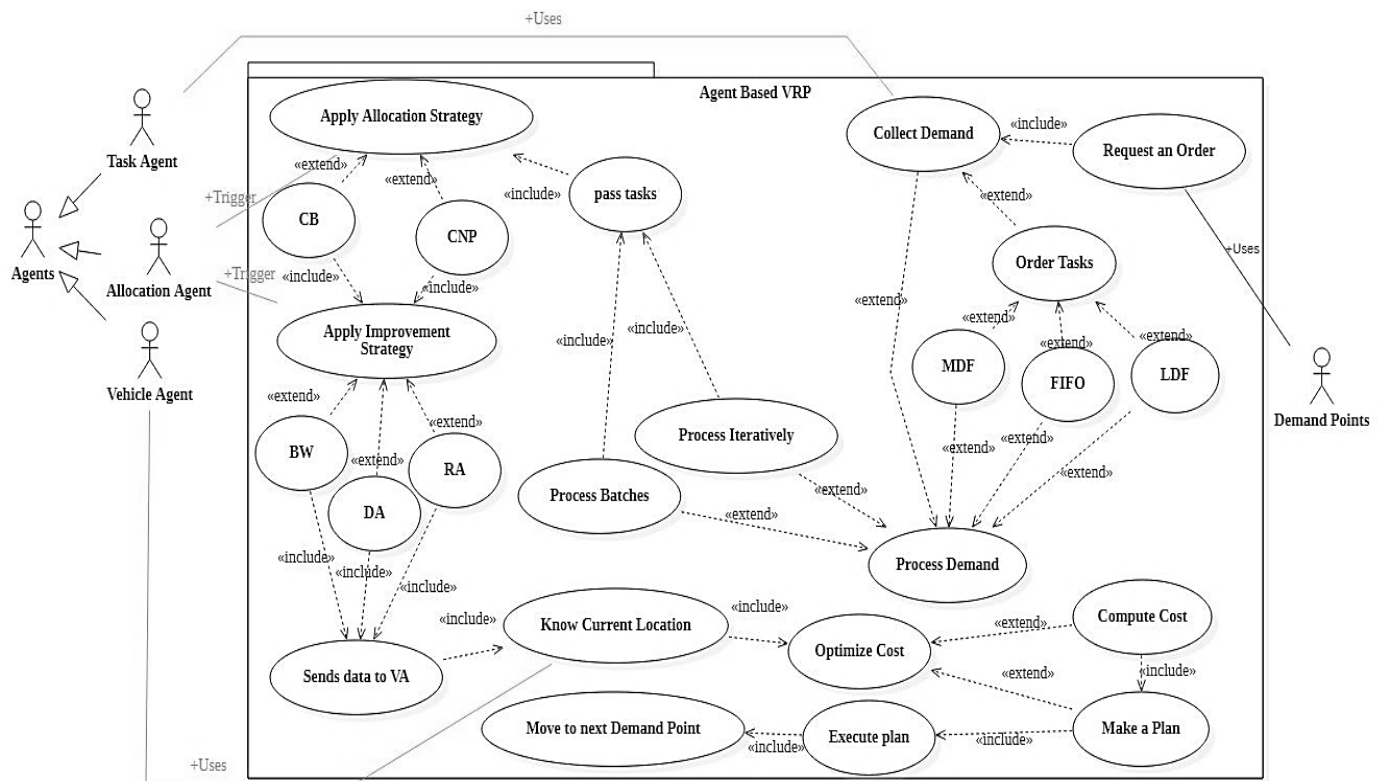


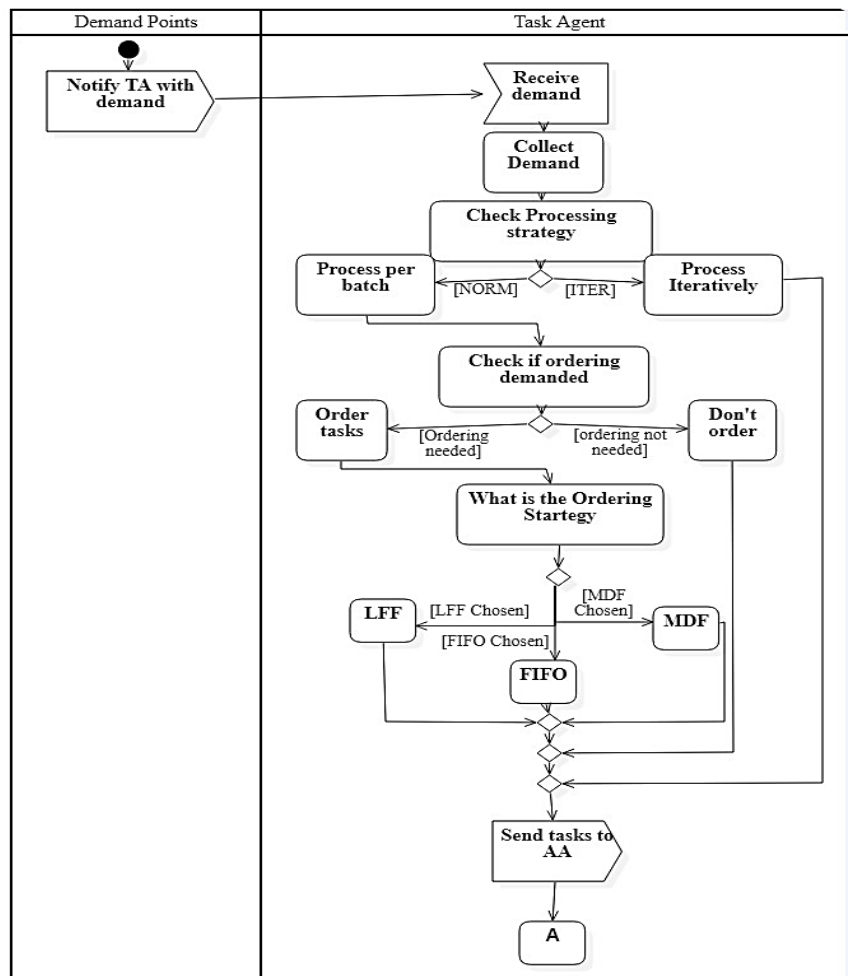
Figure 3. Use case diagram

Table 1. Business actors glossary

Actor	Synonym	Description
Task Agent (TA)		A system that is Responsible for collecting Demand from the demand points, sorting, and sending them.
Allocation Agent (AA)		A system that is Responsible for allocating the demand points to partitions in the most cost-efficient way possible.
Vehicle Agent (VA)	Vehicle	A system that is Represents a vehicle and responsible for the routes ordering.
Demand Points	Customers	A system that notifies the task agent with the demand needed at which point

**Table 2.** Use cases glossary

Use-Case Name	Use-Case Description	Actors (roles)
Most Demand First (MDF)	This use case describes the event of a sorting the demand based on MDF, where the tasks are ordered in a descending order	Tasks Agent (Primary actor)
Least Demand First (LDF)	This use case describes the event of a sorting the demand based on LDF, so the tasks are ordered in an ascending order	Tasks Agent (Primary actor)
First in First Out (FIFO)	This use case describes the event of a sorting the demand based on FIFO where tasks' sequence corresponds to the order of arrival.	Tasks Agent (Primary actor)
Batch Processing (NORM)	This use case describes the event of a Processing the demand-based Batch processing, where all available tasks are sent as one batch	Tasks Agent (Primary actor)
Iterative Processing (ITER)	This use case describes the event of a Processing the demand Iterative processing, where tasks are sent one by one.	Tasks Agent (Primary actor)
Contract-Net Protocol (CNP)	This use case describes the event of allocating the demand to partitions based on Contract-Net Protocol. For every task, the best Vehicle Agent is selected according to insertion estimation satisfying capacity constraints. This strategy contains no backtracking and in case of allocation failure (because of capacity constraint of Vehicle Agents) the whole process is restarted with an added vehicle	Allocation Agent
Capacity Backtracking (CB) strategy	This use case describes the event of allocating the demand to partitions based on CNP, but with backtracking in case of allocation failure. In case when no Vehicle Agent can undertake a new task because of the capacity constraint, the best Vehicle Agent is selected regardless of capacity limitations. This agent removes the worst tasks until the new task fits the increased free space. After that, the removed tasks are allocated again. The reallocation counter controls the number of reallocations and when it reaches the pre-defined maximum, the number of Vehicle Agents is increased, and the process is restarted.	Allocation Agent
Delegate Worst (DW)	This use case describes the event of improving the allocated partitions by delegating the worst tasks if the savings are higher than the insertion cost.	Allocation Agent
Delegate All (DA)	This use case describes the event of improving the allocated partitions by delegating all tasks if the savings are higher than the insertion cost.	Allocation Agent
Reallocate All (RA)	This use case describes the event of improving the allocated partitions as each Vehicle Agent successively removes all its tasks from the plan and allocates it again using the CNP strategy.	Allocation Agent



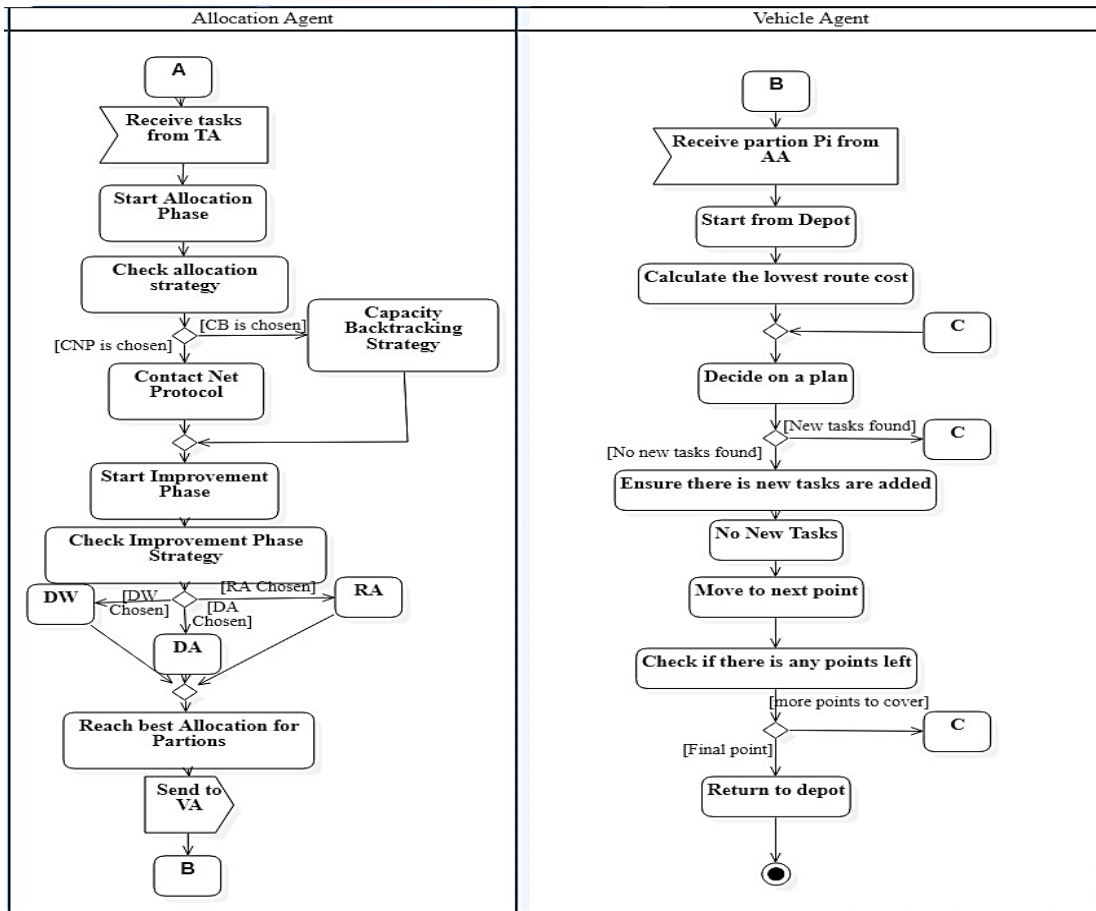


Figure 4. Activity diagram

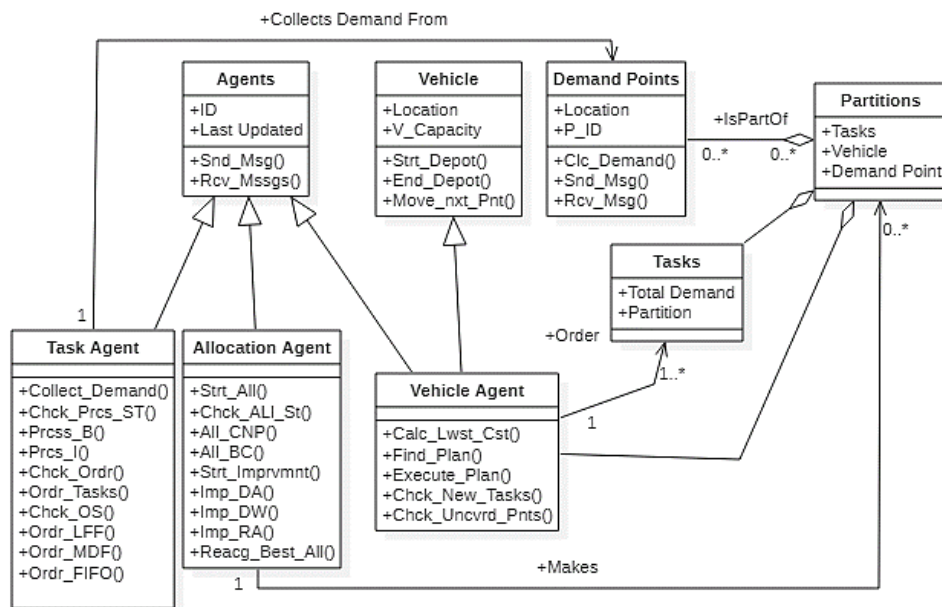


Figure 5. Class diagram

### 3.1.3 Behavioral modelling

The state-machine diagram given in Figure 6 illustrates the state of the Allocation Agent (AA). The AA is chosen to be illustrated by a state machine diagram as it is the actor with the largest number of states and is the most complex one.

As shown in Figure 6, AA's "Starting Allocation phase" state is triggered when it receives tasks from the Task Agent. The allocation next state is based on the allocation strategy. If the allocation strategy of the tasks is to utilize CNP, then AA

state will be "Allocating Using CNP". However, if the allocation strategy is CB, then the AA's state will be "Allocating Using CB". This concludes the allocation phase. The improvement phase is then triggered to improve the initial allocation, making the state of the AA "Starting Improvement Phase". The next state is determined by the improvement strategy adopted. If it is Delegate Worst, AA's state will be "Improving using DW". While if the strategy is to Delegate All, AA's state will be "Improving using DA". If the strategy



is to Re-allocate All, AA's state will be "Improving using RA". The next state would be "Reaching Best Allocation for a partition". When the improvement phase is done and the best allocation is reached, the state would be "Sending the partitions to VA". This means that each partition is to be sent to its allocated vehicle.

The sequence diagram in Figure 7, illustrates how objects interact with each other. The diagram emphasizes how messages are sent and received between objects [26, 27]. To represent the example of a sequence diagram, "Delegate Worst" is chosen among use cases of the system to show its sequence. This specific sequence is chosen due to its complexity. Many scenarios can occur in a single use case and each scenario is related to a sequence diagram. Thus, a use case can have many

sequence diagrams. Since sequence of the primary scenario is complex enough, for ease of readability, the sequence diagram of the primary scenario is given in Figure 7.

Figure 7 illustrates the improvement of the allocation strategy, ignoring any error due to allocation infeasibility. For example, the Allocation Agent starts by inquiring about the worst task from Class "Partition: Partitions" to find the task with the highest cost in the "Tasks" class. The worst task's Insertion Cost (IC) in various vehicles is computed then the vehicle having the minimum IC is called the winner. Next, the IC of the winner is compared to the savings (Sv) of keeping the task at its delegated vehicle. If ICs is less than Sv, the task is to be transferred. Otherwise, nothing changes.

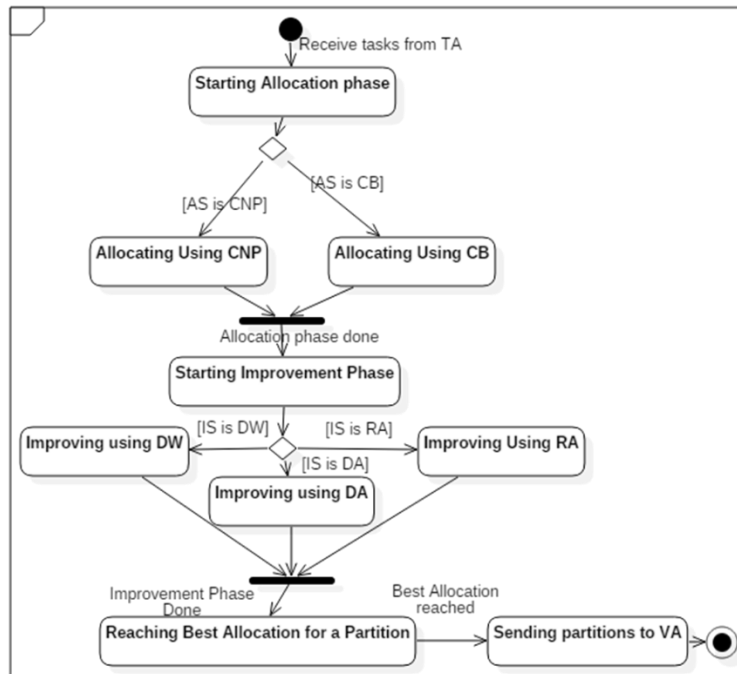


Figure 6. State-machine diagram

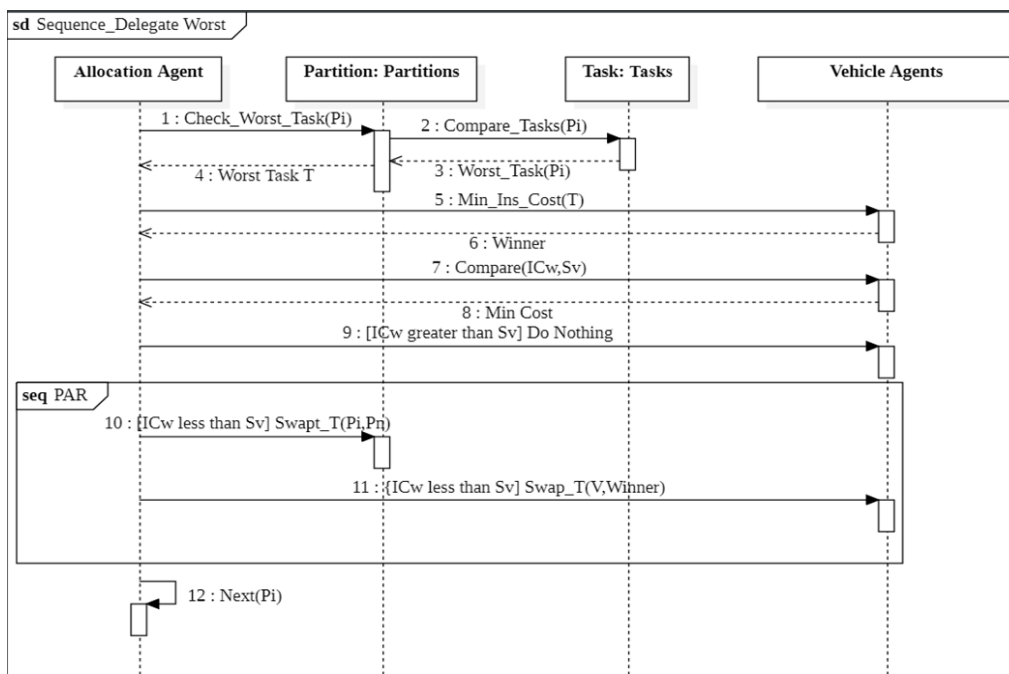


Figure 7. Sequence diagram

#### 4. CONCLUSION

UML modeling is a powerful tool of visualization for complex systems. The vehicle routing problem (VRP) is among the most studied problems in the field of supply chain management. Agent-based methodology have proven efficient in representing many real-life problems. It provides a clear representation of problems in a flexible manner that facilitates the decision-making process. In this paper, the Agent Based VRP was addressed from the modeling viewpoint of UML modeling paradigm. The objective was to provide a clearly visualized representation of the problem, which serves as an input for software engineers who will design and implement the necessary software to solve the VRP problem. Through building functional, structural, and behavioral models of an Agent Based VRP system, this paper proposed a novel perspective for modeling and visualizing the classical VRP. The approach could extend to include several variants of the problem to assist in the decision-making process in one of the known challenges in supply chain management, namely, the vehicle routing problem and its variants.

#### REFERENCES

- [1] Eriksson, H.E., Penker, M. (2000). *Business Modeling with UML: Business Patterns*. New York: Wiley.
- [2] Janis, O., Uldis, D. (2017). *Topological UML Modeling: An Improved Approach for Domain Modeling and Software Development*. Elsevier Science Publishers B. V., PO Box 211 1000 AE Amsterdam, Netherlands.
- [3] Kassem, S., Chen, M. (2013). Solving reverse logistics vehicle routing problems with time windows. *The International Journal of Advanced Manufacturing Technology*, 68: 57-68. <http://dx.doi.org/10.1007/s00170-012-4708-9>
- [4] Asma, B., Mourad, A. (2017). Object-oriented modeling and design of reverse logistics management system using UML. *5th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS)*, Rome, Italy, pp. 1-8.
- [5] Manuel, R., Claudius, S., Ralf, S., Stephan, R., Markus, T. (2017). Digital representation in multicopter design along the product life-cycle. *Conference on Intelligent Computation in Manufacturing Engineering*, pp. 559-564. <https://doi.org/10.1016/j.procir.2016.06.008>
- [6] Dantzig, G.B., Ramser, J.H. (1959). The truck dispatching problem. *Management Science*, 6(1): 80-91. <https://doi.org/10.1287/mnsc.6.1.80>
- [7] Clarke, G., Wright, J.W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research Journal*, 12(4): 519-543. <https://doi.org/10.1287/opre.12.4.568>
- [8] Toth, P., Vigo, D. (2002). An overview of vehicle routing problems. *The Vehicle Routing Problem*, 1-26. <https://doi.org/10.1137/1.9780898718515.ch1>
- [9] Kassem, S., Korayem, L., Khorshid, M., Tharwat, A. (2019). A hybrid bat algorithm to solve the capacitated vehicle routing problem. *2019 Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, Giza, pp. 222-225. <https://doi.org/10.1109/NILES.2019.8909300>
- [10] Fabien, L., Olivier, P., Fabien, T. (2020). A lexicographic minimax approach to the vehicle routing problem with route balancing. *European Journal of Operational Research*, 282(1): 129-147. <https://doi.org/10.1016/j.ejor.2019.09.010>
- [11] Sandra, Z., Sandra, H. (2020). Objectives and methods in multi-objective routing problems: A survey and classification scheme. *European Journal of Operational Research*. <https://doi.org/10.1016/j.ejor.2020.07.005>
- [12] Korayem, L., Khorshid, M., Kassem, S. (2015). A hybrid k-means metaheuristic algorithm to solve a class of vehicle routing problems. *Advanced Science Letters*, 21(12): 3720-3722. <https://doi.org/10.1166/asl.2015.6555>
- [13] Doerner, K.F., Hartl, R.F. (2008). Health care logistics, emergency preparedness, and disaster relief: New challenges for routing problems with a focus on the Austrian situation. *Operations Research/Computer Science Interfaces the Vehicle Routing*, 43: 527-550. [https://doi.org/10.1007/978-0-387-77778-8\\_24](https://doi.org/10.1007/978-0-387-77778-8_24)
- [14] Shaopeng, Z., Rong, C., Yu, J., Zhong, W., Allan, L., Otto, A.N. (2020). Risk-averse optimization of disaster relief facility location and vehicle routing under stochastic demand. *Transportation Research Part E: Logistics and Transportation Review*, 141: 102015. <https://doi.org/10.1016/j.tre.2020.102015>
- [15] Stahlbock, R., Voß, S. (2008). Vehicle routing problems and container terminal operations – An update of research. In: Golden B., Raghavan S., Wasil E. (eds) *The Vehicle Routing Problem: Latest Advances and New Challenges*. *Operations Research/Computer Science Interfaces*, vol 43. Springer, Boston, MA. [https://doi.org/10.1007/978-0-387-77778-8\\_25](https://doi.org/10.1007/978-0-387-77778-8_25)
- [16] Macal, C., North, M. (2005). Tutorial on agent-based modeling and simulation. *Winter Simulation Conference*, Orlando, FL, USA. <https://doi.org/10.1109/wsc.2005.1574234>
- [17] LeBaron, B. (2002). Short-memory traders and their impact on group learning in financial markets. *Proceedings of the National Academy of Sciences of the United States of America*, 99(3): 7201-7206. <https://doi.org/10.1073/pnas.072079699>
- [18] Jacob, L., Niels, B., Rainer, L. (2020). Analysis of resilience strategies and ripple effect in blockchain-coordinated supply chains: An agent-based simulation study. *International Journal of Production Economics*, 228: 10782. <https://doi.org/10.1016/j.ijpe.2020.107882>
- [19] Caihong, L. (2020). Multi-agent modeling of the collaborative operation of the producer service supply chain under the intelligent manufacturing clusters in the Yangtze River Delta. *Journal Européen des Systèmes Automatisés*, 53(4): 487-492. <https://doi.org/10.18280/jesa.530406>
- [20] Wei, D.F. (2020). Modeling and simulation of a multi-agent green supply chain management system for retailers. *Journal Européen des Systèmes Automatisés*, 53(4): 549-557. <https://doi.org/10.18280/jesa.530414>
- [21] Erik, C. (2020). An agent-based model to evaluate the COVID-19 transmission risks in facilities. *Computers in Biology and Medicine*, 121: 103827. <https://doi.org/10.1016/j.combiomed.2020.103827>
- [22] Carley, K.M., Fridsma, D.B., Casman, E., Yahja, A., Altman, N., Chen, L.C., Nave, D. (2006). BioWar: scalable agent-based model of bioattacks. *IEEE Transactions on Systems, Man, and Cybernetics*, 36(2): 252-265. <https://doi.org/10.1109/TSMCA.2005.851291>



[23] Page, B., Knaak, N., Kruse, A. (2007). A discrete event simulation framework for agent-based modelling of logistic systems. *INFORMATIK 2007: Informatik trifft Logistik*. Band 1. Beiträge der 37. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 24-27. September 2007 in Bremen, pp. 397-404.

[24] Meyer, R. (2015). *Event-driven multi-agent simulation*. Paris, France: Springer, pp. 3-16. [http://dx.doi.org/10.1007/978-3-319-14627-0\\_1](http://dx.doi.org/10.1007/978-3-319-14627-0_1)

[25] Vokříněk, J., Komenda, A., Pěchouček, M. (2010). Agents towards vehicle routing problems. *9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*. Toronto, Canada, 1: 773-780.

[26] Jacobson, I., Booch, G., Rumbaugh, J. (1999). *The Unified Modeling Language User Guide*. Addison Wesley.

[27] Kande, M., Mazaher, S., Prnjat, O., Sacks, L., Wittig, M. (1998). Applying UML to design an inter-domain service management application. *The Unified Modeling Language, «UML»'98: Beyond the Notation, First International Workshop, Mulhouse, France, June 3-4, 1998, Selected Papers*, pp. 200-214, 199. [https://doi.org/10.1007/978-3-540-48480-6\\_16](https://doi.org/10.1007/978-3-540-48480-6_16)

## NOMENCLATURE

VRP	Vehicle Routing Problem
ABMS	Agent Based Modelling
UML	Unified Modelling Language
CNP	Contract-Net Protocol
IDEF0	Integration Definition0
MDF	Most Demand First
LDF	Least Demand First
FIFO	First in First Out
TA	Task Agent
VA	Vehicle Agent
AA	Allocation Agent
CB	Capacity Backtracking
DW	Delegate Worst
DA	Delegate All
RA	Re-allocate All
ITER	Iterative Processing
NORM	Batch Processing
ABVRP	Agent Based Vehicle Routing Problem
IC	Insertion Cost
Sv	Savings