International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 13, No.1 (Mar-23)

http://dx.doi.org/10.12785/ijcds/130142

Modelling of 3rd Party Logistics Provider Using Unified Modelling Language

Basma Ismail[1], Michael Geris[1], Nour Abdullatif[1] and Sally Kassem[2,1]

[1] Industrial Engineering, Smart Engineering Systems Center, Nile University , Giza, Egypt
[2]Operations research and Decision Support, Faculty of Computers and Artificial Intelligence, Cairo University, Giza, Egypt

E-mail address: Ba.ismail@nu.edu.eg, M.geris@nu.edu.eg, n.sameh@nu.edu.eg, s.kassem@fci-cu.edu.eg

Received 4 May. 2021, Revised 27 Dec. 2021, Accepted 5 Mar. 2023, Published 16 Mar. 2023

Abstract: Transportation is a vital driver in logistics management. Third party logistics (3PL) involves a company outsourcing logistics services to other companies. Third party logistics aids in facilitating and optimizing transportation by aggregating demand for trucks and drivers. This paper proposes a model of a transportation 3PL System. The system under study is modelled using the Unified Modelling Language 2.0 (UML 2.0) through an analysis of 3PL provider located in Egypt. The system is analyzed and identified with Integrated Definition for Function Modeling Diagram (IDEF0) & context diagram. Functional, structural and behavioral models are formulated to represent the system, using UML 2.0. The system consists of multiple main players: The client, driver, trip operator and the finance department. Since 3PL providers are complex systems requiring accurate analysis and representation, the developed models serve as a visualization tool for better understanding and management of the 3PL provider.

Keywords: UML Modeling, 3PL, TMS

1. INTRODUCTION

Unified Modeling Language (UML) is used to show and exhibit the processes and behaviors that a system goes through. Its main goal is to standardize a common visual theme for all systems to be identified with [1] [2]. It is used for software systems modeling as well as business and non-software systems. UML modeling adopts the object oriented approach focusing on decomposing complex systems into naturally existing objects that contain both data and processes. Hence, creating clear visualization of complex systems from static and dynamic view points [3] [4] [5].

The advantages of UML modeling include: flexibility and readability, proper exhibition of a system's architecture, facilitating the planning phase of a program, as well as facilitating the debugging process [6]. Several research work use UML modeling to exhibit multiple systems' in diverse fields, functionality, structure & behavior. It also has methods for specifying UML security stereotypes, with the goal of guiding developers by marking vulnerable model sections and allowing automatic security test case development, which aids with security concerns. [7] In [8], Discrete Event Specification (DEVS) model is introduced and applied on a logistics system while integrating UML modeling. The authors used the atomic model and the coupled model. The paper suggests that DEVS is a great conceptual modeling tool but does not provide the inputs to perform a simulation process. The paper, then proposes improving the model by extending it with UML modeling, which is considered an implementation model. In [9], UML is implemented on a pilot project in China. Process and information domains are modeled by use case diagram, sequence diagram and class diagram. UML is used to model the classical vehicle routing problem (VRP), while utilizing an agent-based system in [10]. Another work, [11], proposes a system for production planning by developing a use case diagram, an activity diagram, and a communication diagram as a means of visualization for the system. Mohammed & Kassem in [12], addressed a crucial issue in Egypt which is public commuting. The authors represent an online public bus reservation system through illustrating its functions and structure by developing a UML model. E-learning system was modeled using UML. UML modelling has been successfully implemented in the domain of crime record management systems as given in [13].

Transportation is, an important supply chain driver since goods must move across all supply chain stages to make it to the end customer [14]. Responsive transportation

is used to decrease the number of facilities that companies must own to cover all areas of deliveries; hence, centralizing inventories [15]. The process of transportation consists of two main parties: the carrier & the shipper. The carrier transports goods and tries to maximize the revenue out of the process since it is the entity that is making the investments decisions. The carrier picks the transportation mode (air, maritime, rail, intermodal or pipeline) to reach the objective of revenue maximization while maintaining a constant responsiveness level. The shipper's goal is to minimize the expenses of the transportation process. Nonetheless, the shipper should maintain the same level of responsiveness promised to the end customer. Carrier's level of responsiveness is mainly determined and restricted by the available modes of transportation and infrastructures in the specified country/countries that goods will move through [16]. Another important aspect of logistics is reverse logistics. The work of [17] presents analysis of both the process of the 3rd Party Logistics Company(3PL) that perform reverse logistics management systems.

As a part of supply chain management systems, transportation management system (TMS) is a logistics platform that requires planning, operating & improving day to day logistics activities. It provides a proper documentation and analysis of shipments, facilitates the management process of shipping products while also ensuring the best responsiveness of the whole delivery system.

The importance of TMS lies in that it provides a comprehensive visibility, that leads to a better customer satisfaction, hence, generating more revenue for the supply chain. TMS users could be one or more of the following: Retailers, distributors, logistics companies, ecommerce companies and manufacturers. Benefits of implementing an efficient TMS are reducing costs, reducing delivery time, increasing customer satisfaction, increasing security & simplified interface for controlling freights [18]. Optimizing transportation and efficient TMSs have been recognized by several researchers, for example, [19][20].

The importance of modeling several supply chain drivers using UML has been addressed in several research work. On the other hand, third party logistics providers have attracted the attention of many researchers due to its importance. Research on the 3PL has addressed several issues ranging from optimization, cost analysis, customer satisfaction, and several other perspectives. However, after conducting an extensive literature review, no work has been dedicated for modeling 3PL using UML modeling approaches, namely functional, structural, and behavioral models. We believe that 3PL providers, especially in transportation is a complex model, that requires modelling in a visual manner for two reasons: first to allow managers to identify and understand the system's details and hence manage and improve the current system. Second, to allow software engineers to build efficient software that captures all the important details without missing critical aspects and at the same time, without including unnecessary details. Therefore, this work is concerned with providing different UML models that provide clear visualization of a 3PL provider, with main focus on transportation. The developed UML models represent a case study for a third party transportation provider located in Egypt. Although the models represent a case study, most third party transportation providers perform similar processes. Hence, the developed UML models are useful for representing the majority of third party transportation systems.

The rest of this paper is organized such that Section 2 explains details of the system through the problem statement, Section 3 provides details of the different UML models developed to represent the system, and finally, Section 4 concludes the paper.

2. PROBLEM STATEMENT

In this paper, the system reaction to a client's request for a delivery truck is studied. The system starts from the truck request till the delivery of the product(s) to the end distribution center, or customer. Such information is attained by conducting an interview with an employee in a third party transportation provider, referred to in this research as Company X.

The main actors that contribute to the system are: New or old client, admin, finance, trip operator & truck driver.

Company X has two types of contracts:

Monthly contract: The client has a limited number of kilometers and hours per 26 days.

On-Demand contract: The client can request a truck any time. The contract expires after 26 days and contract trips are priced by $/KM.

Fig. 1 illustrates a context diagram, known as a level 0 data flow diagram. The main purpose of the diagram is to identify the flow of information between the system and the external actors who are the driver and the client in the system. Starting by the client, a client sends a request for a trip. Based on this request the system will be working on generating the bill to be sent to the client and will send the location and trip details to an assigned Driver. The system will wait for the driver's response of accepting or rejecting the trip. If the driver accepts the request, he will be required to update the system with the trip progress. Finally, the bill gets generated on the driver's application and the client scans it by using his application as well to mark the trip as complete.

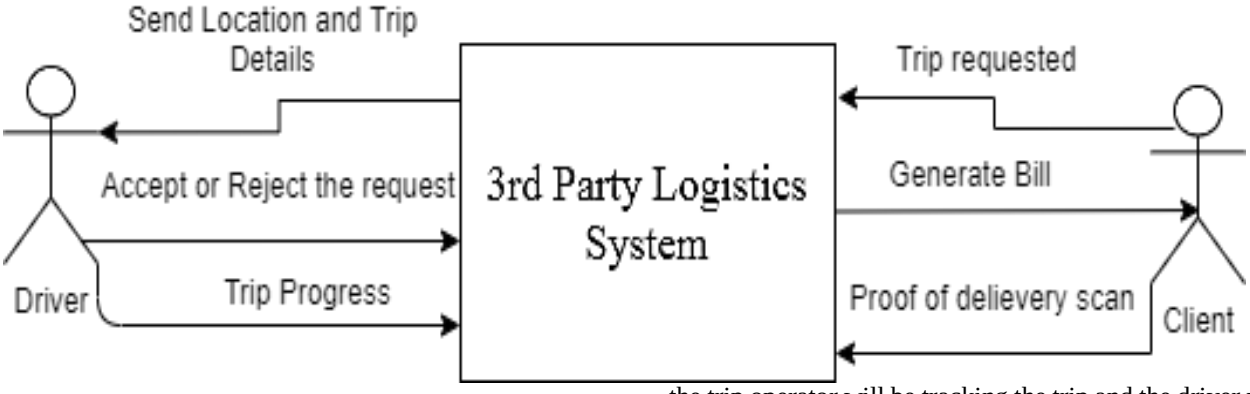

Figure 1. Context Diagram

Fig. 2 illustrates Integrated Definition0 (IDEF0) which defines the main purpose of the system.

The input is the request being sent from a client. The output is the bill, which needs to be scanned by the customer to end the requested trip. Moving to the controls, the system has 6 controls which are used mainly inside the system to process the request. The controls are as follows: check contract type being used, process the request, trip acceptance or rejection, send location and trip details, track the trip and update trip progress.

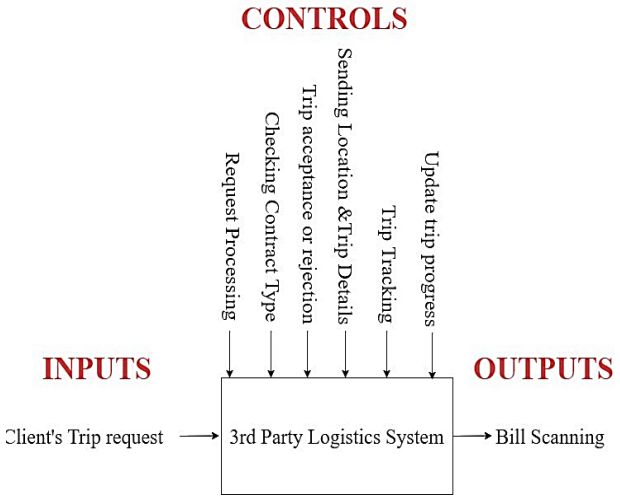

Figure 2. IDEF0 Diagram

The system begins operation once the client sends a request for a trip. The request includes information about the contract type, location and type of vehicle needed. The admin will process the request and send it to finance for checking which contract this client is using to calculate and generate the trip bill. Simultaneously, the trip operator assigns a driver for this trip and waits for the acceptance or rejection from the driver. Once a driver accepts the request, the trip operator will be tracking the trip and the driver will be updating its progress. Finally, once the driver arrives to the location, the client scans the generated bill using the mobile phone application from the driver's mobile phone.

3. UML MODELS

A. Functional modeling

A use case diagram, one of the functional diagrams, describes major actions (use cases) where the system should perform and how these actions are associated with external entities (actors) [1].

In Fig. 3 a use-case diagram is used to illustrate a functional model for the actors. Table 1 contains a list of use cases, their explanation, and the related actor(s). The actors are defined in Table 2. The system starts once the client requests a truck to deliver their products; The admin takes this request and sends it to two departments: Trip Operator and Finance. Both departments will work in parallel. Starting with the finance department, it will check which contract the client is legally bonded by. Accordingly, the finance department will calculate the trip cost and create the bill invoice.

When the trip operator receives the request, he sends the trip data, details, and location to the assigned driver for this delivery. Consequently, the driver has the freedom to either proceed or cancel the pending request. If the driver accepts the request, the trip operator will be tracking the order status until its delivery to the client. The moment the driver arrives, he notifies the trip operator. The client will scan the bill invoice. Finally, when the driver leaves the

drop off location, he notifies the trip operator of his departure.

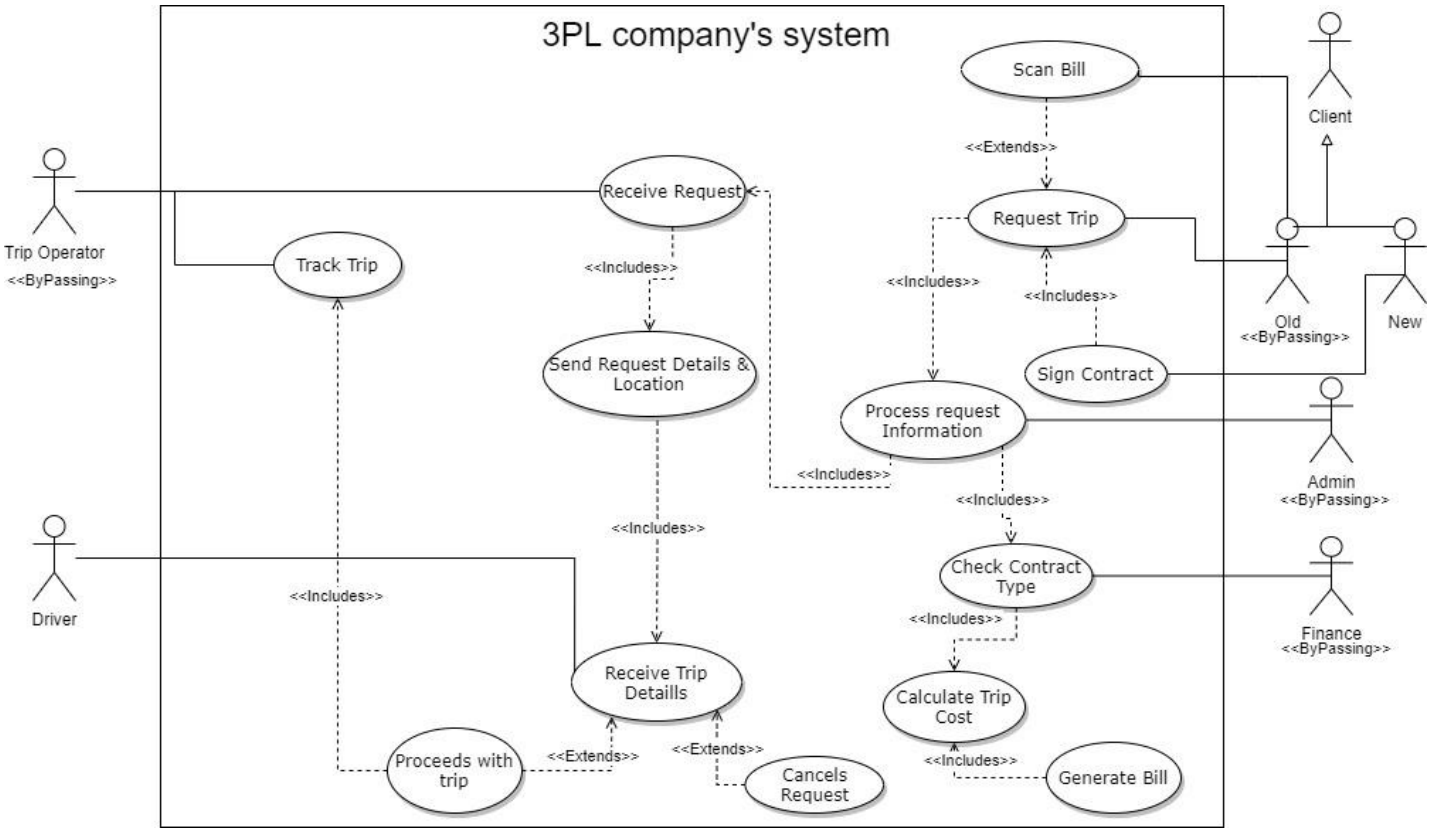

Figure 3. Use Case Diagram

TABLE I. USE CASE GLOSSARY

Use Case	Explanation	Actors
Sign Contract	This use case describes this event when a client checks different types of contracts and choose the type he wants and then sign it.	Client
Request Trip	This use case describes this event of a client needs to deliver products from one to point to another.	Client
Process request information	This use case describes this event when a trip operator receives the request and start processing it to be sent to finance and trip operator	Admin
Check Contract Type	This use case describes this event when the finance receives the request and starts revising client's requests to check the contract type, he/she uses.	Finance
Calculate Trip cost	This use case describes this event executed after checking the contract type	Finance
Generate Bill	This use case describes this event after calculating the trip cost to get the bill.	Finance
Receive Request	This use case describes this event when admin sends the request to trip operator.	Trip Operator
Send Request details & location	This use case describes this event when the trip operator assigns a driver and send to him/her the details of the whole trip.	Trip Operator
Receive Trip Details	This use case describers this event when the driver gets the notification that he has been assigned for a trip.	Driver
Proceed	This use case describers this event when the driver accepts the request.	Driver
Cancel	This use case describers this event when the driver rejects the request	Driver
Track Trip	This use case describers this event when the driver accepts the request and then Trip operator will be tracking the	Trip Operator

	truck	
Scan Bill	This use case describers this event when the driver has arrived, and the client needs to scan the bill to make a proof that the request had been completed and arrived.	Client

TABLE 2. BUSINESS ACTORS GLOSSARY

Actors	Explanation
Client/Customer	A person who triggers the system when he/she requests a trip after choosing and signing the contract.
Admin	A person who receives the request from the client and process it to be send to another actors (Finance and Trip-Operator).
Finance	A person who checks' contract type of the request and based on the bill will be generated.
Trip-Operator	A person who simply assign a Driver for the trip and after driver's acceptance, he will send the location and track the trip.
Driver	A person who executes the trip.

An activity diagram is one of the functional diagrams represented by UML modeling. It is used to describe a dynamic flow of the system based on the workflow of activities that run in the system [1]. The activity diagram representing the current system is divided according to the following swim lanes: Customer, Admin, Finance, Trip Operator & Driver. The system starts in the customer swim lane with a decision node that is used to differentiate between old and new customers. If the customer is new, he will choose the type of contract he wants (monthly or on-demand) then signs it. Security issues have always been a topic to consider in several systems, a few examples include IoT systems [21], and cloud service systems [22]. UML models have their own security threats as well [7]. To overcome security threats, security and validation stereotypes are added to vulnerable parts of the model developed in this work. Stereotype <<encrypt>> is used to alleviate the vulnerability of sending sensible data through the network without encryption. At the end, the customer will request a trip. For security and model validation reasons, Stereotypes <<Bypassing>> and <<sqlinjection>> are used to depict that only old client, the actor, is authorized to access this activity and that the data inserted is added to the database. Moving to the admin swim lane, the admin receives & processes the request. Then, the request is sent to the finance swim lane and trip operator to work on, in parallel. The finance swim lane takes the request and starts checking which contract the customer is using (monthly) or (on demand). For the monthly contract, the finance will calculate the trip cost by how many hours the trip will take; On the other hand, if it is an on-demand contract, the trip will be calculated by how many kilo meters the trip will take and multiply it by the decided price. Finally, the finance department will generate the bill. To ensure the validity of the user and secure the process, Stereotype <<Bypassing>> is used to depict that only Finance, the actor, is authorized to access this activity. For the trip operator swim lane, the operator will send the request to a driver. The driver will receive the details of the trip and must choose between two choices either to proceed or cancel. Once a driver chooses to proceed with the request, the operators will add the location and client details on the GPS. At the same time, the operator will be tracking the trip for safety concerns as soon as the driver starts the trip. Afterwards, the system pauses temporarily. However, if the driver cancels the request, there will be a loop returning to the trip operator to request another driver. Finally, in the driver swim lane, when the driver receives the location, he will start the trip. Whenever the driver arrives or leaves a delivery point, he needs to update the trip operator. Finally, once the driver arrives at the destination, he will ask the client to scan the bill, and the system ends. For further illustration, pseudocode is provided to clarify the most complex activities. First, the steps needed for a (New, Old) customer to request a trip:

Algorithm
Begin
Input RequestRide()
Request (credentials)
    Input credentials
    Input contract_type
    Output contract
    IF CredentialsMatching = failed **Then**
        PROMPT "Do you want monthly or on demand contract"
            If Prompt = Yes **Then**
                Signup ()
            SET contract_type
            DISPLAY contract
            Else
            PROMPT "Thank you for using our service"
            ENDIF
    ELSE
        PROMPT request trip
    ENDIF
END

In this Algorithm the steps needed for Finance department to calculate the cost of a trip are shown:

```
Algorithm
Begin
Generate(contract)
        Input contract_type
        Output Bill
        IF contract_type = on demand
                READ hours per trip
                Compute bill as hours times hourly rate
        ELSE
                READ kilometers driven
                Compute bill as kilometers times
kilometers rate
        ENDIF
        DISPLAY Bill
END
```

In this pseudocode, the steps needed for Trip Operator to request a Driver are shown:

```
Algorithm
Begin
Order_driver (trip_data, trip_request)
        Input trip_data
        Input trip_request
        Output track_trip
        PROMPT "Do you accept following trip?"
        IF trip_request = true THEN
                PROMPT send location and details
        ELSE
                CALL Order_driver
        END IF
END
```

In this pseudocode, the steps needed for driver to update trip progress is shown:

```
Algorithm
Begin
Delivery (clients_location)
                Input clients_details
                Output scan_bill
                PROMPT trip_progress
                 IF trip_progress = clients_location THEN
                     PROMPT scan_bill
                   ELSE,
                      PROMPT trip_progress
                        ENDIF
END
```

In this section the process of connecting a driver is shown:

```
Algorithm
Begin
Order_Setup()
        CALL Request
        IF contract = VALID THEN
           CALL Generate
        ELSE, CALL Order_Setup
            IF Bill = true THEN
               CALL Order_driver
                IF TRUE
                   CALL Delivery
        ENDIF
END
```

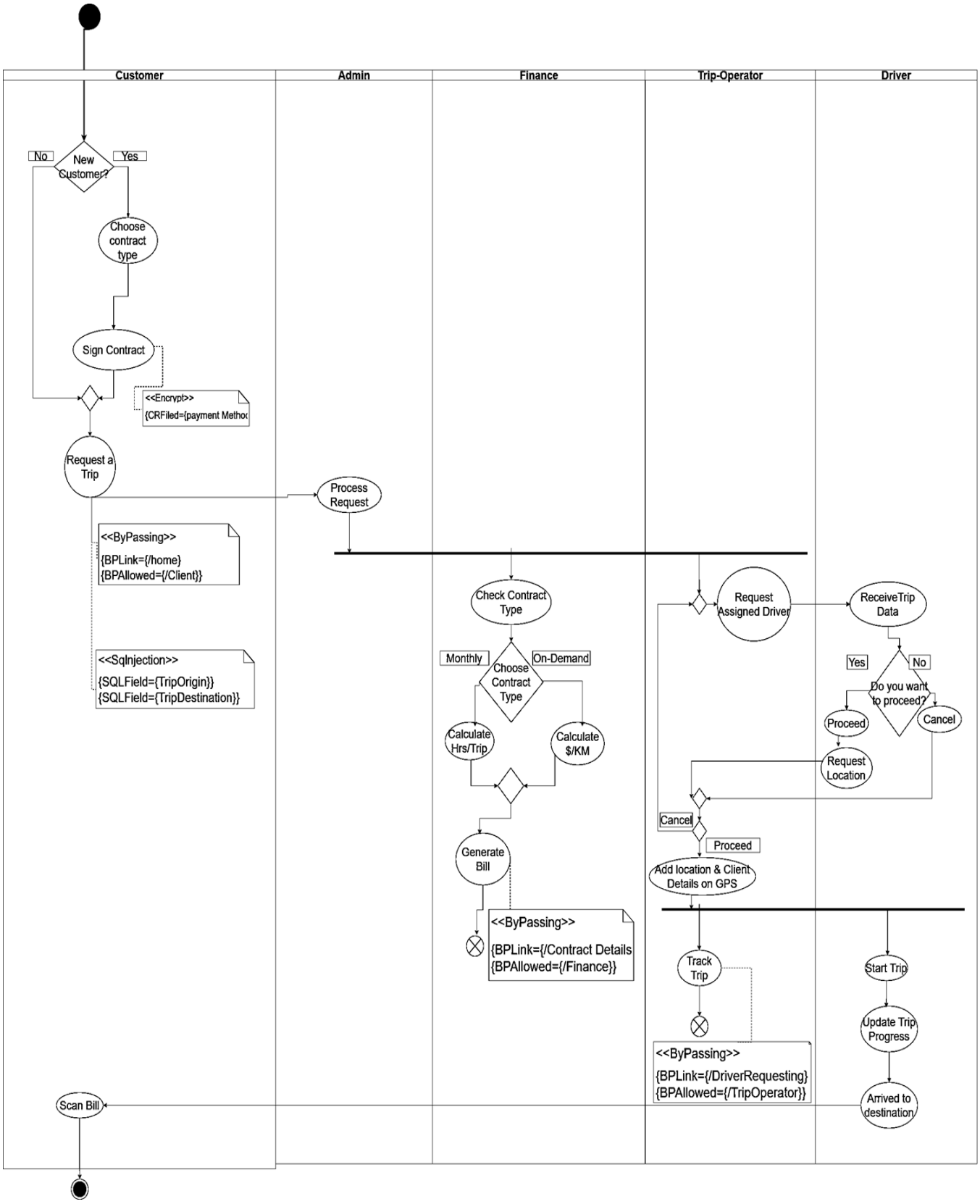

Figure 4. Activity Diagram

B. Structural Modeling

Class diagrams consist of the system's classes which include the operations and attributes. The classes are being linked with different types of relationships [4]. As shown in Fig. 5 below, the class diagram that represents the current system consists of 7 classes, with an association relationship between each class, except that the bill is a part of the contract class and they are connected with an aggregation relationship. The relationships are explained as follows: The client signs a certain contract & requests a trip; the finance department revises the contract, which has the bill class as a part of it. The driver executes the trip, while the trip is being tracked by the class trip operator. Most classes contain IDs to distinguish between each object of the class.

The Main classes are Client, Finance, and Driver. The Client class represents the clients both old and new. The client attributes are ID, Name, Phone Number, Branch Address, and Governorate. A client can Make Payments (), Requests Trips (), sign terms (), and can

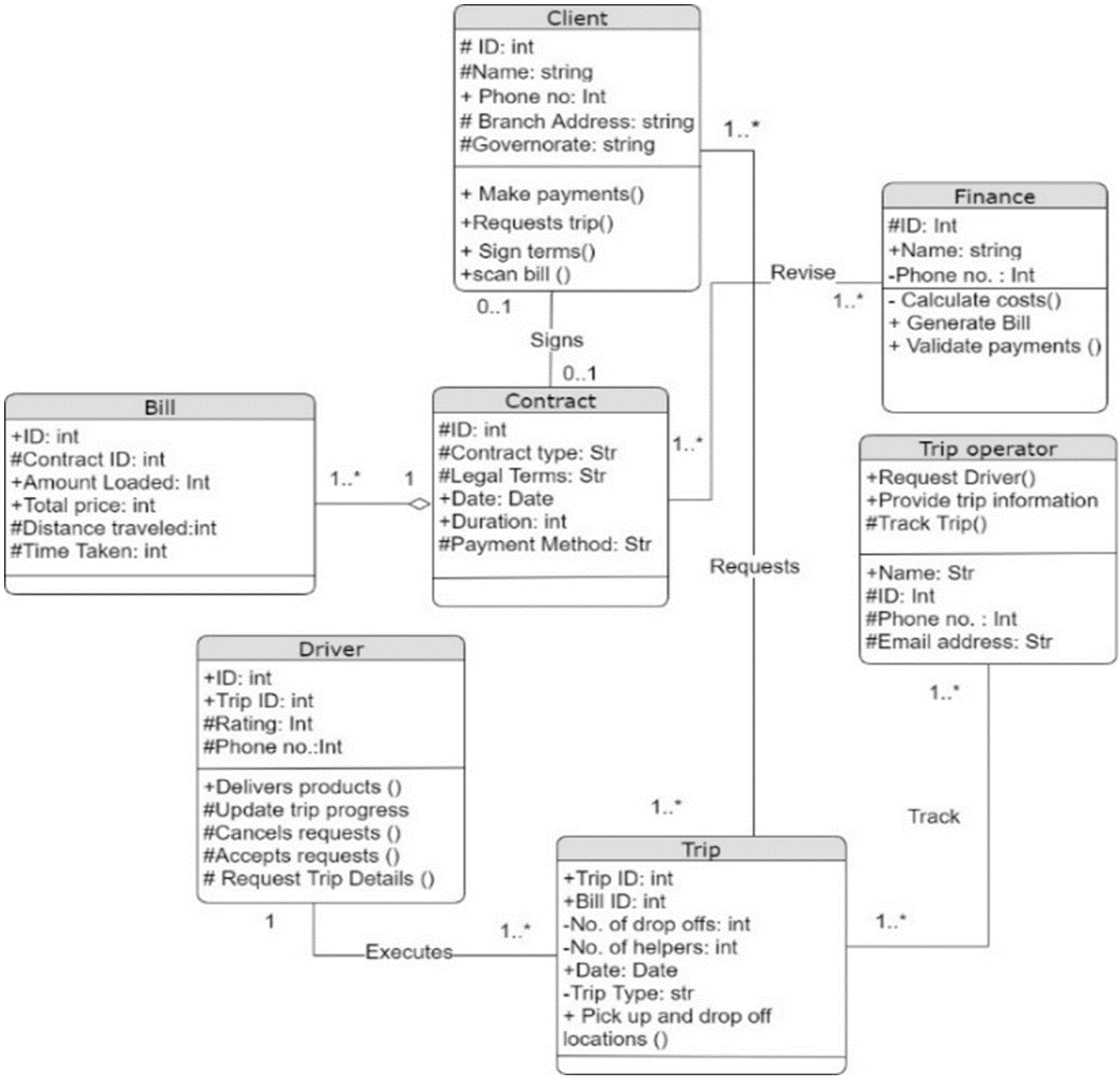

Figure 5. Class Diagram

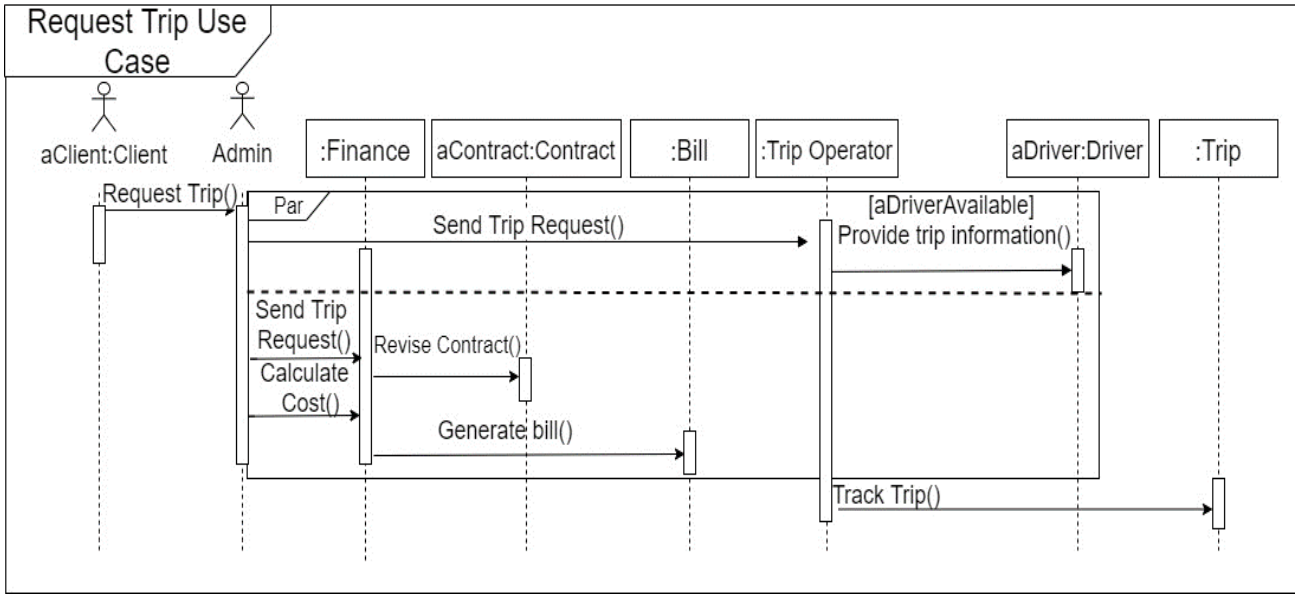

Figure 6. Sequence diagram

Bills (). Finance Class's attributes are ID, Name, and Phone Number while its methods are Calculate Costs (), Generate Bill (), and Validate Payments.

Driver's class attributes are ID, Trip ID, Rating and Phone Number. Driver's methods are Delivers Products (), Update Trip Progress (), Cancel Requests, Accept Requests (), And Request Trips Details ().

C. Behavioral Modeling

A sequence diagram is one of the behavioral diagrams which illustrates the interactions of objects in a time sequence. It illustrates the flow of events to help in identifying what the objects are and the interactions needed to achieve the functionality of the system [23].

In Fig.6, the messages between the objects of the request trip use case are displayed using a sequence diagram. First, a certain client sends a trip request that the admin actor

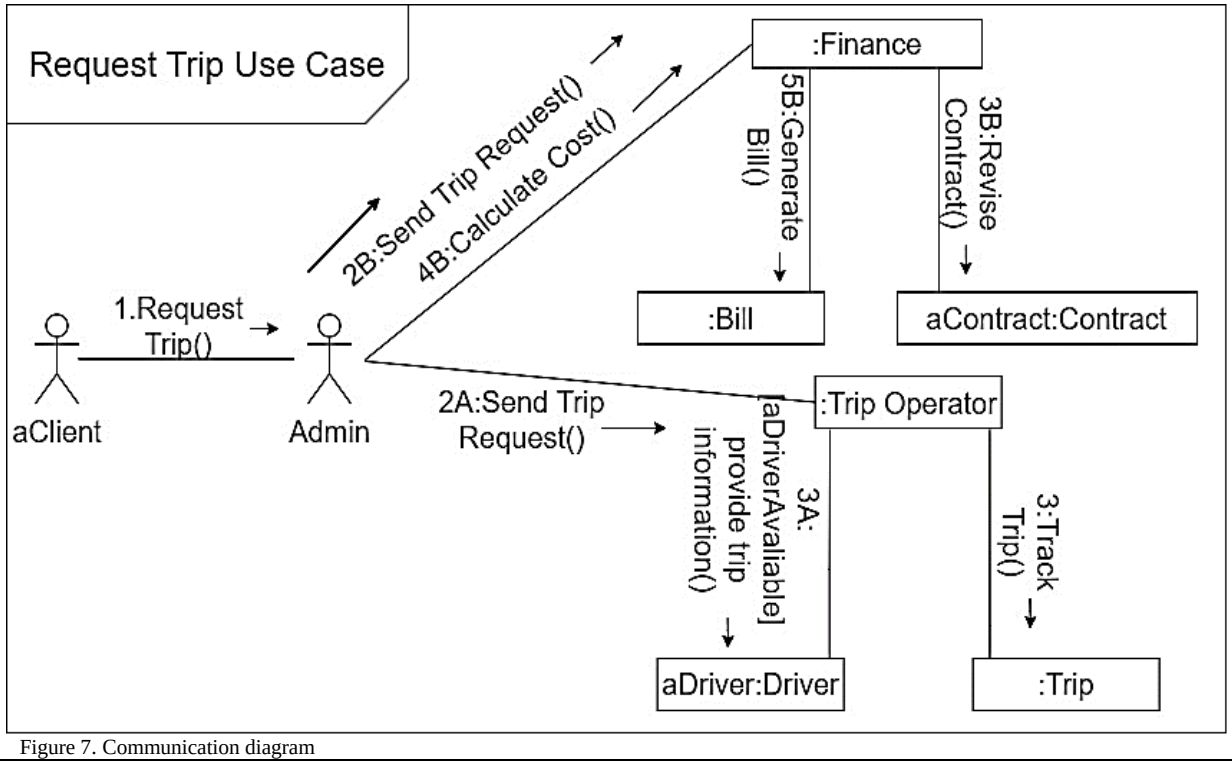

Figure 7. Communication diagram

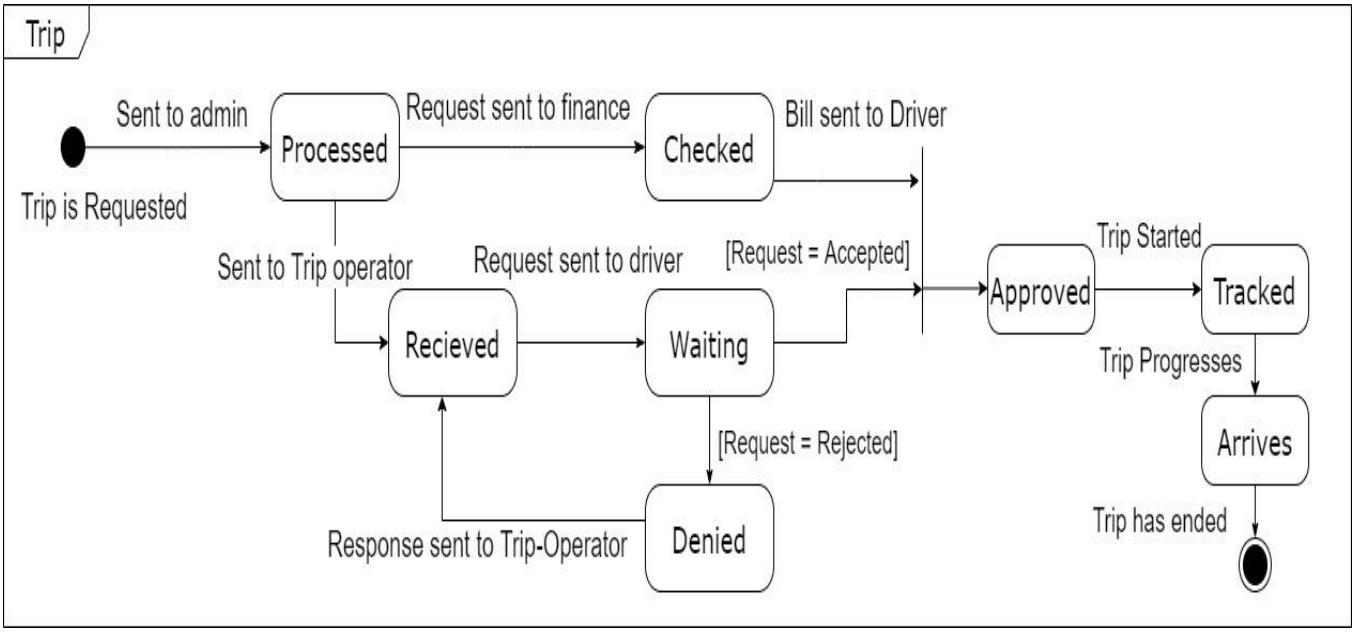

Figure 8. State Machine Diagram

processes and analyzes its information. Then, the admin sends to the finance department an initialization message to calculate the trip's total cost and at the same stage, the admin sends the trip request to the trip operator as well. The trip operator then sends a request to the driver and if the driver accepts and is available, the trip operator provides him with the details of the trip; such condition is shown by the guard condition [aDriverAvailable]. At the same time, the finance department revises the contract by accessing the contract's class for its attributes and then generates the bill. Finally, the trip operator tracks the trip.

Communication diagram, illustrated in Fig. 7, is another type of behavioral diagrams and it shares the same information with the sequence diagram. However, communication diagrams focus more on the messages exchange between objects [24].

Fig.8 shows the different states that the trip class goes through from the start till the end of the system and exhibits what kind of events trigger such change. First, the trip class is initialized, by being requested, the trip gets processed by the admin, gets sent to, both, finance department and trip operator. The finance department checks it while the trip operator sends it to the driver and the request stays waiting for the response of the assigned driver. If the driver accepts, the trip will be approved and gets tracked and arrives to destination. If the driver rejects the request, the request therefore becomes denied and goes back to the trip operator to reassign it.

4. CONCLUSION

Transportation is a very important driver in supply chain management, especially for third part logistics providers. The benefits of UML modeling have been illustrated and proved effective as a visual tool for complex systems' analysis and design. Although UML modeling has been utilized initially as an information systems tool, it is now well recognized by industrial engineers as an important modeling and analysis tool for several industrial engineering systems. Hence, this paper developed several UML diagrams to represent 3PL provider specialised in transportation services. Use case, activity, class, sequence, state & communication diagrams illustrate the 3PL companies' system's structure, behaviors & functionalities. The presented UML model serves as a visualisation tool, and first step towards design and implementation of an efficient software for the case study in hand. Since most 3PL providers in the field of transportation experience similar process, the developed model is useful for most 3PL companies providing transportation services. Future work for this model includes adding more features that transportation 3PL providers may experience, and adding more object and state diagrams to represent complex objects available in the system.

REFERENCES

[1] B. Rumpe, Modeling with UML, Springer, Cham, 2016. https://doi.org/10.1007/978-3-319-33933-7

[2] D. Bork, D. Karagiannis and B. Pittl, "A survey of modeling language specification techniques," Information Systems, vol. 87, 2020. https://doi.org/10.1016/j.is.2019.101425

[3] I. Veitaite and A. Lopata, "Knowledge-Based UML Models Generation from Enterprise Model Technique," in International Conference on Information and Software Technologies, 2017. https://doi.org/10.1007/978-3-319-67642-5_26

[4] D. Dakic, D. Stefanovic, T. Lolic, S. Sladojevic and Anderla, "Production planning business process modelling using UML class diagram," in 17th International Symposium INFOTEH-JAHORINA, East Sarajevo, 2018. 10.1109/INFOTEH.2018.8345535

[5] A. Suchenia, P. Lopata, P. Wisniewski and B. Terlecka, "Towards UML representation for BPMN and DMN models," in III International Conference of Computational Methods in Engineering Science, 2019. https://doi.org/10.1051/matecconf/201925202007

[6] F. Alhumaidan, "A Critical Analysis and Treatment of Important UML Diagrams Enhancing Modeling Power," Intelligent Information Management, vol. 4, no. 5, 2012. 10.4236/iim.2012.45034

[7] P. Karine, O. Alex, Z. Avelino, O. Flávio. "Specifying Security Aspects in UML Models." CEUR Workshop Proceedings. 413, 2009.

[8] Z. Zhiyong, L. Jie and Z. Xinhui, "UML Modeling for Dynamic Logistics System Based on DEVS," TELKOMNIKA Indonesian Journal of Electrical Engineering, 2013. 10.11591/telkomnika.v11i4.2397

[9] Y. Zhou, Y. Chen and H. Lu, "UML-based Systems Integration Modeling Technique for the Design and Development of Intelligent Transportation Management System," IEEE International Conference on Systems, Man and Cybemetics, 2004. 10.1109/ICSMC.2004.1401349

[10] N. Abdullatif and S. Kassem, "Modelling of Agent-Based Vehicle Routing Problem Using Unified Modelling Language," Journal Européen des Systèmes Automatisés, 2020. https://doi.org/10.18280/jesa.530604

[11] S. Nasr and S. Kassem, "Modeling the Production Planning and Control System using UML," in 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference, Giza, 2020. 10.1109/NILES50944.2020.9257906

[12] A. R. Mohammed and S. S. Kassem, "UML Modeling of Online Public Bus Reservation System in Egypt," in 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), Sakheer, 2020. 10.1109/ICDABI51230.2020.9325604.

[13] A. Hisham, A. Ahmed, M. Khaled, , N. Abdullatif, S. Kassem, "Modelling of crime record management system using unified modeling language", Ingénierie des Systèmes d'Information, 2021. https://doi.org/10.18280/isi.260404

[14] V. Lukinskiy and N. Pletneva, "Impact of solutions for goods transportation on business efficiency and traffic safety," in Thirteenth International Confrence on Organization and Traffic Safety Management in Large Cities (SpbOTSIC 2018), Saint Petersburg, 2018. doi.org/10.1016/j.trpro.2018.12.125.

[15] J. Barker, A. Gibson, A. Hofer, C. Hofer, I. Moussaoui and M. Scott, "A competitive dynamics perspective on the diversification of third-party logistics providers' service portfolios," in Transportation Research Part E: Logistics and Transportation Review, 2021. doi.org/10.1016/j.tre.2020.102219.

[16] S. Chopra and P. Meindl, Supply Chain Management, Pearson Education, 2013. 10.1016/j.pursup.2008.08.001

[17] D. Park, "Implementation of 3PL Service Providers' Efficient Reverse Logistics Management System," International Journal of Advanced Science and Technology, vol. 11, pp. 67-78, 2018. http://dx.doi.org/10.14257/ijast.2018.119.06

[18] E. Koster, "The impact of implementing a transport management system by a fertilizer supplier in South Africa," Journal of Contemporary Management, vol. 12, 2016.

[19] M. Shalaby, A. Mohammed, and S. Kassem, "Supervised fuzzy c-means techniques to solve the capacitated vehicle routing problem," The International Arab Journal of Information Technology, 2021.

[20] H. Bakht,et al., "An analysis of Administrative management, financial and security barriers in E-COMMERCE adoption in small to medium size Enterprises (SME'S) in the United Kingdom," International Journal of Computing and Digital Systems, vol. 7, no. 6, pp. 337–346, 2018.

[21] J. Alshaer, "Towards secure IoT: Securing messages dissemination in intelligent traffic systems", International Journal of Computing and Digital Systems, 2019, http://dx.doi.org/10.12785/ijcds/070602.

[22] I. Ahmad, H. Bakht, "Security Challenges from Abuse of Cloud Service Threat", International Journal of Computing and Digital Systems, 2019, http://dx.doi.org/10.12785/ijcds/080103.

[23] G. Greeff and R. Ghoshal, Practical E-Manufacturing and Supply Chain Management, vol. 1. Elsevier , 2004.

[24] S. Friedenthal, A. Moore and R. Steiner, A Practical Guide to Sysml: Systems Modeling Language, San Francisco: Morgan Kaufmann Publishers Inc, July 2008.