



HAL
open science

Ontologies and semantic web for an evolutive development of logistic applications

Hayder Hendi

► **To cite this version:**

Hayder Hendi. Ontologies and semantic web for an evolutive development of logistic applications. Web. Université du Littoral Côte d'Opale, 2017. English. NNT : 2017DUNK0462 . tel-01724396

HAL Id: tel-01724396

<https://theses.hal.science/tel-01724396>

Submitted on 6 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Université Lille Nord de France
Pôle de Recherche
et d'Enseignement Supérieur

ulco UNIVERSITÉ
DU LITTORAL
CÔTE D'OPALE

Numéro d'ordre : 42

ULCO

École doctorale ED Régionale SPI 72
Unité de recherche LISIC

Thèse présentée par **Hayder HENDI**
Soutenue le **4 décembre 2017**

En vue de l'obtention du grade de docteur de l'ULCO

Discipline **Informatique**
Spécialité **web sémantique**

Titre de la thèse

Ontologies et Web Sémantique pour une construction évolutive d'applications dédiées à la logistique

Thèse dirigée par Cyril FONLUPT directeur
Mourad BOUNEFFA co-directeur
Adeel AHMAD co-directeur

Composition du jury

<i>Président du jury</i>	Abderrafiaa KOUKAM	professeur à l'UTBM
<i>Rapporteur</i>	Adnan YASSINE	professeur à l'Université Le Havre Normandie
<i>Examineur</i>	Dominique LECLLET-GROUX	MCF HDR à l'UPJV-Amiens
<i>Directeurs de thèse</i>	Cyril FONLUPT	professeur à l'ULCO
	Mourad BOUNEFFA	Maître de Conférences à l'ULCO
	Adeel AHMAD	Maître de Conférences à l'ULCO



Université Lille Nord de France
Pôle de Recherche
et d'Enseignement Supérieur

ulco UNIVERSITÉ
DU LITTORAL
CÔTE D'OPALE

ULCO

École doctorale ED Régionale SPI 72

Unité de recherche LISIC

Thèse présentée par **Hayder HENDI**

Soutenue le **4 décembre 2017**

En vue de l'obtention du grade de docteur de l'ULCO

Discipline **Informatique**

Spécialité **web sémantique**

Titre de la thèse

Ontologies et Web Sémantique pour une construction évolutive d'applications dédiées à la logistique

Thèse dirigée par Cyril FONLUPT directeur
Mourad BOUNEFFA co-directeur
Adeel AHMAD co-directeur

Composition du jury

<i>Président du jury</i>	Abderrafiaa KOUKAM	professeur à l'UTBM
<i>Rapporteur</i>	Adnan YASSINE	professeur à l'Université Le Havre Normandie
<i>Examineur</i>	Dominique LECLLET-GROUX	MCF HDR à l'UPJV-Amiens
<i>Directeurs de thèse</i>	Cyril FONLUPT	professeur à l'ULCO
	Mourad BOUNEFFA	Maître de Conférences à l'ULCO
	Adeel AHMAD	Maître de Conférences à l'ULCO



Université Lille Nord de France
Pôle de Recherche
et d'Enseignement Supérieur



ULCO

Doctoral School ED Régionale SPI 72
University Department LISIC

Thesis defended by **Hayder HENDI**
Defended on **4th December, 2017**
In order to become Doctor from ULCO
Academic Field **Informatics**
Speciality **Semantic Web**

Thesis Title

Ontologies and Semantic Web for an evolutive development of logistic applications

Thesis supervised by Cyril FONLUPT Supervisor
Mourad BOUNEFFA Co-Supervisor
Adeel AHMAD Co-Supervisor

Committee members

<i>Committee President</i>	Abderrafiaa KOUKAM	Professor at UTBM
<i>Referee</i>	Adnan YASSINE	Professor at Université Le Havre Normandie
<i>Examiner</i>	Dominique LECLET-GROUX	HDR Lecturer at UPJV-Amiens
<i>Supervisors</i>	Cyril FONLUPT	Professor at ULCO
	Mourad BOUNEFFA	Associate Professor at ULCO
	Adeel AHMAD	Associate Professor at ULCO

L'ULCO n'entend donner aucune approbation ni improbation aux opinions émises dans les thèses : ces opinions devront être considérées comme propres à leurs auteurs.

Mots clés : web sémantique, ontology, logistique, optimisation, web services, owl, description logique, problème d'acheminement de véhicule, problème de train de passagers, problème de terminal de conteneur

Keywords: semantic web, ontology, logistic, optimization, web services, owl, description logic, vehicle routing problem, passenger train problem, container terminal problem

Cette thèse a été préparée au

LISIC

Le Laboratoire d'Informatique Signal et Image
de la Côte d'Opale

Maison de la Recherche Blaise Pascal

50, rue Ferdinand Buisson

CS 80699

62228 Calais Cedex

France

☎ +33 (0)3 21 46 36 53

📠 +33 (0)3 21 46 55 75

✉ secretariat@lisic.univ-littoral.fr

Site <http://www-lisic.univ-littoral.fr/>



*Je dédie ce travail
à tous ceux qui le méritent*

À mon directeur bien-aimé !

À mon co-directeur bien-co-aimé !

À ma famille bien-moi aider pendant la période d'étude !

Science sans conscience n'est que
ruine de l'âme.

François Rabelais

I can resist everything, except
temptation!

Oscar Wilde

Il est plus facile de désintégrer un
atome qu'un préjugé.

Albert Einstein

ONTOLOGIES ET WEB SÉMANTIQUE POUR UNE CONSTRUCTION ÉVOLUTIVE D'APPLICATIONS DÉDIÉES À LA LOGISTIQUE**Résumé**

Le domaine de la logistique implique souvent la résolution de problèmes combinatoires complexes. Ces derniers font également implicitement référence à des processus, acteurs, activités et méthodes concernant divers aspects qu'il faut considérer. Ainsi, un même problème peut faire intervenir des processus de vente/achat, transport/livraison et gestion de stock. Ces processus sont tellement divers et interconnectés qu'il est difficile pour un logisticien de tous les maîtriser. Dans cette thèse, nous proposons l'explicitation, par le biais d'ontologies, de connaissances conceptuelles et sémantiques concernant les processus logistiques. Ces connaissances explicites sont alors mises à contribution pour construire un système à base de connaissances permettant de guider les logisticiens dans la construction, de façon incrémentale et semi-automatique, de solutions informatiques à un problème qui leur est posé à un moment donné. Nous mettons en oeuvre une ontologie concernant le domaine de logistique connectée à une ontologie associée à la problématique de l'optimisation. Nous établissons ainsi un lien sémantique explicite entre le domaine de la logistique et celui de l'optimisation. Cela permet aux logisticiens d'identifier de façon précise et sans ambiguïté le problème logistique auquel il est confronté et les problèmes d'optimisation associés. L'identification des problèmes conduit alors à un processus de choix des solutions allant du choix du processus logistique précis à mettre en oeuvre à celui de la méthode de résolution du problème combinatoire et cela jusqu'à la découverte du composant informatique à invoquer et qui est matérialisé par un service web. L'approche que nous avons adoptée et mise en oeuvre a été expérimentée avec les problèmes de routage de véhicules, le problème de transport ferroviaire de passagers et le problème de terminaux de conteneurs.

Mots clés : web sémantique, ontology, logistique, optimisation, web services, owl, description logique, problème d'acheminement de véhicule, problème de train de passagers, problème de terminal de conteneur

LISIC

Le Laboratoire d'Informatique Signal et Image de la Côte d'Opale – Maison de la Recherche Blaise Pascal – 50, rue Ferdinand Buisson – CS 80699 – 62228 Calais Cedex – France

ONTOLOGIES AND SEMANTIC WEB FOR AN EVOLUTIVE DEVELOPMENT OF LOGISTIC APPLICATIONS**Abstract**

Logistic problems are often complex combinatorial problems. These may also implicitly refer to the processes, actors, activities, and methods concerning various aspects that need to be considered. Thus the same process may involve the processes of sale/purchase, transport/delivery, and stock management. These processes are so diverse and interconnected that it is difficult for a logistic expert to compete all of them. In this thesis, we propose the explications with the help of ontologies of conceptual and semantic knowledge concerning the logistic processes. This explicit knowledge is then used to develop a reasoning system to guide the logistic expert for an incremental and semi-automatic construction of a software solution to an instantly posed problem. We define an ontology concerning the inter-connected logistics and associated optimization problem. We, henceforth, establish an explicit semantic link between the domains of logistics and the optimization. It may allow the logistic expert to identify precisely and unambiguously the confronted logistic problem and the associated optimization problem. The identification of the problems then leads to a process to choose the solutions ranging from the choice of the precise logistic process to be implemented to that of the method to solve the combinatorial problem until the discovery of the software component to be invoked and which is implemented by a web service. The approach we have adopted and implemented has been experimented with the *Vehicle Routing Problems*, the *Passenger Train Problems*, and the *Container Terminal Problems*.

Keywords: semantic web, ontology, logistic, optimization, web services, owl, description logic, vehicle routing problem, passenger train problem, container terminal problem

Remerciements

Avec joie et bonheur, ma gratitude va vers mes directeurs de thèse, les Professeurs **Cyril FONLUPT**, **Mourad BOUNEFFA**, et **AdeelAHMAD**. C'est avec ces quelques mots que je viens leur témoigner ma satisfaction d'avoir pu collaborer avec eux sur mes travaux de recherche et que grâce à cela, un temps, nous avons été quelque peu réunis. Je leur suis infiniment reconnaissante de m'avoir apporté bien plus que leur soutien, leur professionnalisme et leur savoir-faire. Sans eux, je n'aurais pas pu mener à bien mes recherches et aboutir à un travail concret.

Je remercie vivement mes rapporteurs de thèse, Monsieur les Professeurs **Abderrafiaa KOUKAM** et **Adnan YASSINE** pour le temps qu'ils ont consacré à la lecture ainsi que pour la qualité de leur expertise et de leurs critiques constructives qui ont grandement permis d'améliorer la version finale de ce travail.

Je souhaiterais remercier également tout mes amis les doctorants de Liban et de France pour nos discussions très agréables ainsi que la bonne ambiance qui régnait entre nous et qui a été un facteur de motivation supplémentaire. Notamment, **Ali Darwich**, **Aya Mourad**, **Vinh TRUONG**, **Samah Hijazi**, **Hiba AL Assaad**, **Ahemad Darwish**. Mes chers amis, je vous souhaite beaucoup de réussite dans votre vie professionnelle et familiale.

Surtout, je voudrais bien remercier toute ma famille. Cette thèse n'aurait pas été possible sans le soutien moral de mes parents, mes frères, mes sœurs. C'est grâce à vous ! Je remercie également mes enfants **Ali** et **Ridha** et mon épouse **Sidel**. **Sidel**, je suis très reconnaissante de ton soutien moral qui m'a permis de me lever motivé, le coeur léger et l'esprit tranquille depuis le début de ma thèse. Très humblement, je voudrais te dire merci pour ton soutien pendant mes périodes de doutes et pour tes multiples encouragements répétés.

Table of Contents

Résumé	xvii
Remerciements	xix
Table of Contents	xxi
List of Tables	xxv
List of Figures	xxvii
Introduction	1
I Introduction	5
1 The Concepts of Logistics and Optimization	7
1.1 Introduction of Logistics	8
1.2 The Elements of Logistics	8
1.2.1 Transport management	8
1.2.2 Warehouse management	9
1.2.3 Inventory management	10
1.2.4 Packaging	10
1.2.5 Information processing	10
1.3 Supply Chain Management	10
1.4 The key issues in Supply Chain Management	12
1.4.1 Distribution network issues	13
1.4.2 Warehouse management issues	13
1.4.3 Transportation management issues	13
1.4.4 Inventory control issues	13
1.4.5 Production source issues	13
1.5 The global optimization of SCM	14
1.6 Optimizing the logistics transportation	15

1.7	Optimization concepts versus the logistics problems	17
1.7.1	Continuous Optimization versus Discrete Optimization	17
1.7.2	Unconstrained Optimization versus Constrained Optimiza- tion	18
1.7.3	None, One or Many Objectives	18
1.7.4	Deterministic Optimization versus Stochastic Optimiza- tion	19
1.8	The mathematical formulation of an optimization problem . .	20
1.9	The Optimization Methods	21
1.10	Conclusion	23
2	The Ontologies and Semantic Web	25
2.1	Introduction	25
2.2	The Ontologies	26
2.2.1	Ontologies versus Knowledge Representation	27
2.2.2	The Ontology Application Scenarios	27
	Neutral Authoring	28
	Ontology-based Specification	28
	Common Access to Information	28
	Ontology-based Search	29
2.3	The Description Logic	29
2.3.1	The Definition of the basic Blocks of DL	30
2.3.2	The Description Logic languages	31
2.3.3	The Terminological Knowledge with TBox Axioms	32
2.3.4	Asserting facts with ABox axioms	34
2.3.5	The relationships with RBox axioms	35
2.3.6	Role restrictions	35
2.4	The Semantic Web	37
2.4.1	The Semantic Web Architecture	38
2.4.2	The Resource Description Framework	38
2.4.3	The Resource Description Framework Schema	41
2.4.4	The Ontology Web Language	41
	The OWL Classes	43
	The OWL Properties	44
	The OWL Instances	46
2.4.5	The RDF Query Language (SPARQL)	46
2.5	Conclusion	47
3	Logistics and Optimization Ontologies	49
3.1	Introduction	49
3.2	The logistics and optimization ontologies	50

3.3	The logistics ontologies	50
3.3.1	The Ontologies of Supply Chain Operations Model . . .	50
	The logistics top level ontology	51
	The logistics process type ontology	54
	The logistics process category ontology	54
	The logistics process element ontology	55
3.3.2	The Ontology-Driven Case Based Reasoning System . .	56
3.3.3	Semantic services application framework	57
3.4	The Optimization Ontologies	58
3.4.1	Ontology for Simulation optimization	59
	The Optimization Component	59
	The Optimization Problem	60
	The Optimization Method	60
3.4.2	The Ontology for Optimization	61
3.4.3	The General Optimization Ontology	62
	The structure of General Optimization Ontology	62
3.5	Conclusion	63
 II The Reasoning System		65
 4 The Architecture of the Reasoning System		67
4.1	Introduction	67
4.2	The Reasoning System Architectures	68
4.2.1	The Process Builder	69
4.2.2	The Query Engine	69
4.2.3	The Logistics Ontologies	70
4.2.4	The Optimization Ontology	70
4.2.5	The Artefacts Repository	71
4.2.6	The Web Services	72
4.3	Conclusion	73
 5 The Logistics Ontologies		75
5.1	Introduction	75
5.2	The top level of logistics ontologies	76
5.2.1	The Logistic Resource concepts	78
5.2.2	The Actors	78
5.2.3	The Service Concepts	80
5.2.4	The Services Level Parameter	81
5.2.5	Other Logistic concepts	82
5.3	The Logistic Optimization Problems	84

5.3.1	The Vehicle Routing Problems	86
5.3.2	The Passenger Train Problems	92
5.3.3	The Container Terminal Problems	97
5.4	Conclusion	101
6	The Optimization Ontology	103
6.1	Introduction	103
6.2	Conceptualization of optimization ontologies	104
6.3	The Optimization Components	106
6.4	The Optimization Problems	108
6.5	The Optimization Methods	109
6.6	The Reasoning system for Logistics Problems	114
6.7	Conclusion	117
III	The Implementation	119
7	Implementation prototype of validation	121
7.1	Introduction	121
7.2	The flow chart of Reasoning System	122
7.2.1	Finding the type of logistic problem and its solution method	122
Find the type of Logistics Problems	123	
Find the list of Optimization Methods	128	
7.2.2	Find Optimal Web Services	128
7.3	The prototype of reasoning system	131
The insert and modification part	136	
7.4	Conclusion	143
IV	Conclusion	145
8	The Conclusion and Perspectives	147
8.1	Conclusion	147
8.2	Perspectives	150
	Bibliography	153

List of Tables

- 3.1 The individuals of logistics and optimization ontologies 53
- 7.1 The results of execution web services 131

List of Figures

- 1.1 The logistics elements 9
- 1.2 The Logistics Network [60] 11
- 1.3 Formulation of transportation problem 21

- 2.1 The architecture of knowledge representation based on DL . . . 30
- 2.2 Architecture of semantic web [20] 39

- 3.1 The Top Level of Logistics Ontology 54
- 3.2 The Delivery Process Type Ontology 55
- 3.3 The Deliver stocked product concepts 55
- 3.4 The Metrics Ontology 56
- 3.5 The Process Element Ontology 56
- 3.6 The ontology of SCM project 57
- 3.7 Semantic services application framework 58
- 3.8 Semantics representation of SSAF 59
- 3.9 The SoPt ontology 60
- 3.10 The ONTOP ontology 61
- 3.11 The problem model 63
- 3.12 The solution methods 64

- 4.1 The architecture of the Reasoning System 68

- 5.1 The top level of supply chain management 77
- 5.2 The Logistics Resources 78
- 5.3 The actor concepts 79
- 5.4 The Logistics Company 80
- 5.5 The Services 81
- 5.6 The Metrics Concepts 82
- 5.7 The Goods Concepts 83
- 5.8 The Location concepts 83
- 5.9 The Logistics Problems 84
- 5.10 The Logistics Ontologies to define VRP Problem 88

5.11	The VRP Problem concepts path Problem	88
5.12	The Top level of Passenger Train Problems	93
5.13	The graph of Passenger Train Problems	94
5.14	The top level concepts of Container Terminal Problems	98
6.1	The top level Optimization Ontology	105
6.2	The Optimization Classes	106
6.3	The Optimization Problems	107
6.4	The type of Optimization Methods	109
6.5	The type of Exact Methods	111
6.6	The type of Approximate Methods	111
6.7	The type of Heuristics Methods	112
6.8	The type of Improvement Methods	113
6.9	The type of Meta-Heuristics Methods	113
6.10	Ontology classes of the Capacity vehicle Routing Problem	115
6.11	Ontology classes for the Rolling stock Problem	116
7.1	The flow chart to identify type of problem and solving method	122
7.2	An Example uses DFS to identify Vehicle Routing Problem	124
7.3	The resulted RCPs of Vehicle Routing Problem	127
7.4	An Example of CPDB	128
7.5	The flow chart to find web services	130
7.6	The home page of the reasoning system	132
7.7	The interface that are shown the Resources Concepts	133
7.8	Define the attributes of VRP	133
7.9	Find list of Optimization Methods	134
7.10	Display available list of Web Services	135
7.11	Display total distance of Web Service	135
7.12	Display the results of a called web service	136
7.13	Display main page of Modification Part	137
7.14	Add new type of logistics problem	137
7.15	Define the Concepts of logistics problem	138
7.16	Define the Objective Function	139
7.17	Define the attributes of logistics problem	140
7.18	Display the Individuals of VRP	140
7.19	The interface to specify the type of logistics Problem	141
7.20	Display the individuals of VRP to add new optimization method	141
7.21	Display list of optimization methods to solve the CVRP	142
7.22	Add new Web Service to solve a logistics problem	143

Listings

- 2.1 RDF code snip 39
- 2.2 An Example of the syntax of the OWL 42
- 2.3 An Example of the sameAs syntax 46
- 2.4 An Example of the All Different syntax 46

Introduction

The Logistics Processes involve different types of flows such as physical flows, financial documents, products, etc. It deals with multiple systems on different geographical areas. The logistics processes are a network of activities of manufacturers, suppliers, transporters, and customers. Each of these actors perform different activities to achieve their well established goals. Sometimes, their activities and time schedules may generate difficult situations. There may also emerge situations with conflicting objectives.

The storage of such diverse data demand to define a common vocabulary for its better understanding. It is for this reason, we propose an ontology based approach to share the different aspects of logistics. We define the ontologies for the logistic problems, which are further extended to associate the optimization methods that resolve these problems. It is therefore, we propose to develop a reasoning system capable to manage the logistic and optimization ontology repositories. It allows a logistic expert to interact with the logistic repositories with the help of queries.

The main objective ofn this thesis, is to provide the logistic experts a tool that can assist them in the understanding, implementation, and evolution of different processes constituting their "business".

We attempt to treat the unclearness of the various logistic processes. Therefore, we propose to use the ontologies and semantic web to reduce the ambiguity among logistic processes. The logistics expert need to know the exact type of logistic problem to seek an optimal solution. Hence, we explore the conceptual domain of logistics and optimization, to built the ontology repositories. We define the various logistic problems along with optimization methods to solve them. These solution methods can be further associated with the existing

software components (web services) that implement these methods.

A logistics expert can identify her problem by specifying its characteristics. (S)he describe the language elements (vocabulary) used in her domain. The reasoning system may assist him to specify the vocabulary to identify the problem. It may then propose the optimization methods, that can be used to solve the identified problem. We use semantic modeling to establish these associations. The reasoning system leave the liberty to logistic expert to choose a proposed optimization method and it subsequently assist the logistic expert to transfer the problem data to a web-service that implements the selected method.

We explain the semantic model on three levels of abstraction: the general level, logistic problem level, and the process level. The general level defines the concepts of logistic. The logistic problem levels defines the concepts regarding different characteristics of logistic problems. The process level defines the associations between logistic problem and the optimization concepts.

We develop an Ontology Based Reasoning System that uses the conceptual domain of logistics and optimization. The global architecture of Reasoning System includes the *Process builder*, the *Query Engine*, the *Artifacts Repository*, the *Ontologies repositories*, and the web services.

The core of reasoning system is ontologies. We use the Description logic languages to analyze and define the concepts, relationships (roles), and the axioms. We benefit from the semantic web techniques to develop and extract the knowledge in the ontologies.

We initially define the concepts and roles of some general logistic problems to validate the proposed approach. These problems are *Vehicle Routing Problem (VRP)*, the *Passenger Train Problems (PTP)*, and the *Container terminal Problems (CTP)*. We then define the axioms to identify the exact variant of logistic problems.

The manuscript in this thesis is organized as follows:

The chapter 1 discusses the principles of logistics and optimization technique. It purposes a layer based representation of key elements on these two areas, particularly those which were the subject of our interest in the context of this thesis. The chapter 2 describes the main conceptual tool; we define the notion of ontologies. We discuss the main structures to implement ontologies and

applications of ontologies, the most famous among them are the semantic web. The chapter 3 describes the ontologies in relevance to their application in fields of logistics and optimization. The chapter 4 describes the global architecture of ontology based reasoning system. The chapters 5 and 6 describe the ontologies that we develop for the conceptual descriptions of logistics and optimization, respectively. The chapter 7 discusses the general flow processes of the ontology based reasoning system. We discuss its implementation details, and show, how the logistic expert can interact with it. The chapter 8 concludes our work and discuss the results, weaknesses, and prospects.

Part I

Introduction

The Concepts of Logistics and Optimization

In this chapter, we review the major principles of logistics and the optimization techniques, we also discuss the relationships among the different logistic concepts such as the relationships between the logistics processes and the optimization techniques and so on their dependencies on the algorithms which implements them. We can then broadly classify this study in two parts.

In the first part, we discuss the general concepts of logistic and main logistics elements. Initially, we consider the Supply Chain Management (SCM) and its key issues. We, henceforth, discuss the difference between the logistics Management and the Supply Chain Management. We, later on, explain the global optimization of Supply Chain Management, and finally, we specify the logistics transport with the help of our established vocabulary.

During the second part, we mainly focus to review the optimization concepts to further optimize the logistics problems. In this study we explore the involved optimization's problems and methods (adopted for logistics). Therefore, we classify these concepts and discuss the Mathematical Formulation of optimization. We illustrate this study with the help of the examples taken from the transportation problems and formulate the concerned optimization methods.

1.1 Introduction of Logistics

The term "Logistics" originates from the ancient Greek word. Generally, logistics may be defined as "the processes of planning, implementing, and controlling the efficient, effective flow and storage of goods, services, and related information from point of origin to point of consumption for the purpose of conforming to customer requirements" [42]. It has been often used in the context of the current business logistics. It is also used in the military discipline, as means of the supply of necessary arms, ammunition and rations, whenever they are needed. For example when the arrange has to move from it's own base to a forward position.

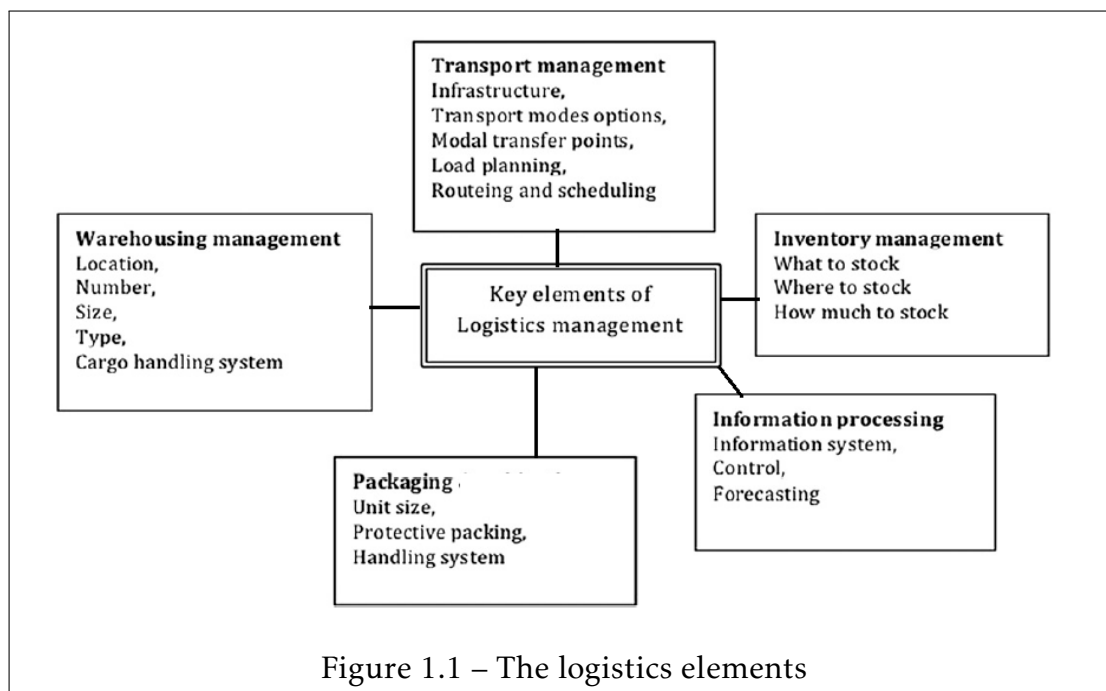
The business logistics originally concerns "The management related to the positioning of resources". The logistics may be a branch of engineering that deals with the "people based systems" rather than "machine based systems". The modern logistic concepts and practices lead to provide cost and time effective services not only for military but also for the non-military organizations, mainly having the commercial activities. The logistic, in this regard, provides multiple services, like transport of goods from one point to another, warehousing them in a suitable place, inventory control, packaging, and other administrative activities such as order processing, etc. [31].

1.2 The Elements of Logistics

The Figure 1.1 shows the general elements of logistics. The logistics may include five key elements such as the management of transport, warehouses, inventories, packaging and information processing. We briefly discuss each one of them, in the following.

1.2.1 Transport management

Transport is among the major elements of most logistics services. The significant components of transport management may involve :



- the modes of transport including the road, rail, waterways, air, pipeline; either multimodal or intermodal.
- the transport infrastructure according to the geographical conditions and the type of delivery such as overnight express, normal, long distance, etc.
- the load planning (in the cargo unit), scheduling, and routing.

1.2.2 Warehouse management

The warehouse management may implicate the number of warehouses and their different locations. Which means the warehousing policy, is it central (involving one warehouse) or it is conceptually decentralized (involving multiple warehouses, distributed on different locations). It refers also to the size of warehouse (s), type of storage such as for refrigerated cargo, electronics, garments, and eventually the material handling equipment [31].

1.2.3 Inventory management

The inventory management may include strategical decisions such as what to stock, how much to stock, and where to stock? Inventory management is different from the warehouse management, because inventory management is mainly concerned with the amount of stock (either of the products or of raw material) whereas the warehouse management deals with the housing aspect of this stock [31].

1.2.4 Packaging

The packaging is also among the important elements of logistics. It concerns each product at its different levels such as raw material, semi-finished, finished. The cost of packaging may depend on the type of product. For example, for higher value goods, the packaging cost can also be more whereas for the raw material the cost has to be relatively less [31].

1.2.5 Information processing

Information processing provides a liaison to put together the whole logistic system. The growth of reasonably priced hardware and software has put sophisticated management information systems within the reach of even the smallest organization. Indeed firms are now linking their internal logistics information systems with those of their vendors and customers as a means of adding more value to the entire chain. Such an open exchange of information can result in faster order placement, quicker benefit delivery, and greater accountability throughout the logistics process [31].

1.3 Supply Chain Management

The Supply Chain Management (SCM) is a very diversified domain. It has never been simple to decide what to include or exclude in the SCM and its constituents. Its subsequent activities can be different even among the people involved in the same processing chain. Essentially, it involve the activities related

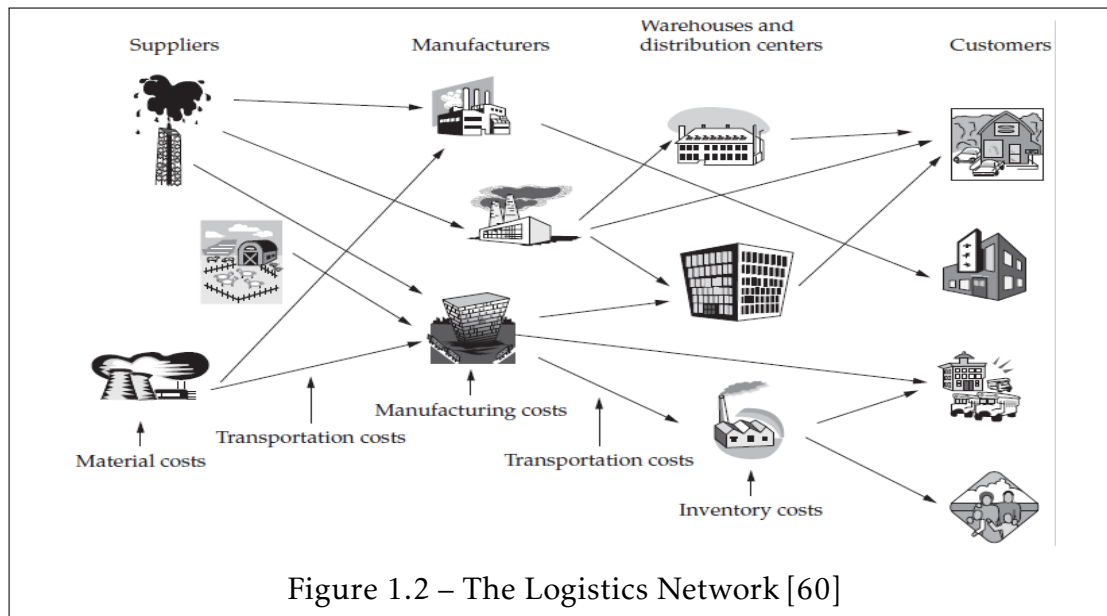


Figure 1.2 – The Logistics Network [60]

to the flow and transformation of goods; from the raw materials to the finished products; and also from the manufacturing until the delivery to the end user.

The SCM can also be considered as a set of approaches used to efficiently integrate suppliers, manufacturers, warehouses, and stores. So that the merchandise is produced and distributed at the right quantities, to the right locations, and at the right time, in order to minimize system wide costs while satisfying service level requirements [60].

The SCM study may lead to several observations. We may refer to the Figure 1.2 to further explain the complex interactions among different logistic elements. It may consider each effective facility related to the cost and the roles involved in the making of the product, then its delivery to the customers. It may further consider the supplier locations, manufacturing facilities through warehouses, and later on deliveries towards the distribution centers, retailers, and stores.

The general objective of SCM is to be efficient and cost-effective across the entire the system. It asks to minimize the total system wide costs including the distribution of inventories of raw materials, goods progress, finished products and eventually their transportation. Thus, the emphasis is not on simply minimizing the transportation cost or reducing inventories but to address a

formalized approach for entire supply chain management.

In fact the Logistics Management is a sub-set of SCM that persuade primarily the efficient management of goods. Although, in the current study, we consider the SCM in its broader terms which refers the connections from the manufacturers (through suppliers and transporters) to the consumers [62].

An efficient SCM premise upon an effective integration of customers, manufacturers, and suppliers through the different storage locations. It urges to undergo through established activities at many levels i.e. from the strategic level (through the tactical) to the operational level [60].

1.4 The key issues in Supply Chain Management

The debatable issues of supply chain management can be extended to the set of core activities of a company. It may instigate the activities on different strategic and decision levels, as discussed below :

- The issues, involved at the strategic level, can have a long-lasting effect on the company. The depending decisions may concern the product design; what to make internally and what to outsource. It might concern supplier selection and the strategic partnering. Furthermore, it may concern the decisions regarding the number, location, and capacity of warehouses, manufacturing plants, and the flow of material through the logistics network.
- The issues, involved at the tactical level, can effect the periodic decisions. Mainly, they require the updates on different time intervals, for instance at every quarter or every year, etc. The decisions may concern the purchasing and production at fixed time periods, inventory policies, and transportation strategies.
- The issues, at operational level, concerns the daily (routine) decisions such as scheduling, lead time quotations, routing, and truck loading, etc.

In the following, we discuss some of the key issues, questions, and trade-off associated with different decision types.

1.4.1 Distribution network issues

The Distribution Network Configuration deals with several factories (producing products) for a set of retailers, that are distributed on different geographical areas. So, which set of warehouses is appropriate? and what management needs to be reorganized or redesigned in the distribution network?

1.4.2 Warehouse management issues

Any change in warehouses management can effect the production and then the factories and transportation, etc. Their management depends on production levels for each product at each factory. It requires to know, how many warehouses are used? and what are their locations and capacities?.

1.4.3 Transportation management issues

These concern the management of the set of transportation fleets between facilities, either from factory to warehouse or from warehouse to retailer, in order to reduce the total cost of production, inventory, and transportation to further satisfy the service level requirements.

1.4.4 Inventory control issues

Inventory control consider the retailers that maintains an inventory of a particular product. Whereas the customers demand changes over time, the retailer can use only historical data to predict demand. The retailer's objective is to decide at what point to record a new batch of the product, and how much to order so as to minimize the inventory ordering and holding costs. Such as why should the retailer hold inventory in the first place?

1.4.5 Production source issues

Production Sourcing may include many source industries that provide the basic ingredients to manufacture the product. These need to find a balance between transportation and manufacturing costs. In particular these aim to

reduce the production costs. It typically implies that each manufacturing facility is responsible for a small set of products so that large batches are produced, hence reducing the production costs.

These issues are not necessarily independent of each other. There are not only many common causes but also the homogeneous problems among different issue sources. Their management is a complex optimization problem. The comprehension of this heterogeneity is further complicated while we need to find optimal solutions. For example, production source issues may lead to higher transportation costs. Similarly, reducing the transportation costs could imply that each facility is flexible and has the ability to produce most or all products, but this leads to small batches and hence increases production costs. Finding the right balance between the two cost components is difficult but needs to be done monthly or quarterly.

1.5 The global optimization of SCM

Optimization of Supply Change Management issues is a significant challenge, even at an abstract level. In this case, building the best system-wide or with globally optimal integrated solution is a difficult task. A variety of factors contribute to this challenging problem :

1. The supply chain has many facilities distributed over a large geography.
2. The different facilities have different conflicting objectives such as the suppliers and manufacturers want to produce consistently large quantities with flexible delivery dates while distributors also wants the same usually on urgent demands. Unfortunately, although most manufacturers would like to implement long production runs, they need to be flexible to their customers' needs and changing demands. Thus, the suppliers goals are in direct conflict with the manufacturers desire of flexibility [60].

Indeed, the production decisions are typically made without precise information about customer demand. The ability of manufacturers to match supply and demand depends largely on their ability to change supply volume as information about demand arrives. Similarly, the manufacturers

objective of making large production batches typically conflicts with the objectives of both warehouses and distribution centers to reduce inventory. To make matters worse, this latter objective of reducing inventory levels typically implies an increase of the transportation costs [60].

3. The supply chain must be a dynamic system that evolves over time. This means not only do customer demand and supplier capabilities change over time, but supply chain relationships also evolve over time. For example, as customers power increases, there is increased pressure placed on manufacturers and suppliers to produce an enormous variety of high-quality products and, ultimately, to produce customized products [60].
4. System variations over time are also an important consideration. Even when demand is known precisely (e.g., because of contractual agreements), the planning process needs to account for demand and cost parameters varying over time due to the impact of seasonal fluctuations, trends, advertising and promotions, competitors pricing strategies, and so forth. These time-varying demand and cost parameters make it difficult to determine the most effective supply chain strategy, the one that minimizes system wide costs and conforms to customer requirements [60].

The global optimization of SCM implies the significance of optimization not only across the supply chain facilities, but also across all the processes associated with the development and supply chains. That mean, it is important to identify processes and strategies that optimize, alternatively, synchronize, both chains simultaneously [60].

1.6 Optimizing the logistics transportation

The transportation is among the most important elements of logistics and supply chain management. It deals, on the one hand, with the number of costumers who need some kind of goods, and on the other hand, with the number of dispatchers (e.g. warehouses or producers) who provide these goods. The transportation is the task for organizing such shipping or transport. It primarily requires to gather the information about the possible means of transport, the

costs for using different vehicles or for renting and maintaining additional warehouses/hubs [65]. When this information is available, it further investigates the number of possible ways to organize the traffic. The significant factors in such choices can be as follows :

- Which dispatcher will deliver goods to which customers ?
- In what quantity the goods are dispatched ?
- What is the capacity (maximum truck load) of dispatcher during a transportation ?
- Do we transport our goods directly from one destination to the other or do we consolidate them somewhere else ?

Our choice must not rely solely on chance or some very general considerations, but it should be optimized in some way. Optimization targets may be different, we could opt for minimizing cost, minimizing time (or some kind of delay), minimizing environmental impacts, etc [31].

Even if we decide to optimize the overall cost, this does not mean that we ignore the other possible optimization targets. A delivery which takes a long time may cost a lot of money, and furthermore, usual time restrictions must be applied. Restrictions or constraints are generally an important concept to keep in mind. As normally, the aim, to reduce the costs, is limited by factors like working hours or maximum truck loads which have to be checked for every "solution" that come up with. Otherwise, it would be a risky choice and might not be feasible.

Up to this point, the following should have been done :

1. Gather all relevant information.
2. Decide what must be optimized.
3. Look for possible restrictions or constraints of the problem.

The mathematical model has to be formulated to accurately represent the problem. It has two major advantages:

- The model and also the optimization question becomes very precise. This allows a formal finding of a solution, to determine whether it is feasible or not.
- It permits the implementation of the solution by using the executable

algorithm to solve the problem. It means a solution is attempted to be found and improved in each step by searching the strategies from which the understanding of the problem can be derived.

The algorithm should be short and compatible for small instances to be executed reasonable time. The solution should be presentable in a table or in form of a graph in a human comprehensible way. It can be helpful, if the strength of the algorithm can be compared to a naive solution, or to the other algorithms along with the optimal solutions. At least one of the algorithms presented should be strong enough to give nearly optimal solutions to the example problems [31].

1.7 Optimization concepts versus the logistics problems

The optimization is an important tool in decision science for the analysis of physical and logistics systems. In order to use this tool, some objectives that are quantitative measures of the performance of the system, must be identified. These objectives could be profit, time, potential energy, or these can also be a combination of quantities that can be represented by a single number.

The objective depends on certain characteristics of the system, called variables. The goal is to find values of the variables that give the optimal value for the objective. Often the variables are restricted, or constrained.

The optimization may consider the certain optimization problem and optimization methods (algorithms). Among the classification of optimization methods, some can be adopted in the logistics domain. In the following, we discuss some of those which can provide the guidance to better classify both the logistics optimization problems and methods [47].

1.7.1 Continuous Optimization versus Discrete Optimization

Some models make sense if the variables take values from a discrete set, often a subset of integers, whereas other models contain variables that take real value. Models with discrete variables are called discrete optimization problems, while

the models with continuous variables are called continuous optimization problems [48]. We can classify the optimization methods (depending on optimization problem) into Continuous Optimization Methods and Discrete Optimization Methods. Continuous optimization problems tend to be easier to solve than discrete optimization problems. The smoothness of the functions means that the objective function and constraint function values at a point x can be used to deduce information about points in a neighbourhood of x [18].

1.7.2 Unconstrained Optimization versus Constrained Optimization

Another distinction is between problems with or without constraint variables. Unconstrained optimization problems arise directly in many practical applications, they also arise in the reformulation of constrained optimization problems in which the constraints are replaced by a penalty term in the objective function. Constrained optimization problems arise from applications in which there are explicit constraints on the variables. The constraints on the variables can vary widely from simple bounds to systems of equalities and inequalities that model complex relationships among the variables. Constrained optimization problems can be further classified according to the nature of the constraints (e.g. linear, nonlinear, convex) and the smoothness of the functions (e.g. differentiable and non differentiable) [18].

1.7.3 None, One or Many Objectives

The majority of optimization problems have a single objective function. There exist some cases with multiple objective functions while there also exist some cases with no objective function. Multi-objective optimization problems arise in many fields, such as logistics, when optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. While in case of feasibility problems, the goal is to find values for the variables that satisfy the constraints of a model with no particular objective to optimize. Also, the complementarity problems are pervasive in engineering and economics. The

goal is to find a solution that satisfies the complementarity conditions [18].

1.7.4 Deterministic Optimization versus Stochastic Optimization

The deterministic optimization assumes that the data for the given problem are known accurately. However, for many actual problems, the data cannot be known accurately for a variety of reasons. The first reason is due to simple measurement error. The second and more fundamental reason is that some data represent information about the future, for example product demand or price for a future time period, and simply cannot be known with certainty. The uncertainty is incorporated into the model. Robust optimization techniques can be used when the parameters are known only within certain bounds, the goal is to find a solution that is feasible for all data and optimal in some sense [18].

Stochastic programming models take advantage of the fact that probability distributions governing the data are known or can be estimated. The goal is to find some policies that are feasible for all (or at least for one) of the possible data instances and optimizes the expected performance of the model.

Usually, we can choose among the many optimization methods to solve an optimization problem. They exist mostly in form of algorithms (implemented in some programming language or integrated in some software tool). Still, there isn't a universal optimization method but rather a collection of algorithms, each one can be tailored to a particular type of optimization problem [47].

The user may avail the liberty to choose a suitable method. This choice is important, as it may determine whether the problem would be solved rapidly or slowly and, whether the solution would be found at all.

After the application of an optimization model. We must be able to identify the fact that the solution has been found or not. In many cases, there are mathematical expressions known as optimal conditions. These can be used to test whether the current set of variables can provide the solution to the problem. If the optimal conditions are not satisfied, they may give useful information on how the current estimate of the solution can be improved. The model can then be improved by applying techniques such as sensitivity analysis, which reveals the

sensitivity of the solution to changes in the model and data. The interpretation of the solution in terms of the application may also suggest different ways in which the model can be refined or improved (or corrected) [47].

1.8 The mathematical formulation of an optimization problem

The mathematical formulation of optimization is written considering the minimum or the maximum values of the objective functions, depending on constrained variables.

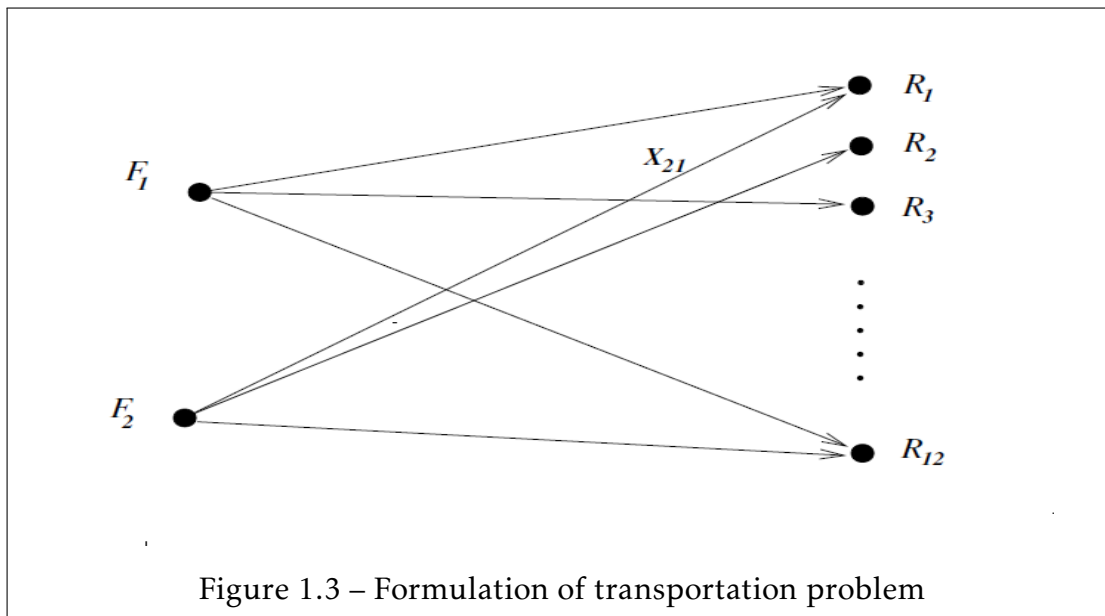
In the following, we discuss some of the notations used in our formulations :

- X is the vector of variables, also called, unknowns or parameters.
- OF is the objective function, a (scalar) function of X that we wish to maximize or minimize.
- c_i are constraint functions, which are objective functions of X . It define certain equations and inequalities to satisfy the unknown vector X .

There exit many optimization formulation, for instance, let us take an example of transportation problem for illustration purpose. It may arise in manufacturing and transportation as shown in Figure 1.3.

It primarily concerns a chemical company that has 2 factories denoted as F_i , where $i \in \{1, 2\}$, and a dozen retail outlets denoted as R_j where $j \in \{1, \dots, 12\}$. Each factory F_i can produce a_i tons of a certain chemical product each week, a_i is called the capacity of the factory. Each retail outlet R_j has a known weekly demand denoted as b_j (weekly demand of outlet j) in tons of the product. The cost of shipping for one ton of the product from factory F_i to retail outlet R_j is c_{ij} .

The problem is to determine, how much of the product to ship from each factory to each outlet so as to satisfy all the requirements and minimize the cost. The variables of the problem are X_{ij} , $i \in \{1, 2\}$ and $j \in \{1, \dots, 12\}$, where X_{ij} is the number of tons of the product shipped from factory F_i to retail outlet R_j . We can define the problem as follows :



$$OF = \min \sum c_{ij} X_{ij} \quad (1.1)$$

$$\sum_{j=1}^{12} X_{ij} \leq a_i, \quad i \in \{1, 2\} \quad (1.2)$$

$$\sum_{i=1}^2 X_{ij} \geq b_j, \quad j \in \{1, \dots, 12\} \quad (1.3)$$

$$X_{ij} \geq 0, \quad i \in \{1, 2\}, \quad j \in \{1, \dots, 12\} \quad (1.4)$$

This type of problem is known as a linear problem, since the objective function and the constraints are all linear functions. In a more practical model, we can also include costs associated with manufacturing and storing the product [47].

1.9 The Optimization Methods

Optimization is the act of achieving the best possible result under given circumstances. The goal of all such decisions is either to minimize the effort or

to maximize the benefit. The effort or the benefit can be usually expressed as a function of variables.

Hence, optimization method is the process of finding the conditions that give the maximum or the minimum value of a function. It is obvious that a point x may either correspond to the minimum value of a function f_x , or the point x may correspond to the maximum value of the function f_x . There is no single method available to solve all the optimization problems efficiently. Hence, a number of methods have been developed to solve the different types of problems.

The optimization methods are also known as mathematical programming techniques, which are a branch of Operations Research (OR). The operations research can be composed of the following areas:

- Mathematical programming methods: these are useful in finding the minimum of a function of several variables under a prescribed set of constraints.
- Stochastic process techniques: these are used to analyze problems which are described by a set of random variables of known distribution.
- Statistical methods: these are used in the analysis of experimental data and in the construction of empirical models.

The Mathematical programming is a vast area of mathematics and engineering, it can include, among others, the followings:

- calculus of variations and optimal control,
- linear, quadratic and non-linear programming,
- geometric programming,
- integer programming,
- network methods (PERT),
- game theory.

The stochastic methods are partly random input data arising in such areas as real-time estimation and control, simulation-based optimization where Monte Carlo methods run estimates of an actual system and problems where there is experimental (random) error in the measurements of the criterion[9].

In these cases, the knowledge about the function values is contaminated by random "noise". It leads naturally to algorithms. We can use statistical inference tools to estimate the "true" values of the function and/or make statistically

optimal decisions about the next steps. These methods can be classified into :

- stochastic approximation (SA),
- stochastic gradient descent,
- finite-difference SA,
- simultaneous perturbation,
- scenario optimization.

1.10 Conclusion

In this chapter, we discuss the principle of Logistics and the Optimization. In fact, it is necessary to define clearly the general concepts of these two domains. We briefly review the basic elements of logistics and their correspondences that can mitigate the different logistic issues. We later on claim that the logistics problems are most often the optimization problems. These may require one or many objective functions to be satisfied. It is for this reason, we proceed with the mathematical formulation of these problems and propose logistics and optimization ontologies to better understand and solve them. For instance, we choose the transportation problem, among other supply chain management issues, for the sake of illustration. In the next chapter, we discuss in detail the ontologies and semantic web technologies in relevance to solve the logistics issues. These concepts are used to further develop the logistics and optimization ontologies as shown in chapter 5 and 6. These ontologies also express relationships between the logistics and optimization concepts.

The Ontologies and Semantic Web

2.1 Introduction

An ontology is the formal specification of a concept and its relationships among other associated terminologies within a system. It bridges the gap of lack of vocabulary in different domains to establish the formal links and enables the knowledge sharing. These can further be made available to a larger public by the use of semantic web technologies. In our work, we initially proceed with the explicit definition of the logistics concepts, and the problems by means of ontologies. We then enable a user access to choose the available solutions with the help of semantic web techniques. These are later on developed in form of a reasoning system that provide the assistance to a logistic expert to define his problem and chose a result. We then integrate a tool set to deal with the defined ontologies, making them available for the user to query his problems and find an optimal solution using the semantic web language and description logic.

In this chapter, we present the definition of the principle ontologies used in our proposed system. We intend to compare the defined ontologies between the knowledge representation and database schema. For this purpose, we use the description Logic which provides the formal foundation of ontologies. Meanwhile, we also present the semantic web tools along with some standard languages used to implement the ontologies.

2.2 The Ontologies

An Ontology can be defined as "a formal, explicit specification of a shared conceptualization". The conceptualization refer to an abstract model representing how people think about things in the real world, usually restricted to a particular subject area. An explicit specification means that the concepts and relationships in the abstract model are given explicit names and definitions [67]. In this regard :

- The definition term is a specification, the meaning of concept or relationship depends on its name.
- Formal means the use of an encoding language to define the specification of the "meaning". The aim is to avoid confusing incoherent definitions. Generally the used languages are logic-based ones. These languages have yet been used for many decades to represent the knowledge in the Artificial Intelligence community.
- Formality is an important way to avoid the ambiguity that is prevalent in natural language and other informal notations. These also opens the door for automated inference to derive new information from the specification meanings.
- Shared means the main purpose of an ontology that is generally to be used and reused across different applications and communities.

There has been much discussion on what exactly counts as an "ontology", however there is a common core that runs through virtually all approaches :

- A vocabulary of terms that refers to the things of interest in a given domain.
- Some specifications of meaning for the terms, ideally grounded in some form of logic.

The Ontologies can represent many different kinds of things. The things can be represented as concepts in the ontology (sometimes called classes) and are typically arranged in taxonomies of classes and sub-classes. At most each class is associated with various properties (also called slots or roles) describing the features and attributes as well as various restrictions on them (sometimes called facets or role restrictions). An ontology can include a set of instances (also called

individuals) of the classes constituting a knowledge base.

2.2.1 Ontologies versus Knowledge Representation

Ontologies are one of the important constituents of semantic-based technologies which include the areas of knowledge representation and automated inference that arise within the Artificial Intelligence community.

In the literature and development community, many different representations for formalism have been explored, and reasoning engines are being continuously developed. The important feature of these technologies is the proven existence of a trade-off between representational power of a language (i.e. the ability to represent/express many different kinds of knowledge) and the efficiency of the reasoning engines.

Nowadays, the ontologies have become a sub-area of the knowledge representation. The knowledge representation languages are used to represent formal ontologies, and standard inference engines are used to make reasoning over ontologies. Similarly, the knowledge bases (KB) can be used to provide a better backbone for the ontology. The major difference between ontologies and KB has been that the ontologies are the fact that focus on knowledge sharing [67].

2.2.2 The Ontology Application Scenarios

The definition of the ontology may be a "shared and common understanding of a domain that can be communicated between people and application systems". We first identify scenarios for the community within the system, called as ontology application scenarios. We, later on, adopt them to show a particular situation of ontologies and then use them for some specific purpose. We discuss in below, the ontology application scenarios considered as four categories of ontology applications. Each one of these categories can identify one or more specific application scenarios. For example, the neutral authoring category may include two scenarios (as described here after) concerning the authoring ontology and authoring operational data [33].

Neutral Authoring

In neutral authoring, the principle is to give the company the possibility of authoring for developing her own neutral ontology. Therefore, translators can be developed from this ontology for the required terminology by the different target systems. Moreover, this scenario allows knowledge reuse, improve maintainability, and long term knowledge. The neutral ontology must include features ensuring no information loss in translation. There fractures may be supported by all of the target systems. The trade-off here is a loss of functionality of some of the tools due to the fact that certain special features may not be usable.

Ontology-based Specification

This scenario can also be called “Ontology-Driven Software Engineering”. The idea is to consider the ontology of special domains as a basis of a software system specification. The idea is to first create an ontology characterizing and specifying a certain domain and use this ontology as a partial set of requirements for building the software.

Common Access to Information

This scenarios has a basic idea, it uses the ontologies through multiple target applications (or humans), accessing the heterogeneous sources of information which are otherwise unintelligible. The advantage of this scenario is the gain of interoperability and knowledge reuse.

The scenario can be categorized into different ways, as follows :

1. The direct consumers of the information may be humans or computer applications.
2. The information artifacts may play the role of an ontology, or operational data. The later may be non-computational (e.g., product data) or computational (e.g., web-services).

The important distinction concerns target applications that accept working on same shared ontology or whether each has its own local ontology. In the former case, the information is made intelligible one *via* translators, and in the

later case, *via* ontology mapping rules. Finally, access to the information may be *via* sharing or exchange [33].

Ontology-based Search

This scenario mostly deals with information facilitating search. The ontology is used as a structuring device for information repository. The ontology can then be used as advanced indexing mechanism into such repositories.

2.3 The Description Logic

Description Logic (DL) are formal languages generally used for knowledge representation and reasoning [4]. The DLs have been used to represent the knowledge representation before the advent of ontologies and Semantics Web. These are standardized by the World Wide Web Consortium (W3C) [37]. These include a family of languages widely used to define ontological models.

The DLs have been used as modeling languages since the mid 1980s. As, the name Description Logic indicates they can be equipped with a formal logic-based semantic. It is one of the characteristics of these languages. Such formal semantic allows humans and systems to exchange the knowledge without ambiguities of its real meaning, and also infer additional information from the facts that depends on the logical deduction in an ontology [37]. The important feature is assertion on reasoning as a central service. The reasoning allows one to infer implicitly represented knowledge from the knowledge contained in the knowledge base [4].

Description Logic supports inference patterns that occur in many applications of intelligent systems, and it is used by humans to describe and understand the world that is classified into concepts and individuals [4].

The maturity of the field is reflected by the adoption of description logic as specification paradigm for ontological descriptions. It is culminating in the standardization of the OWL¹ web ontology language by the World Wide Web Consortium (W3C) and tools for automated inference [56].

1. <https://www.w3.org/OWL/>

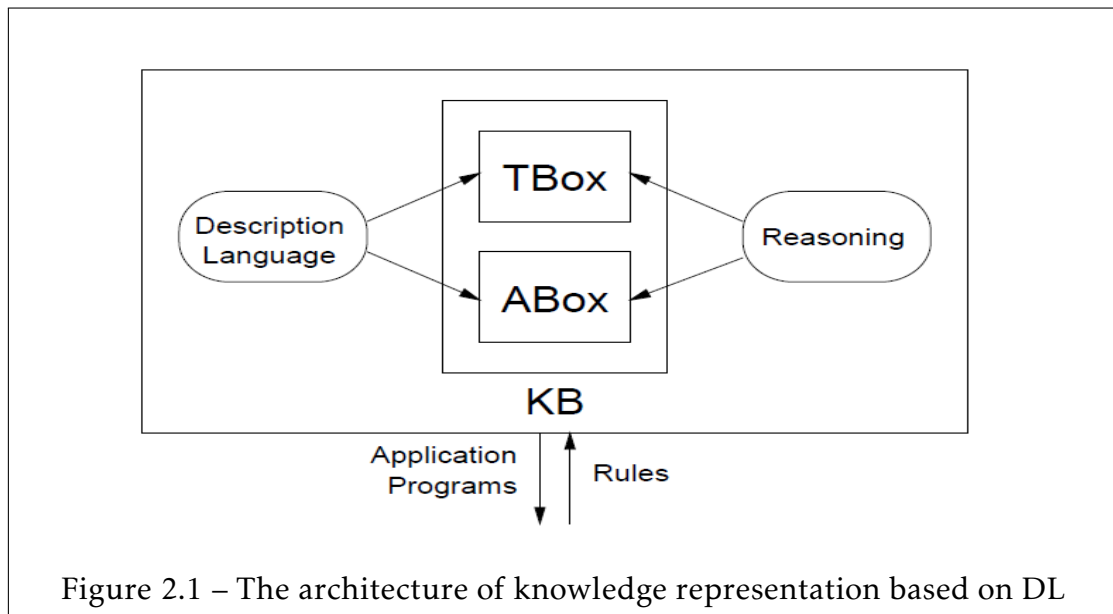


Figure 2.1 – The architecture of knowledge representation based on DL

2.3.1 The Definition of the basic Blocks of DL

The Figure 2.1 describes the architecture of a knowledge representation system based on Description Logic. It provides facilities to set up the knowledge base (KB), to reason about their content, and to manipulate them. The knowledge base is comprised of three parts :

The TBox introduces the terminology, i.e., the concepts of an application domain, while the ABox contains assertions about named individuals in terms of this concepts and Rbox contains roles or predicates representing relations between the individuals [4].

For example, an ontology, modeling the domain of people and their family relationships may use the following :

- Concepts, such as *Person* that is either *Male* or *Female*
- Individuals, such as *john* is a *Male* and *juila* is a *Female*
- Role, such as *Marriage* to represent the relation between individual set i.e., *Marriage(john, juila)*.

The DL ontology does not fully describe a particular situation. It consists of a set of statements called axioms, each of which must be true in the described situation. These axioms typically capture only partial knowledge about the

situation that the ontology is describing, and there may be many different states of the world that are consistent with the ontology [37].

2.3.2 The Description Logic languages

The general elements of DL languages can be specified as atomic concepts and roles. The complex descriptions can be built from them inductively with concept constructors.

Let us use the letters A and B to denote atomic concepts and the letter R for atomic roles. Let us also use letters C and D to denote concept constructors. The Description languages are distinguished by the constructors they provide. In this section, we discuss various languages from the family of \mathcal{AL} -languages (*Attributive Languages*).

The language \mathcal{AL} has been introduced in 1991 by Schmidt as a minimal language [4]. It is of practical interest. There are other languages of this family defined as extensions of \mathcal{AL} .

The syntax of concept descriptions in \mathcal{AL} can be formed according to the following :

$$\begin{aligned}
 C, D &\rightarrow A \mid (\textit{AtomicConcept}) \\
 &\quad \top \mid (\textit{UniversalConcept}) \\
 &\quad \perp \mid (\textit{BottomConcept}) \\
 &\quad \neg A \mid (\textit{AtomicNegation}) \\
 &\quad C \cap D \mid (\textit{Intersection}) \\
 &\quad \forall R.C \mid (\textit{ValueRestriction}) \\
 &\quad \exists R.\top \mid (\textit{LimitedExistentialQuantification}).
 \end{aligned}$$

The negation \mathcal{AL} may be applied to atomic concepts. Only the top concept is allowed in the scope of an existential quantification over a role. The formal semantic \mathcal{AL} -concepts can be specified by interpretation I that consist of a non-empty set Δ^I (the domain of the interpretation) and an interpretation function. The interpretation function assigns to every atomic concept A a set $A^I \subseteq \Delta^I$ and

to every atomic role R a binary relation $R^I \subseteq \Delta^I \times \Delta^I$. The interpretation function is extended to concept descriptions by the following inductive definitions:

$$\begin{aligned} \top^I &= \Delta^I \\ \perp^I &= \phi \\ (\neg A)^I &= \Delta^I \setminus A^I \\ (C \cap D)^I &= C^I \cap D^I \\ (\forall R.C)^I &= \{a \in \Delta^I \mid \forall b. (a, b) \in R^I \rightarrow b \in C^I\} \\ \exists R.\top^I &= \{a \in \Delta^I \mid \exists b. (a, b) \in R^I\} \end{aligned}$$

Let us consider, the two concepts C and D are equivalent, we then write $C \equiv D$, if $C^I = D^I$ for all interpretations I . For instance, going back to the definition of the semantic of concepts, one can easily verify that $\forall hasChild.Female \cap \forall hasChild.Student$ and $\forall hasChild.(Female \cap Student)$ are equivalent[4].

2.3.3 The Terminological Knowledge with TBox Axioms

TBox axioms describe relationships between concepts. For example, the fact that all females are person is expressed by the concept inclusion :

$$Female \sqsubseteq Person \tag{2.1}$$

In this case, we can say that the concept *Female* is subsumed by the concept *Person*. Such knowledge can be used to infer further facts about individuals.

Therefore, the equivalence concepts must have the same instances. Boolean concept constructors provide basic Boolean operations. The conjunction, disjunction, and negation of logical expressions are closely related to the familiar operations of intersection, union, and complement of sets [37].

For example, the *Mother* concept is subsuming of *Female* because all mothers are females. Also that all *Mother* is a *Parent*, but what we really mean is that mothers are exactly the *Female Parent*. The DL support such statements $Female \cap Parent$ by allowing us to form complex concepts such as the *intersection* (also called conjunction), it represents the set of individuals that are both

Female and *Parent*. A complex concept can be used in axioms in exactly the same way as an atomic concept, e.g., in the equivalence:

$$Mother \equiv Female \cap Parent \quad (2.2)$$

Union (also called disjunction) is the dual of intersection. For example, the concept $Mother \cup Father$ describes those individuals that are either *Father* or *Mother*. Again, it can be used in an axiom such as

$$Parent \equiv Father \cup Mother \quad (2.3)$$

This states a *Parent* as either a *Father* or a *Mother* (and *vice versa*). Sometimes we are interested in individuals that do not belong to a certain concept, e.g., the *Women* who are not *Married*. These could be described by the complex concept:

$$Female \cap \neg Married \quad (2.4)$$

Where *the complement* (also called negation) $\neg Married$ represents the set of all individuals that are not married. It can be useful to be able to make a statement about every individual, e.g., to say that everybody is either a *Male* or a *Female*. This can be accomplished by the axiom:

$$\top \subseteq Female \cup Male \quad (2.5)$$

Where, the *top concept* \top is a special concept with every individual as an instance, it can be absolutely viewed as a union for the concept and his complement such as $C \cup \neg C$.

According to this modeling nobody can be both a *Male* and a *Female* at the same time. We can declare the set of male and the set of female individuals to be disjoint[37].

$$Female \cap Male \subseteq \perp \quad (2.6)$$

Where, the *bottom concept* \perp is inverse of \top , it means no individual can be a instance of both. This can be shown, as follows:

$$\perp \cap \top = \phi \quad (2.7)$$

The additional concept expressions, combined with the equivalence and inclusion, can describe the complex situations such as the analysis of concept, which asserts that two concepts do not share any instances.

2.3.4 Asserting facts with ABox axioms

The ABox axioms capture knowledge about named individuals. The most common ABox axioms are concept assertions such as *Julia* is a *Mother* or, more precisely, the individual named *Julia* is an instance of the concept *Mother* such as axiom 2.8 can be denoted by the expression.

$$\textit{Mother}(\textit{Julia}) \quad (2.8)$$

The role assertions represent relationship between named individuals and may be denoted by expression $\textit{marriage}(\textit{Julia}, \textit{John})$ that means *Julia* is married to *John*.

$$\textit{marriage}(\textit{Julia}, \textit{John}) \quad (2.9)$$

Although, it is intuitively clear that *Julia* and *John* are different individuals, this fact does not logically follow from what we have stated so far. The DL do not make the unique name assumption, so different names might refer to the same individual unless explicitly stated otherwise. The individual inequality assertion such as axiom 2.10 is used to assert that *Julia* and *John* are actually different individuals.

$$\textit{Julia} \approx \textit{John} \quad (2.10)$$

On the other hand, an individual equality assertion states that two different names are known to refer to the same individual. It can be defined as the axiom 2.11.

$$\textit{john} \approx \textit{johnny} \quad (2.11)$$

The situations can arise, for example, when combining knowledge about the same domain from several different sources, a task that is known as ontology

alignment [37].

2.3.5 The relationships with RBox axioms

The RBox axioms refer to properties of roles. As for concepts, DL support role inclusion and role equivalence axioms. For example, the inclusion states that *parentOf* is a sub role of *ancestorOf*, i.e., every pair of individuals related by *parentOf* is also related by *ancestorOf* such as the axiom 2.12.

$$parentOf \sqsubseteq ancestorOf \quad (2.12)$$

In 2.13, the role composition can be used to describe roles such as *uncleOf*. Intuitively, if *Charles* is a brother of *Julia* and *Julia* is a parent of *John*, then *Charles* is an uncle of *John*. This kind of relationship between the roles *brotherOf*, *parentOf* and *uncleOf* is captured by the complex role inclusion axiom

$$brotherOf \circ parentOf \sqsubseteq uncleOf \quad (2.13)$$

The role composition can only appear on the left-hand side of complex role inclusions. Furthermore, in order to retain knowledge reasoning, the complex roles inclusions are governed by additional structural restrictions to specify whether or not a collection of such axioms can be used together in one ontology. The axiom 2.14 shows disjoint roles, nobody can be both a parent and a child of the same individual, so the two roles *parentOf* and *childOf* are disjoint.

$$Disjoint(parentOf, childOf) \quad (2.14)$$

The RBox axioms include role characteristics such as reflexively, symmetry and transitivity of roles. These are closely related to a number of other DL features.

2.3.6 Role restrictions

The most interesting feature of DL, is its ability to link concepts and roles together in one statements. For example, the role *parentOf* has relationship with the concept *Parent*, the *Parent* is someone who is a parent of at least one

individual. This can be defined by the axiom 2.15.

$$Parent \equiv \exists parentOf.\top \quad (2.15)$$

The restriction $\forall parentOf.\top$ is a complex concept saying that a parents is an individual related at least one of the individual by the role *parentOf*. Therefore, the concept *parentOf.Female* describes those individuals that have a daughter. To represent the set of individuals all of whose children are female.

$$\forall parentOf.Female \quad (2.16)$$

Existential and universal restrictions are useful in combination with the top concept for expressing domain and range restrictions on roles. The restrictions include kinds of individual that can be in the domain and range of a given role. We can use the axiom 2.17 to restrict the domain of *sonOf* to male individuals.

$$\exists sonOf \subseteq Male \quad (2.17)$$

Also to restrict the range of a role, we can use the axiom 2.18

$$\top \subseteq \forall sonOf.Male \quad (2.18)$$

In combination with the assertion *sonOf(John,Julia)*, these axioms would then allow us to deduce that *John* is a *Male* and *Julia* is a *Parent*. This is due to the fact that all *son* may be *Male*.

Number restrictions allow us to restrict the number of individuals that can be reached *via* a given role. For example, we can form the at-least restriction, for example the parent that have at least 2 child can be defined in axiom 2.19.

$$childOf.Parent \geq 2 \quad (2.19)$$

Also the axiom 2.20 can define the parent that have at most 2 child.

$$2 \leq childOf.Parent \quad (2.20)$$

Finally, local reflexivity can be used to describe the set of individuals that are related to themselves *via* a given role. For example, the set of individuals that talk to themselves.

$$\exists \text{talkTo.Self} \quad (2.21)$$

2.4 The Semantic Web

The Semantic Web can be defined as "an extension of the World Wide Web using new technologies and standards dealing with interpretation of exchanged data and also with automatically inferring the useful information from these data".

There are a huge amount of data in the web but the computers can't be able to understand or make any decisions with this data. The Semantic Web can enable the processing of the data to infer well defined meaning for a better communication between computers and people. The semantic web makes data to become "*Smart*". The *Smart* data means the information of the web becomes so richly interconnected that the computer can be more able to infer as humans.

The World Wide Web contains billions of web pages which are inter-linked through hyper-links. The daily activity of users in Internet, searching web pages, retrieve relevant information pertaining to the user in shortest possible time. The Web 2.0 based search engines have major drawbacks as it lacks interoperability between machines and meta-data, that tends towards knowledge management crisis.

In this time, the search engines require powerful and complex algorithms in order to parse the keywords given by the users. Semantic Web is based on the principal of interoperability between machines and giving them power to think. It aims at attaching meta-data, specifying relations between web resources and knowledge management, in order to process and integrate data by the users [20].

2.4.1 The Semantic Web Architecture

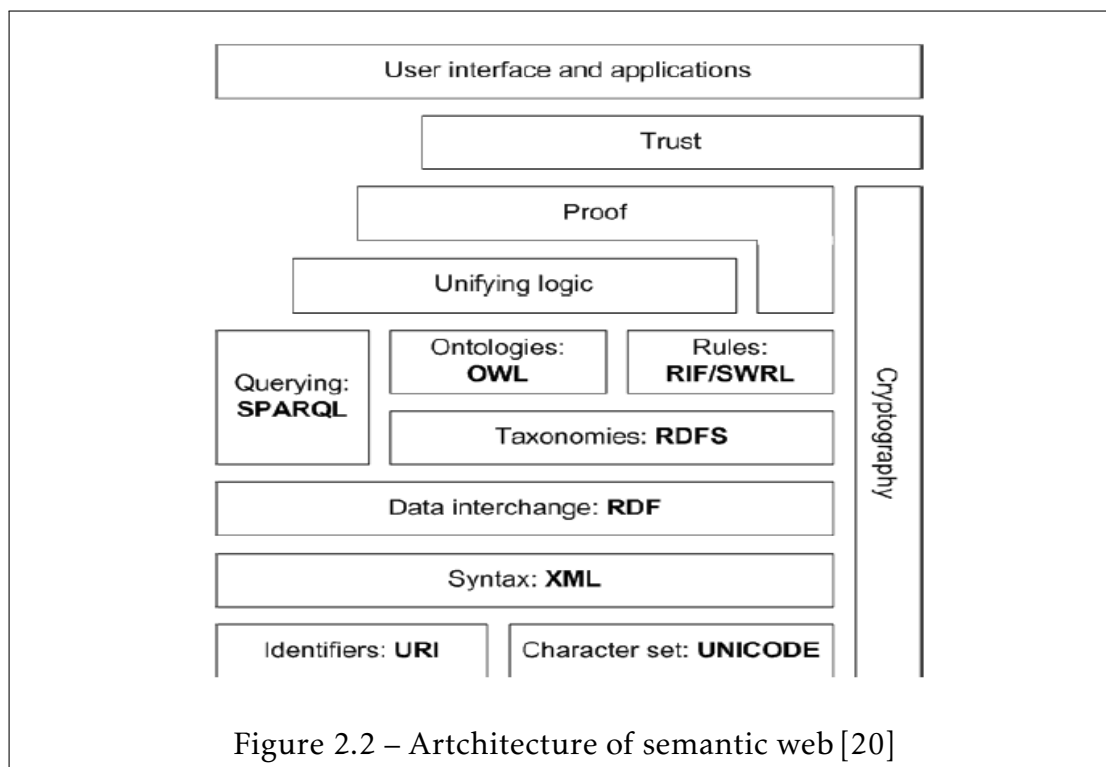
The figure 2.2 illustrates the architecture of the Semantic Web. This includes many layers depending on used languages and their variants.

The first layer URI/IRI (Uniform Resource Identifier / International Resource Identifier) is a string of standardized form in order to uniquely identify web resources and documents. The Uniform Resource Locator (URL) is a subset of URI included to localize the web resources. The IRI is an internationally accepted form. XML layer with the XML name-space and XML schema ensures that there is common syntax used in the semantic web. XML is the key for platform independence and data exchange using a common language.

The core data representation in semantic web is Resource Description Framework (RDF), it is based on triples (subject – predicate – object) and forms a graph pertaining to the given data in the same form. It is the grammar of the document whereas XML is the words understood by machines. The Ontology Web Language (OWL) uses description logic and provides strong semantics. The SPARQL is used to make queries for RDF and OWL ontologies. Syntactically, it is similar to SQL. The correct logic with the rules implied proves the ontologies with trusted inputs to conclude trusted outputs. Digital signatures and cryptography is applied in order to maintain security during information exchange [20].

2.4.2 The Resource Description Framework

The Resource Description Framework (RDF) is a graph-based data model with nodes and edges. It is related to semantic networks. The RDF statements can include three components: *Subject*, *Predicate* and *Object*. The subject is a source of the edge and must be a resource. In RDF, a resource can be anything that is uniquely identifiable *via* a Uniform Resource Identifier (URI). More often than not, this identifier is a Uniform Resource Locator (URL), which is a special case of URI. However, URIs are more general than URLs. In particular, there is no requirement that a URI can be used to locate a document on the Internet. The object statement is the target of the edge. It is like subject, but it can also be a resource identified by a URI. It can alternatively be a literal value as well, like a string or a number.



The predicate statement determines the kind of relationship, held between the *subject* and *object*. It can be identified by a URI. In order to exchange RDF graphs, the W3C recommendation defines an XML syntax for them.

We show below a code snip of RDF in listing 2.1, for the sake of further illustration:

Listing 2.1 – RDF code snip

```

1 <rdf:RDF
2 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns"
3 xmlns:p="http://example.org/pers-schema">
4 <rdf:Description rdf:about="http://example.org/~jdoejane">
5 <p:knows rdf:resource="http://example.org/~jsmithjohn"/>
6 <p:name>Jane Doe</p:name>
7 <rdf:type
8 rdf:resource="http://example.org/pers-schemaPerson"/>
9 </rdf:Description>

```

```
10 </rdf:RDF >
```

As we may notice, all RDF documents usually have a root element, such as *rdf:RDF*. The use of a colon in an element or attribute name indicates that it is a qualified name. Qualified names are used with XML name-spaces to provide shorthand references for URIs. The *xmlns:rdf* attribute on the second line specifies that the "rdf" prefix is used as an abbreviation for the name-space "*http://www.w3.org/1999/02/22-rdf-syntax-ns*". The *xmlns:p* attribute defines *p* another prefix that can be used to form qualified names. Qualified names have the form *prefix:localname* To construct the full URI for a qualified name. Simply it append the local name part to the name-space in correspondence to the prefix.

The *rdf:RDF* element may contain an *rdf:Description* sub-element that is used to identify a resource and to describe some of its properties. Every *rdf:Description* element encodes one or more RDF statements. the subject of each of the statements is the resource given by the "*rdf:about*" attribute, which has the URI "*http://example.org/jdoejane*" as its value. This *rdf:Description* element has three property sub-elements, and thus encodes three statements. The first sub-element is an empty element with the qualified name *p:knows* based on the name-space declaration at the beginning of the document, this refers to the resource "*http://example.org/pers-schemaknows*".

This is the predicate of the statement. Any resource that is used as a predicate is called a property. The *rdf:resource* attribute is used to specify that "*http://example.org/jsmithjohn*" is the object of the statement. In this case the object is a full URI, but it could also be a relative URI. It is also possible for statements to have literals as objects. The second sub-element of the *rdf:Description* encodes such a statement. The corresponding statement has predicate "*http://example.org/pers-schemaname*" and object "*Jane Doe*". By wrapping this text in `<p:name>` start and end tags, we indicate that it is a literal.

The final sub-element of the *rdf:Description* is *rdf:type*. Using the name-space declaration at the beginning of the document. We can determine that this refers to the predicate "*http://www.w3.org/1999/02/22-rdf-syntaxns*" type. This is a property defined in RDF that allows one to categorize resources. The *rdf:resource* attribute is used to specify the category; in this case "*http://example.org/pers-*

schemaPerson". In RDF, types are optional and there is no limit to the number of types that a resource may have [23].

2.4.3 The Resource Description Framework Schema

The RDF is a data model and it does not have the capability to provide the significant semantics. However, the RDF Schema associates a kind of semantic to the data model by means of RDF. The RDF Schema defines some concepts like classes, properties and roles. So the defined in RDF are typed by means of classes, etc.

Therefore, the vocabularies that are provided by RDF Schema to define classes and properties can be included as "*rdfs:Class*, *rdf:Property* (from the RDF namespace), *rdfs:subClassOf*, *rdfs:subPropertyOf*, *rdfs:domain*, and *rdfs:range*". It may also include properties for documentation, including *rdfs:label* and *rdfs:comment*.

RDF Schema has the limitation that it has very weak semantic primitives. This is one of the reasons for the development of Ontology Web Language (OWL)². Each of the important RDF Schema terms are either included directly in OWL or are super ceded by new OWL terms [23].

2.4.4 The Ontology Web Language

The OWL consists of three languages depending on increased ability to express semantic such as OWL Lite, OWL DL, and OWL Full.

1. OWL Lite is mostly used by the users who need simple modeling.
2. The OWL DL is more suitable for expressive description logic, and it includes all of the features that are described in this chapter. Both OWL DL and OWL Lite can use every resource whether a class, object property, data type property, or an instance.
3. OWL Full has the same features as OWL DL, but it is possible to treat a class as an instance, and there is no need to explicitly declare the type of each resource.

2. <https://www.w3.org/OWL/>

All these languages allow to describe classes, properties and instances, but the weaker languages have restrictions on what can be stated or how it may be stated.

The OWL is primarily concerned with the terminology definitions that can be used in RDF documents, i.e., classes and properties. Most ontology languages have some mechanism for specifying a taxonomy of the classes. In OWL, you can specify taxonomies for both classes and properties.

For example, the syntax of the OWL, as shown below in listing 2.2, considers the root of an OWL document as `rdf:RDF` element. It provides some degree of compatibility between the two standards.

Listing 2.2 – An Example of the syntax of the OWL

```
1 <!DOCTYPE rdf:RDF [  
2 <!ENTITY owl "http://www.w3.org/2002/07/owl">] >  
3 <rdf:RDF xmlns:owl ="http://www.w3.org/2002/07/owl "  
4 xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns "  
5 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema ">  
6 <owl:Ontology rdf:about="">  
7 <rdfs:label>My Ontology</rdfs:label>  
8 <rdfs:comment>An example ontology</rdfs:comment>  
9 </owl:Ontology>  
10 <owl:Class rdf:ID="Person" />  
11 <owl:Class rdf:ID="Man" >  
12 <rdfs:subClassOf rdf:resource="Person" />  
13 </owl:Class>  
14 <owl:ObjectProperty rdf:ID="hasChild" />  
15 <owl:ObjectProperty rdf:ID="hasDaughter">  
16 <rdfs:subPropertyOf rdf:resource="hasChild" />  
17 </owl:ObjectProperty>  
18 <owl:DatatypeProperty rdf:ID="age" />  
19 <owl:ObjectProperty rdf:ID="isParentOf">  
20 <owl:inverseOf rdf:resource="isChildOf" />  
21 </owl:ObjectProperty>
```

```
22 <owl:ObjectProperty rdf:ID="isTallerThan">
23 <rdf:type rdf:resource="owl:TransitiveProperty" />
24 </owl:ObjectProperty>
25 <owl:ObjectProperty rdf:ID="isFriendOf">
26 <rdf:type rdf:resource="owl:SymmetricProperty" />
27 </owl:ObjectProperty>
28 <owl:ObjectProperty rdf:ID="hasSSN">
29 <rdf:type rdf:resource="owl:FunctionalProperty" />
30 <rdf:type rdf:resource="owl:InverseFunctionalProperty" />
31 </owl:ObjectProperty>
32 </rdf:RDF>
```

The root tag *rdf:RDF element* contains attributes for each of the name-space prefixes used in the document, such as *owl*, *rdf*, and *rdfs* name-spaces. By using the empty string as the value for the *rdf:about* attribute, we indicate that the base URL of the document should be used as its URI.

The OWL elements serve two basic purposes :

1. Identify the current document as an ontology.
2. Serve as the container for meta-data about the ontology.

In this example, the ontology has *rdfs:label* and *rdfs:comment* properties, both of which are defined in RDF Schema. The *rdfs:label* property provides a human-readable name for the ontology, while the *rdfs:comment* property provides a textual description of the ontology. Both of these can be used to describe the ontology in an ontology library [23].

The OWL Classes

In OWL, we can declare a resource to be a class, by defining its *rdf:type* as *owl:Class*. As shown in the listing 2.2 above, we consider two classes: Person and Man. It is essential to realize in the example that *rdf:ID* is used to identify classes. The names are expanded to a full URI using the ontology's base URI. It is used for name classes and also for properties, it is regarded as an important aspect of the Semantic Web. Human languages frequently have polysemous terms, that is,

words that have multiple meanings. The computer languages can also denote multiple behaviors responding to a same name, where this concept is known as polymorphism.

The meaning of a term must be a unique symbol. Therefore, the solution compounds the problem of synonym, where in general different symbols may be used with the same meaning. If the same class name is used to mean the same thing in different ontologies, then technically these classes will have different full URIs.

As shown, in the listing 2.2, we can not assume that *your:Person* and *my:Person* refer to the same class. However, once discovered, such problems can be easily resolved, since we can use OWL axioms to explicitly relate such classes. If we wish to specify additional information describing the class, then we include properties from the RDFS and/or OWL vocabularies (represented by sub-elements in the XML syntax). The *rdfs:subClassOf* property can be used to relate a class to more general classes. We state that *Man* is a subclass of *Person*. A class can also be said to have exactly the same members as another class using the *owl:equivalentClass* property. This is often used for synonymous classes, particularly when the classes originate in different ontologies, as discussed in the previous paragraph [23].

The OWL Properties

The OWL defines two type of properties

1. The Object properties specify relationships between pairs of resources.
2. The Data type properties specify a relation between a resource and a data type value.

There are a number OWL and RDF terms that can be used to describe properties. As shown, in the listing 2.2 above, we declare *hasChild* and *hasDaughter* as object properties, and also declare *age* as a data type property. While classes in RDF and OWL are typically named using an initial capital letter, and the properties name typically has an initial lower case letter. However, like class names, property names use mixed capitals in complex names.

It is similar the classes, we can be described properties by including sub elements. Can using the statement *rdfs:subPropertyOf* predicate states that every *subject* and *object* pair using the subject property is also a valid *subject/object* pair using the object property. In this way taxonomies of properties can be established.

As shown, in the listing 2.2 above, the *hasDaughter* property is a *rdfs:subPropertyOf* the *hasChild* property. Then, it can be deduced that if *Jack* has the property *hasDaughter Sydney* then *Jack hasChild Sydney* is true. The *owl:equivalentProperty* states that the property extensions of the two properties are the same. Hence, every *subject* and *object* pair for one property can be a valid *subject* and *object* pair for the other property. This is the property analog of *owl:equivalentClass*, and is frequently used to describe synonymous properties.

The *rdfs:domain* and *rdfs:range* properties are used to specify the domain and range of a property. The *rdfs:domain* of a property specifies that the subject of any statement using the property is a member of the class it specifies. Similarly, the *rdfs:range* of a property specifies that the object of any statement using the property is a member of the class or data type it specifies. Although these properties may seem straight forward, they can lead to a number of misunderstandings and should be used carefully. Also the OWL can define number of constructors that specify the semantics for properties such that if it is defined as inverse of relationship like *owl:inverseOf*. For example, the *isParentOf* property can be used as the *owl:inverseOf* of the *isChildOf* property. Thus, if A is the parent of B, then B is necessarily the child of A. Furthermore, the OWL can also define a number of property characteristics, such as *owl:TransitiveProperty*, *owl:SymmetricProperty*, *owl:FunctionalProperty*, and *owl:InverseFunctionalProperty*.

As shown, in the listing 2.2 above, the *is* is a transitive property, while the *isFriendOf* is a symmetric property. The *owl:FunctionalProperty* constructor states that each resource uniquely identifies its value for the property. That is, no resource can have more than one value for the property. On the other hand, the *owl:InverseFunctionalProperty* constructor states that each property value uniquely identifies the subject of the property. For those familiar with databases, the *owl:InverseFunctionalProperty* specifies a property that can be used as a primary key for the class of object that is the domain of the property [23].

The OWL Instances

The OWL can relate instances by using the *owl:sameAs* property to state that the two instances are identical. It is useful in distributed settings such as the Web, where different entities may use different identifiers to refer to the same things.

For example, multiple URLs may refer to the same person such as shown in listing 2.3. A person may have different URLs for their personal and work web pages, or possibly even multiple work web pages as they change jobs over time.

Listing 2.3 – An Example of the sameAs syntax

```
1 <p:Person rdf:about="http://www.cse.lehigh.edu/~heflin/_">
2 <owl:sameAs
3   rdf:resource="http://www.cs.umd.edu/~heflin/" />
4 </p:Person>
```

It is also possible to say that two instances are different individuals. The *owl:differentFrom* property is used to do this. It is often required to show that a set of individuals is pairwise distinct. The *owl:AllDifferent* constructor is used for this purpose, as shown in listing 2.4 [23].

Listing 2.4 – An Example of the All Different syntax

```
1 <owl:AllDifferent>
2 <owl:distinctMembers rdf:parseType="Collection">
3 <p:Person rdf:about="Bob" />
4 <p:Person rdf:about="Sue" />
5 <p:Person rdf:about="Mary" />
6 </owl:distinctMembers>
7 </owl:AllDifferent>
```

2.4.5 The RDF Query Language (SPARQL)

The SPARQL is a query language that is developed primarily to make a query on RDF graphs. The SPARQL can be used to express queries across diverse data

sources, whether the data is stored naively as RDF or viewed as RDF *via* middleware. The SPARQL contains some of capabilities such as required querying and optional graph patterns along with their conjunctions and disjunctions. The SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graphs. The building block for SPARQL queries is Basic Graph Patterns (BGP). A SPARQL BGP is a set of triple patterns. A triple pattern is an RDF triple in which zero or more variables might appear. Variables are taken from the infinite set variable which is disjoint from the above mentioned sets. A solution to a SPARQL patterns to a source RDF graph G is a mapping from the variables in the query to RDF terms such that the substitution of variables in the BGP would yield a subgraph of G (according to the definition of subgraph matching in RDF semantics).

More complex SPARQL queries can be constructed from BGPs by using projection (*SELECT operator*), left join (*OPTIONAL operator*), union (*UNION operator*) and constraints (*FILTER operator*). The semantics for these operations are defined as algebraic operations over the solutions of BGPs [61].

2.5 Conclusion

In this chapter, we discuss in detail the ontological conceptualization and semantic web technology. The ontologies are defined and used to conceptualize the shared vocabulary of different domains. It may help to query the data, based on description logic, which are written using some formal language. It provide formal means for knowledge representation and reasoning. The semantic web techniques are used to develop the ontologies. The semantic web organizes the information in form of inter-linked data units, accessible through hyperlinks. It help to build dynamic web pages which contains the information, resulted from smart data approaches. Various languages of semantic web can be used to develop and extract the knowledge from ontologies. The resource description framework lies at the core of semantic web technology. We can use standard languages, like OWL to build syntactically valid data stores which can be indexed by some query language like SPARQL. We intend to build some tools that analyze

and define the concepts of Logistics and Optimization. It is therefore, in the next chapter we explore the relevance of ontologies that may help us to define logistic ontologies. It may also help us to better understand the much needed optimization of logistic problems.

Logistics and Optimization Ontologies

3.1 Introduction

We explore the possibilities to represent the logistics vocabulary using the ontologies. Logistics processes are complex set of activities. A formal definition of logistics concepts may help us to better narrate the logistic problems and find the relevant optimal solutions. It is for this reason, we define the fundamental data stores in form of ontologies to gather the logistics information. The logistic problems can be defined in such data stores, which can be further exploited, later on, to investigate the exact type of logistic problems. These are also used to incorporate the optimization methods. Similarly, these methods can be later on queried in relevance to the attached logistic problems.

We formally adopt the ontology based approach to define both the logistics and optimization. The available literature provide limited support to deal with the logistics and optimization, simultaneously. It is therefore, we mostly relied on our own defined ontologies and described the corresponding queries. In this chapter, we initially discuss the logistic ontologies to develop the scenarios that may help us to build a logistics reasoning system. We intend to incorporate the optimization ontologies in our defined system. Hence, we explore in detail the optimization ontologies, currently available in the literature. We then exploit

the possibilities of their incorporation in our system.

3.2 The logistics and optimization ontologies

In the existing literature, many ontologies have been developed regarding the logistics and optimization domains; the table 3.1 shows list of ontologies till now for 2001. The work as cited in [2] and [3] concerns broadly the city logistics. The authors in [45] and [69] contributes the work concerning the monitoring system for logistics. The main interest, of the available ontologies, has been mostly focused on the definition of the general concepts the supply chain management of logistics. Among others, let us briefly classify the available ontologies into six categories, which are considered more close to the current work. Among them, the three are related to the logistics while the other three concerns the optimization. These can be discussed, in further detail, in the following sections.

3.3 The logistics ontologies

It has been more than a decade, the ontologies has been defined and used widely in different research domains. We explore the existing literature concerning the logistic ontologies. In this regard, we briefly discuss some of the most relevant research work in the following sections.

3.3.1 The Ontologies of Supply Chain Operations Model

Leukel [40] proposed ontologies dealing with the Supply Chain Operations Model (SCOR). The supply chain is a system of entities around the manufacturers, suppliers, transporters, and customers. It is basically involved in producing/-transforming the goods or services from manufacturers to customers. The SCOR provides a comprehensive set of modeling the supply chains.

The SCOR consists of a model stack of top level, configuration level, and process element level, as follows :

- Top level : this level distinguishes five core management processes called 'process types' that are relevant for all firms in a supply chain. These are

plan, source, make, deliver, and return.

- Configuration level : it provides for each process type of the top level a set of 'process categories' which represent different operational strategies that a company pursues. For instance, the process categories for 'source' represent sourcing strategies. A company can describe its logistics processes in a so called 'process map' by connecting the process categories. In addition, metrics and best practices are assigned to the categories.
- Process element level : it decomposes the process categories by adding process element definitions and process element information inputs / outputs. For instance, a particular source category may be decomposed into process elements of receiving, verifying, and finally stocking the goods. Metrics and best practices of the former level may appear here in more detail. This level provides the most comprehensive set of modeling primitives.

The SCOR model has been further extended with definition of yet another level of ontologies, in relevance to the logistics process. The details of which are as under :

The logistics top level ontology

The figure 3.1 shows the top level of logistics ontology. It includes *Process* class having the sub-classes *Plan*, *Source*, *Make*, *Deliver*, *Return* and the *Metrics*. The ontology also defines relationships among *Process type*, *Company*, and *Good*.

Author	Scope	Application	The top level of concepts
Madni [44]	Supply chain	(1) Crisis action planning and execution, (2) Integrated product process development	Enterprise, process, resource, objective, plan, activity
Pawlaszczyk [51]	supply chains	Agent-based supply chain simulation	Activity, plan, product, organization, time, event, transfer-object, flow, performance
Fayez [16]	Supply chain	Supply chain simulation	Functional units, processes, materials, objects, information, information resources, schemas, performance measures, practices, resources, decisions, views, tiers
Matheus [45]	Logistics	Monitoring of spare parts logistics	Airbases, aircraft, parts, facilities, remote supply depots, event object attribute
Witherell[72]	Optimization	developed optimization ontology to facilitate engineering problems	Continuous, discrete, unconstrained, constrained, stochastic programming, integer programming
Leukel [40]	Logistics	(1) Reconstruct supply chain from knowledge base, (2) Aggregating process fragments	Process, process type, good, company, metric, best practice, information
Hoxha [29]	Logistics	Logistics vocabulary, interoperability and integration	Logistics actor, Logistics role, logistics service, logistics object, logistics KPI, logistics resource, logistics location

Han [22]	optimization	conventional/mathematical programming and simulation optimization	Optimization problem, nonlinear programming, quadratic, linear programming, optimization method, meta-heuristic method, simplex method, optimization component, constraint, objective function, solution, restriction
kowalski [36]	Supply chain	an ontology driven case-based reasoning system	physical object, abstract object, means of transportation, mode of transportation, success factors, KPI
Anand [3]	City logistics	Automated categorization, query answering, modeling and simulation	Stakeholders, objectives, KPI, resources, measures, activity, R&D
Anand [2]	City logistics	the decision making process of the city logistics stakeholder under different situations.	shop, goods, cost, stakeholders, trucks, goods ordering, carrier.
Moussas [46]	Optimization	Designed and structured with the main focus on the optimization.	Optimization problem, iPRODX problem, Graph Problem, DOE Problem, Multi Objective Problem, Quadratic, Nonlinear problem, optimization method, optimization algorithm, genetics algorithm, NLS Algorithm, Simplex Algorithm, Newton Algorithm
Wang [69]	Logistics	monitoring and decision logistics system	Parameter, Diagnostic, Temperature, humidity, location, time and product information

Table 3.1 – The individuals of logistics and optimization ontologies

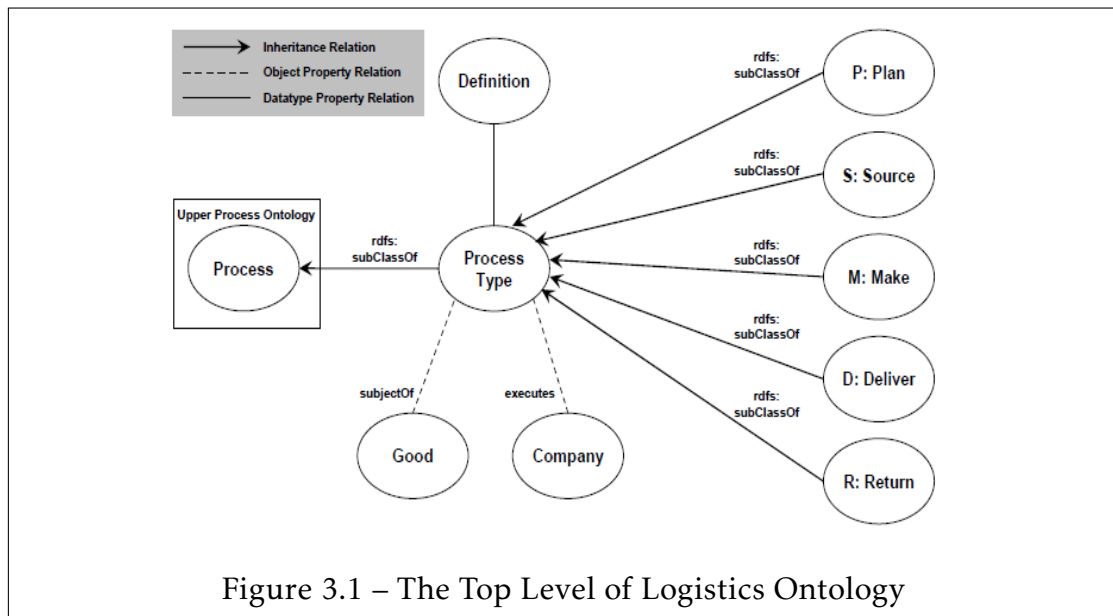


Figure 3.1 – The Top Level of Logistics Ontology

The logistics process type ontology

Three process types have been defined depending on whether the goods are *on a stock*, *made-to-order* thus manufactured for a specific customer order, or *engineered-to-order*, e.g. designed and manufactured specific to a particular customer requirement.

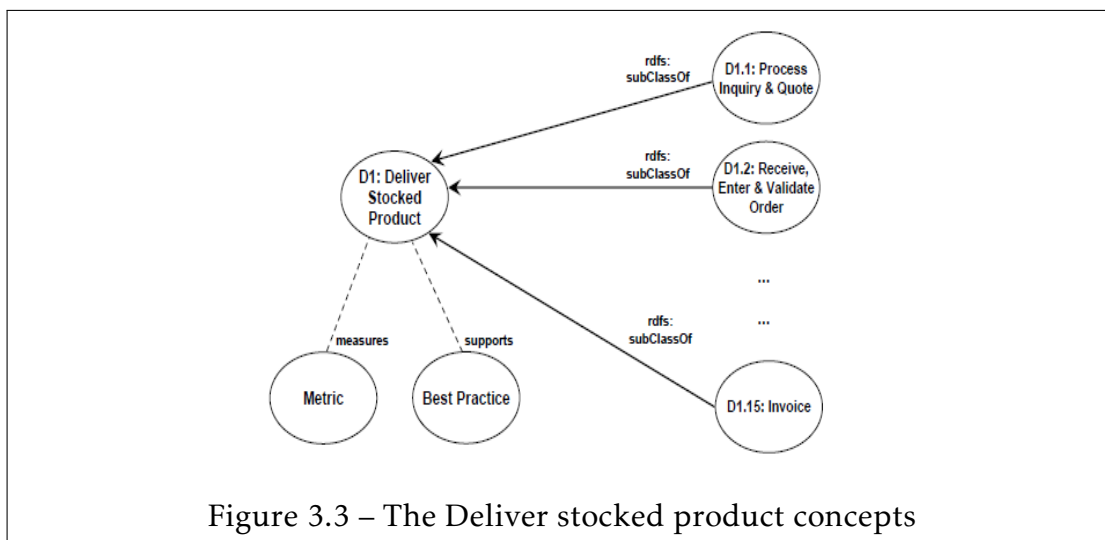
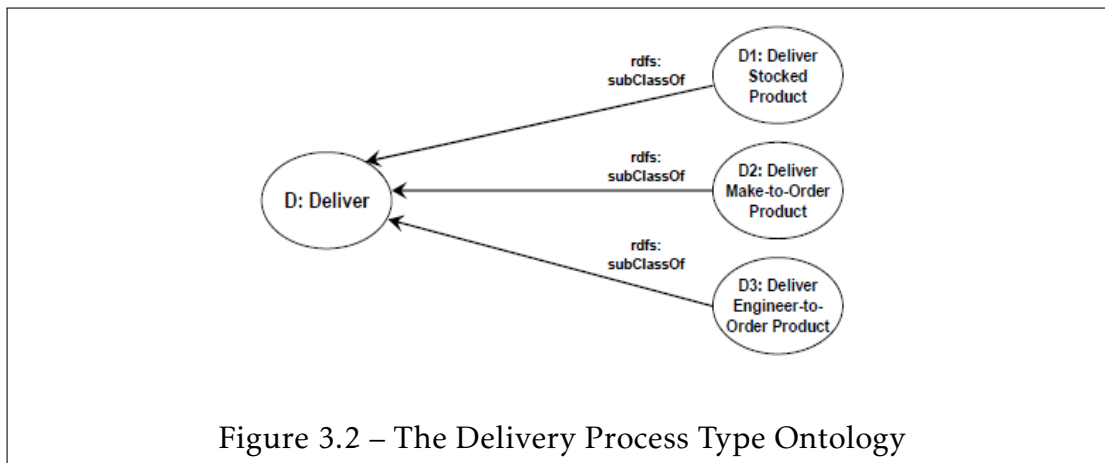
The figure 3.2 shows the concepts of *Delivery* process type refined by three sub-classes *Deliver stocked product*, *Deliver make-to-Order product* and the *Deliver engineered-to-Order Product*.

The logistics process category ontology

This ontology provides a more detailed representation level of logistics elements associated to process types. The Process elements introduce the lowest level of abstraction by specializing process categories.

This ontology also defines the Metrics for assessing the performance of a process category. This also describes the best practices that are empirically proofed means for achieving good performance. In the ontology, both metrics and best practices are included in the model by Object Properties.

The figure 3.3 shows the Process category ontology related to the *Deliver*

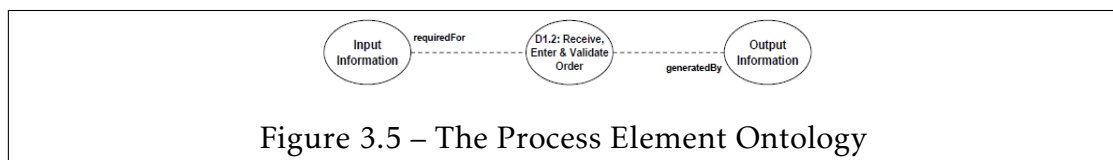
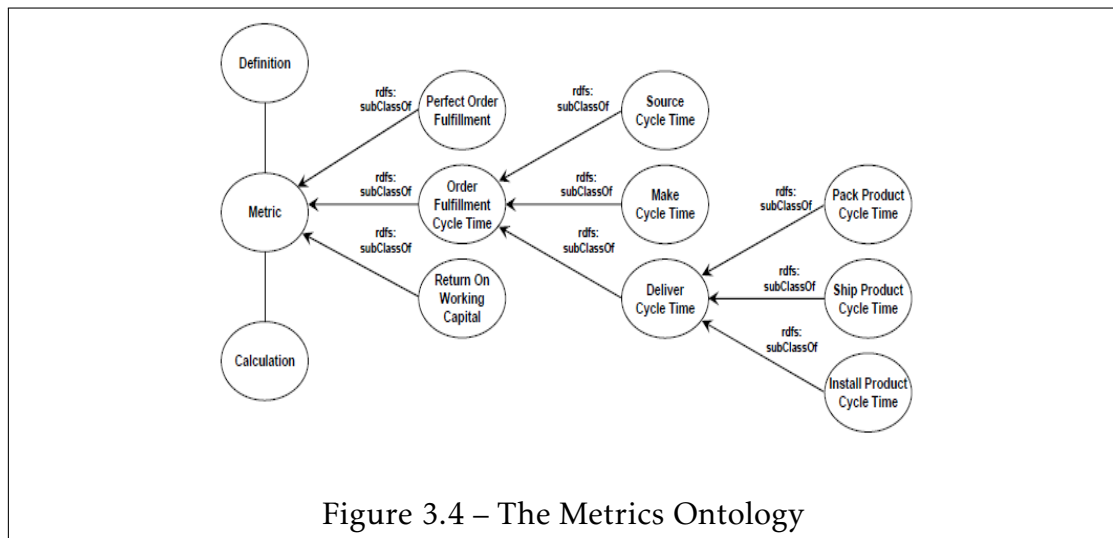


stocked product. Many sub-classes of *Deliver* are defined such as "*Process Inquiry and Quote*", "*Receive, Enter & Validate Order*", "*Invoice*", etc.

The "*Deliver Stocked Product*" class is also related to the "*Metric*" class and support to "*Best practice*" class. These two classes may be refined by hierarchy of sub-classes defining in more details the metrics and best practices such as shown in figure 3.4.

The logistics process element ontology

The Process Element Ontology focuses on the input information required for executing a process element and the output information which is the result of



this execution.

The figure 3.5 does not list the inputs and outputs, since these are defined in another part of the ontology. The allowed information is defined by a constraint on the two Object Property relations.

3.3.2 The Ontology-Driven Case Based Reasoning System

Kowalski [36] has shown case based reasoning system to develop the ontology that measure similarity of logistics knowledge collections written in natural language. The system is based on the ontology concerning the linguistic aspects of the logistics area constructed by OrGoLo (Organizational Innovations *via* Good Governance in Logistics Networks).

The figure 3.6 represents the ontology of the case based reasoning system. This ontology defines the *physical* and *abstract* logistics objects. The *physical objects* include the *means of transportation*, *mode of transport*, *logistics services shipment* and *goods*. The *mode of transport* includes *road*, *water route*, *rail cargo*, and *air route*. The *means of transportation* contains *aircraft*, *truck*, *train*, *rail*, and

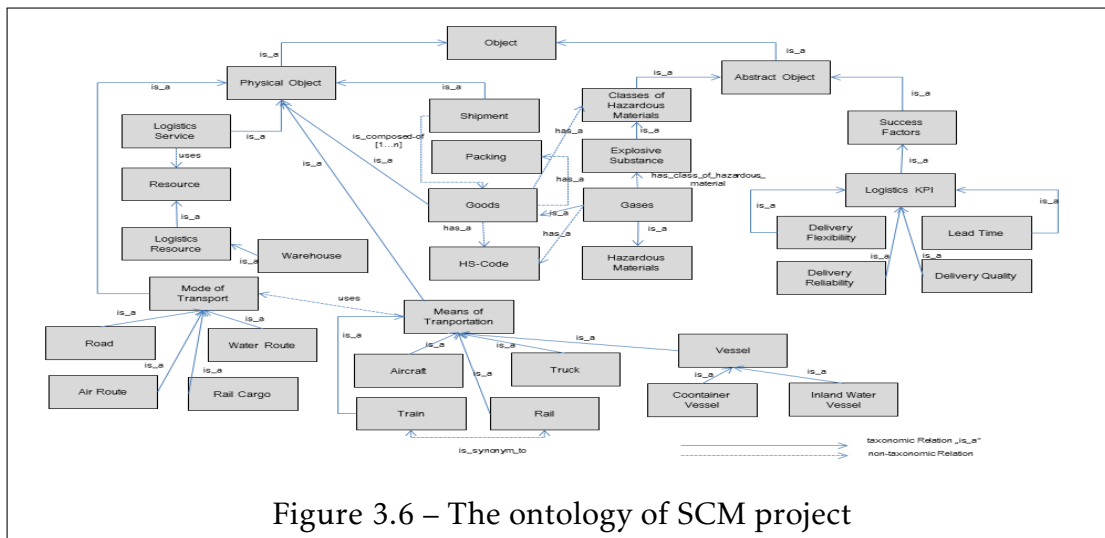


Figure 3.6 – The ontology of SCM project

vessel. The *abstract object* includes *success factors* (Key Performance Indicators) and classes *hazardous materials*. This ontology can include many roles such as *is-composed-of* between *shipment* and *goods* and *uses* between *the means of transportation* and *mode of transport* [36].

3.3.3 Semantic services application framework

Hoxha [29] presents a Semantic Services Application Framework that enables automated and intelligent techniques for discovery, ranking, execution and efficient composition of services into more complex and flexible logistic processes such as shown in figure 3.7. The services should be made available *via* the different Service Providers, who use the appropriate tools for semantic Annotation and Publishing of their services. The semantically annotated services, precisely generate service descriptions, which are saved in a repository together with the ontologies and which use these descriptions.

A Service Requester may search for services *via* a Web-based search interface, specify the desired preconditions, *inputs*, *outputs*, and *effects*. The user query is processed, using complex reasoning techniques, to discover in the repository those descriptions. The returned result may be a list of atomic services or even compositions (composite processes) of services. The list is ranked and displayed to the user, who is then able to execute, according to the implementation made

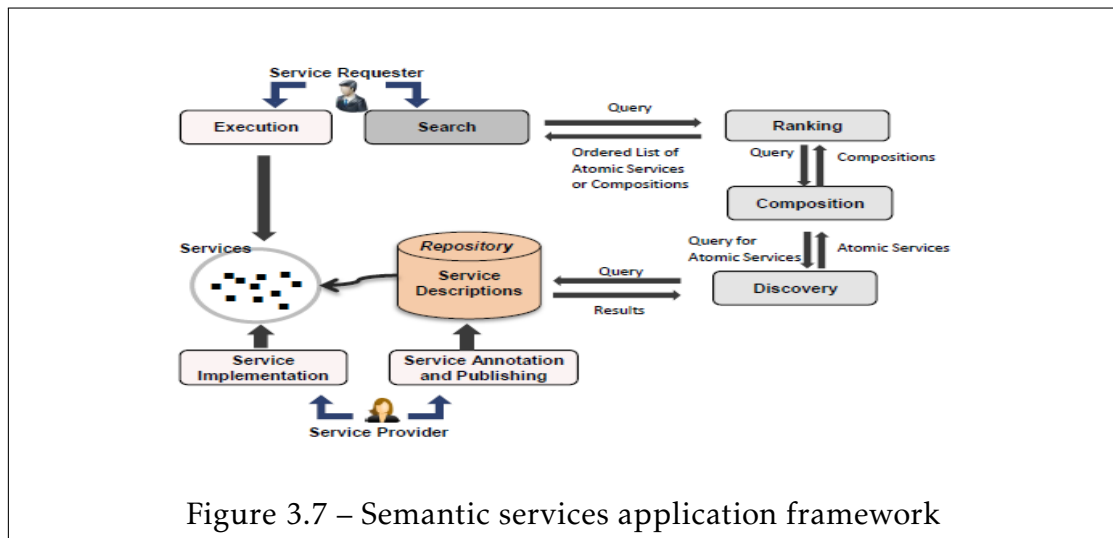


Figure 3.7 – Semantic services application framework

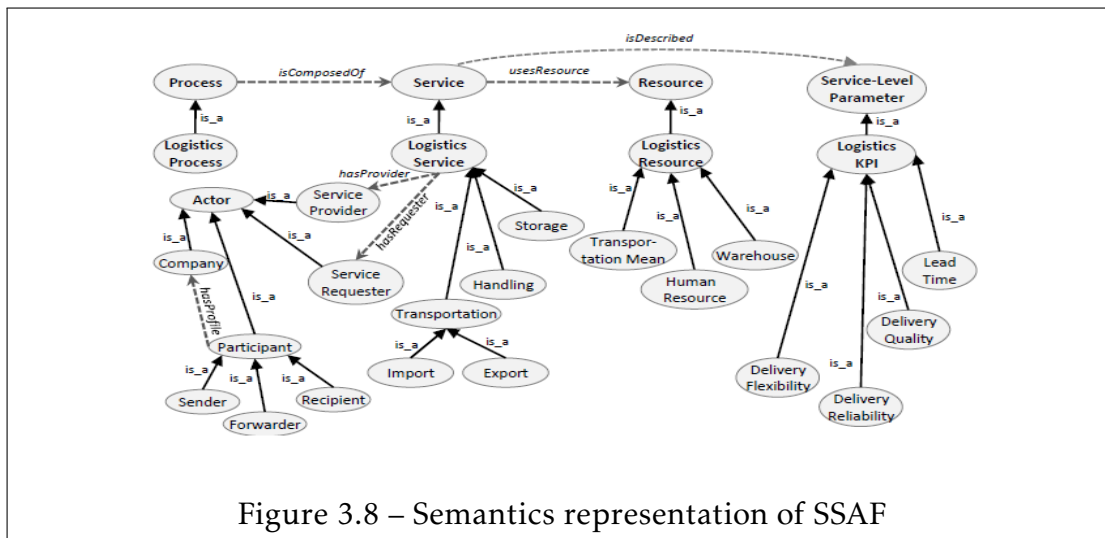
available by the provider, the service that best matches his needs.

The figure 3.8 shows the semantic representation side of the Semantic Services Application Framework. It consists of an ontology that includes the top level concepts as *process*, *service*, *resource*, and *service level parameter*. The *logistics resource* concept is further specialized into more specific concepts. The *transportation mean*, *warehouse*, and *human resources* are derived as sub-classes of class *logistics resource*. *Logistics KPI* (Key Performance Indicators) are specialized, among others, into sub concepts of *Delivery Flexibility* and *Delivery Reliability*. There are also many roles that connect the concepts such as *isComposedOf* and *UsesResource*, etc [29].

The logistic ontologies are subjected to define the logistic problems and the optimal methods that can be used to solve them. Henceforth, in the next section, we discuss the optimization ontologies intended to serve the purpose.

3.4 The Optimization Ontologies

Most of the work in the literature focuses on the problem field (design, simulation, and modeling). It includes the optimization as a part of the solution to derived problems. In this Section, we explore some of the existing individual ontologies related to the optimization problems and their solutions.



3.4.1 Ontology for Simulation optimization

The Ontology for Simulation optimization (SoPT) contains the concepts from both conventional/mathematical programming and simulation optimization. The aims of SoPT is to describe simulation optimization methods. It help to detect the correct tool for each specific case to facilitate component reuse, especially in systems where simulators and optimizers are loosely-coupled. It present an ontology for optimization that was designed and structured with the focus on the optimization domain. It can also be capable to support any other application domain with minimal expansion.

The figure 3.9 shows the top level of SoPT. It includes *optimization problem*, *optimization methods*, *optimization component* [22].

The Optimization Component

The Optimization Component model has the various parts of an optimization problem. It includes *data types* and *restrictions of input variables*, *objective function*, *constraints*, *solution*, and *solution quality*.

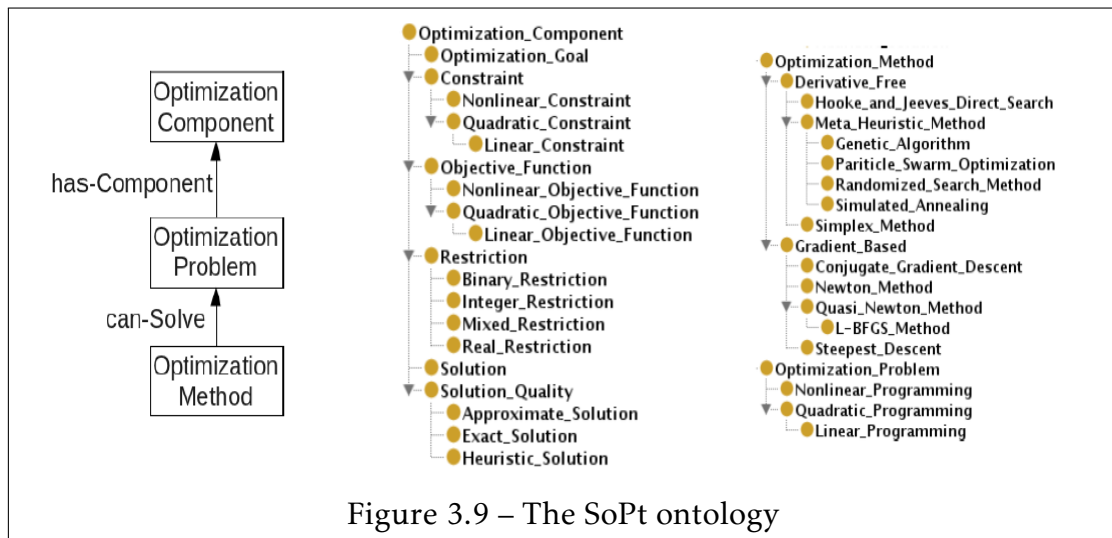


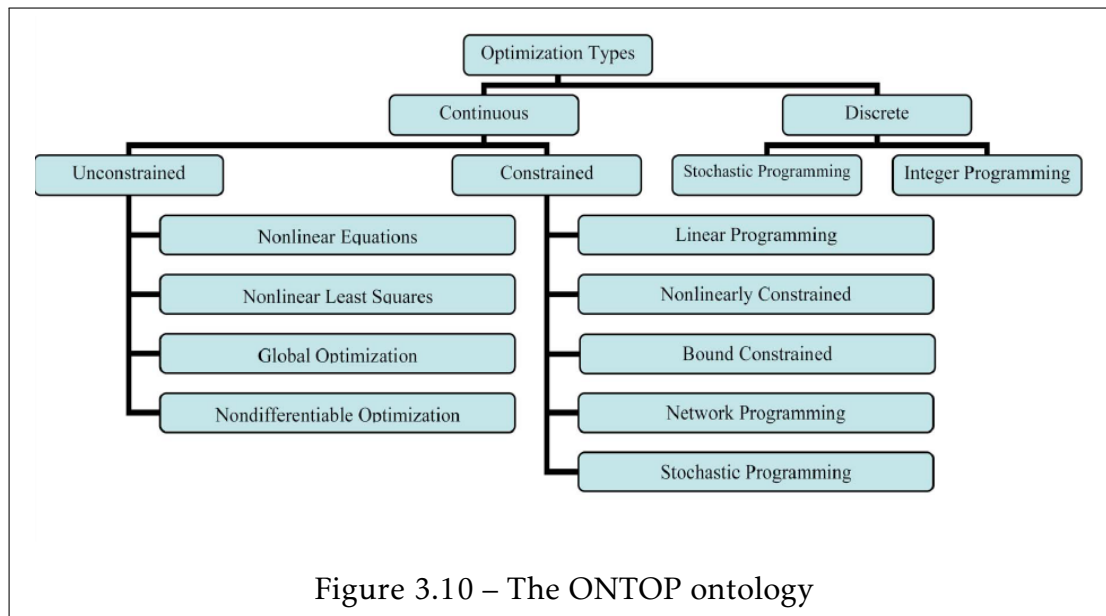
Figure 3.9 – The SoPt ontology

The Optimization Problem

In SoPT, the *Optimization problem* is classified into sub-classes such as *Linear Programming (LP)*, *Quadratic Programming (QP)* and *Nonlinear Programming (NLP)*. It also has roles that link the *Optimization Problem* with both the *Optimization Component* and *Optimization Methods*.

The Optimization Method

The figure 3.9 shows the classification of the *Optimization Method*. The sub-classes that it contains depends on wide variety of methods used for simulation optimization. It is difficult to determine a starting point, because it is focuses on iterative interaction between the simulator and the optimizer. We concentrate on what the optimizer needs from the simulator along with a set of response vectors estimating a portion of response surface. We also consider, how the optimizer explores the parameter space using different search techniques. A simple illustration is a gradient based technique, where search involves direction determination to assure how far to move in that direction.



3.4.2 The Ontology for Optimization

The Ontology for Optimization (ONTOP) specifies the knowledge base that incorporates the standardized optimization terminology and formal method definitions. It often include unrecorded optimization details, such as the concepts or assumptions that may be formed while creating the optimization model. It may also considers the model developer's rationality and justification behind such concepts and assumptions. The ONTOP attempts to assist the engineer step by step through the work flow. It has been initially developed to facilitate case studies of engineering design problem. The figure 3.10 shows the top level of ONTOP, it contains general Optimization Type classified into discrete or continuous concepts [72].

Furthermore, it classifies the *continuous* into *constrained* and *unconstrained*. Likewise, the *discrete* is classified into *stochastic programming* or *integer programming*. The main advantage of ONTOP is its capability to store the optimization knowledge within an ontological structure. It is accentuated by the hierarchical structure possessed by the taxonomy of optimization types. The classes within the taxonomy are defined by the specific properties associated with the optimization types. At the taxonomy's highest levels, it distinguishes between

method types, such as continuous or discrete, constrained or unconstrained, etc. These types are separated by creating sub-classes within the optimization type taxonomy. At its lower levels, the taxonomy distinguishes between specific method types, such as linear programming and topological optimization. Such taxonomy may then allow the number of methods to be increased or reduced as the field of optimization changes. Specific optimization models may fall under their respective optimization method, within the class hierarchy.

These properties are used to define the optimization trait of the model. These are eventually used to determine the permitted optimization methods for the model. Once these methods are determined, the model may then be defined by technique-specific properties [72].

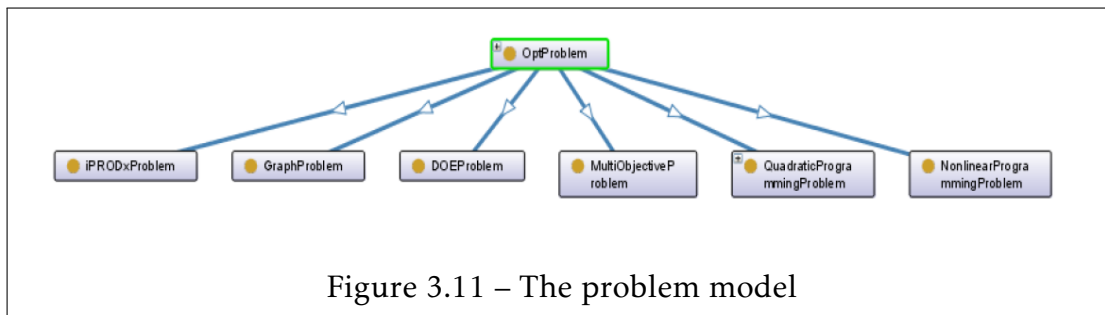
3.4.3 The General Optimization Ontology

The General Optimization Ontology (GOO) is relatively closer to define the ontology for logistics optimization, as it is primitively designed and structured according to the optimization domain. Though, it is capable to support the other application domains with some minor expansion. The authors developed the basic concepts which are common to all optimization problems and these are essential for the core part of the ontology. The aim is to support automatic selection of the appropriate optimization tool for a given optimization problem, therefore it defines how it can be complemented by domain specific modules, tailored to the optimization problems under investigation [46].

The structure of General Optimization Ontology

The Structure of GOO includes the definition for typical optimization problems along with the descriptions of the methods and algorithms applied to solve an optimization task. The GOO concepts are classified into numbers of optimization tasks. The main components are :

- The optimization problem model,
- The optimization solution methods,
- The optimization algorithm.



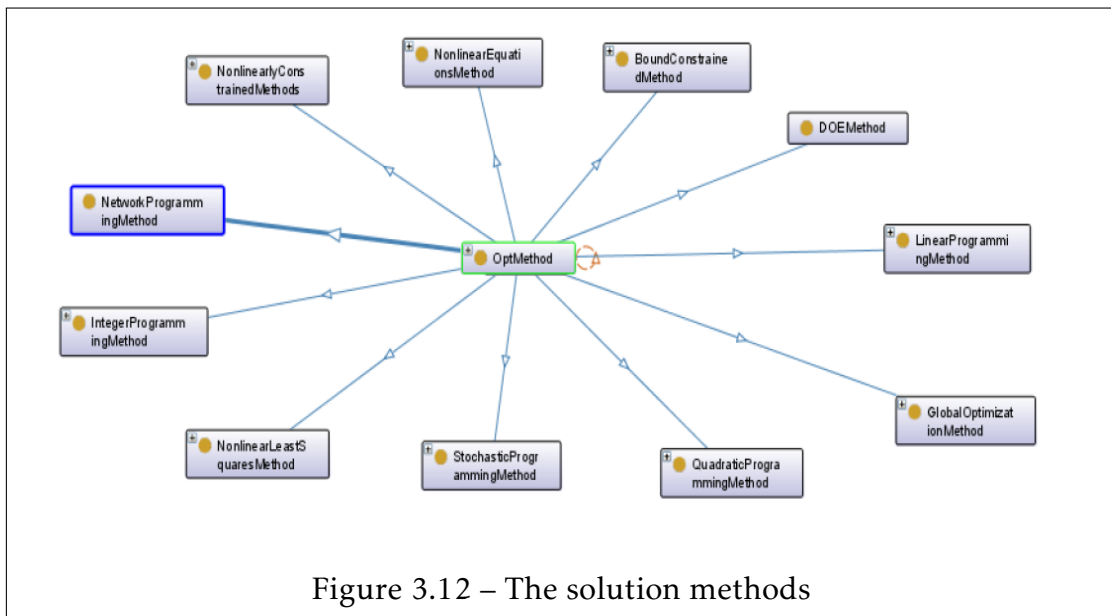
In consequence, the optimization problem model can be solved by the solution methods that invoke one or more optimization algorithms.

The figure 3.11 shows the classification of problem model categories. Each model of problem can be associated to one of the numerous optimization problem. The categories of model of problems depends on the way the problem is specified such as type of objective function, for example linear programming or non-linear programming, and *vice versa*.

The figure 3.12 shows the taxonomy of solution methods that are used in the GOO. It is developed mainly to classify the numerous techniques i.e., there may exist at least one solution method for each category of optimization problems or a method is proposed to solve the specific category problems. There may also exist some solution methods that can solve several problems emerging from many different categories.

3.5 Conclusion

This chapter has been dedicated to explore the related work in the available literature. We discuss in detail the significance of ontologies to define a reasoning system. We investigate the available logistics and optimization ontologies. The logistic concepts can be formalized using ontologies. Our this claim is further justified by the defined logistic ontologies in this domain. We consider the ontologies of supply chain operations model (SCOR) which englobes the supply chain entities in form of ontologies at different levels. Most of the available ontologies are developed to deal with a specific problem, in its respective domain. We intend to extend these work with the help of optimization ontologies for



logistics. We aim to incorporate optimization ontologies to provide convivial means of exploring the logistic problems and attach them to the optimization solution methods. It is therefore, we also discuss the available optimization ontologies. We explore in this regard the SoPT, ONTOP, and GOO. These different optimization ontologies inspire us to adopt a personalized approach to define the optimization ontologies in relevant to the logistic problems. In the next chapter, we present the different aspects of the intended reasoning system and its capabilities to incorporate the logistics and optimization ontologies.

Part II

The Reasoning System

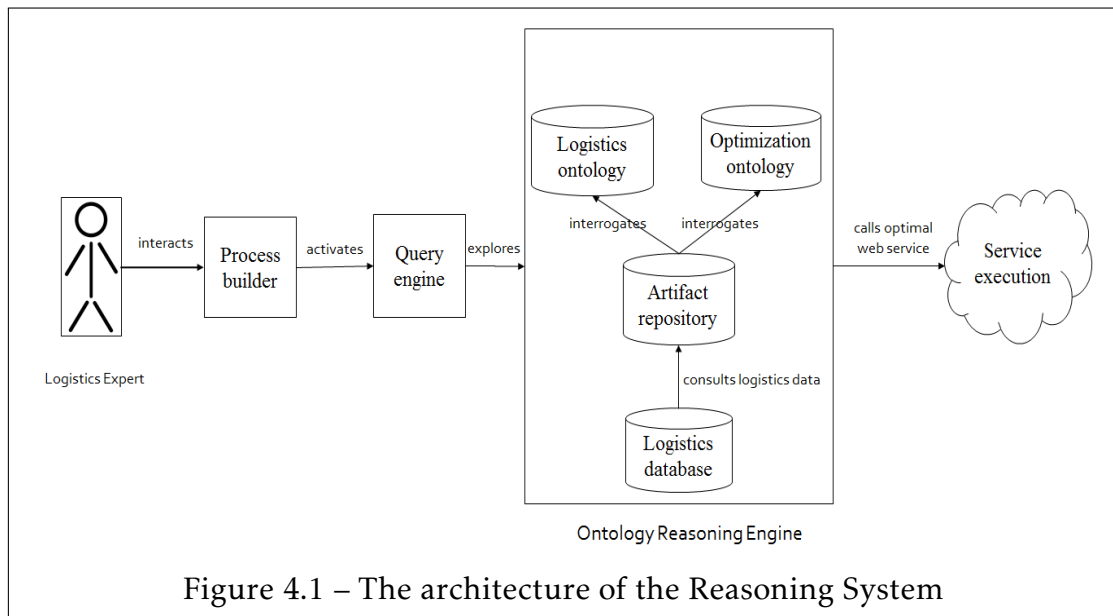
The Architecture of the Reasoning System

4.1 Introduction

Designing a reasoning system is logical approach to host the multiple repositories and manage their interactions. We propose to develop a centralized repository in form of a reasoning system to manage the logistic and optimization ontologies along with other logistic data to facilitate a usable access for the logistic experts. The proposed system is destined to allow a logistic expert to query the logistic problems and then present the relevant optimization solution methods. The logistic expert may choose a solution method which is inherently connected to the web-services (implementing the solution algorithms).

The Reasoning System assists the logistics engineers to identify and specify their problems. It uses concepts domain to explore typical logistics problems and exploit corresponding optimization methods. It then finds the concerned software units (optimal web-services) that implement these methods to solve the particular logistics problem.

In the literature, we find many similar works that use the ontology to define the concepts and/or to identify the problem. The proposed reasoning system manipulates the logistic and optimization ontology repositories. It uses a semi-automatic detection search algorithm for related concepts to identify the type of



logistics problem.

4.2 The Reasoning System Architectures

The figure 4.1 shows the global architecture of Reasoning System. The major constituents of the reasoning system are the *Process Builder*, the *Query Engine*, the *Logistics and Optimization Ontologies*, the *Artefacts Repository*, and finally the *Web Services* that implements the optimization solution methods.

Once the logistics expert interacts with the reasoning system, (s)he uses the process builder. This instantiates the logistics processes with the definition of the problem attributes. It allows, on one end, the logistics expert to interact with the system with the help of usable interfaces, and on the other end, to interact with query engine.

The *Query Engine* is responsible of building and executing the queries, dealing with knowledge extraction from ontologies. It also provides the means for the selection of the optimal web-service among the existing ones that are available on web.

The *Ontologies* may be considered as the core of reasoning system. The reasoning system includes two main ontologies dealing with both logistics and

optimization knowledge. The logistics ontologies are the conceptualization of the domain knowledge associated to logistics and logistics problem. While the Optimization ontologies deal with knowledge concerning the Optimization domain, problems and resolution methods,...,etc.

The *Artefacts Repository* deal with the stored data representing the various artefacts related to the logistics and optimization problems and also the problems resolution process. Such data is particularly useful during the execution of the reasoning system. We may notice some main artefact types such as the *Resource Concepts* representing the concepts involved by a specific problem and the *Resources Concept Paths* representing the possible paths, that may link the concepts by means of various possible roles or relationship linking them. The Artefacts Repository also store information about the available Web Services. Hence, it defines two databases that are the *Conceptual Paths Databases (CPDB)* and the *Web Services Databases (WSDB)*.

Finally, *Web services* part represents the relative software development by the community, which is available on-line. The web services are organized following the problem they solve and the methods they implement to solve there. The different parts of the Reasoning System will be discussed in more details in this section.

4.2.1 The Process Builder

The process builder gives the possibility to logistics experts to interact with the Reasoning system. It is also interconnected with Query Engine. There are many components allowing the logistics experts to identify the attributes of their problem. They may identify their Logistics Resources Concepts (RC), also ask the Query Engine to provide the SPARQL queries that extract the Resources Concepts from the logistic ontologies.

4.2.2 The Query Engine

The Query Engine provides needed queries to extract the knowledge from ontologies or databases. It may respond to many types of queries such as SPARQL and SQL queries. These queries are applied on ontologies. There are many

examples of these queries such as find what is the type of the logistics problem, find the list of methods solving a particular logistics problem. The SQL queries extract data from relational databases such as to find which web services are available. The relational data of the web services is stored in a repository. It can be queried in relevance to the logistics problem and the optimization method. The extracted logistics data can be transferred to the web services, for the execution.

4.2.3 The Logistics Ontologies

The logistics ontology is intended to capture the essence of logistic domain in terms of semantics expressed by means of concepts or classes and relations or roles linking them. The semantics is formally defined by means of axioms using the Description Logic (DL).

We discuss more about logistics ontologies in chapter 5, that focus on the description of the concepts of supply chain management (SCM). We define then the logistics ontology including the top level classes as *Process*, *Service*, *Resource*, *Performance*, *Activity*, and *Logistic Problem*.

Likewise, we define the roles or relationships between these concepts such as *composes* role among *Process* class and *Service* class. Therefore, we define the axioms between related concepts. We use the Description Logic (DL) to define the complex relation among the concepts. Finally, we define the concepts of logistics problems, roles and data properties that are needed to identify each problem.

4.2.4 The Optimization Ontology

The optimization ontology concerns the typical optimization problems along with the description of optimization methods applied to solve them.

The basic structure of this ontology should support optimization processes. It should also focus on how to select and apply a suitable solution for the encountered optimization problem. It reflects that the ontology classes must eventually cover all entities that concur in an optimization task.

In chapter 6, we explain more precisely the Optimization Ontology. We specify the general terminology of optimization ontology. This may include the concepts *Optimization Problem*, *Optimization Method*, *Logistics Optimization Problem* and *Optimization Component*.

We continuously specify sub-classes of each class on top level. For example, the *Optimization Component* has four sub-classes *Data*, *Objective Function*, *Constraints*, and *Parameter*.

Therefore, we define the relationships or roles among the concepts as *isType* role between the *Logistics Optimization Problem* and *Optimization Problem*. We also define the axioms to represent the complex related concepts.

Finally, we build the reasoning system intended to solve the logistics optimization problems. We consider three logistic problems, that are the Vehicle Routing Problems (VRP), the Passenger Trains Problems (PTP) and the Container Terminal problems (CTP), for the sake of more precise illustration.

4.2.5 The Artefacts Repository

The Artefacts Repository is responsible to identify and store the meta-data or concepts used by the logistics experts. These meta-data are particularly useful to identify the Resources Concepts Paths (RCPs) for each Resources Concepts (RC). The meta-data are also useful to produce the Similar Concept Path Groups (SCPG) for RCPs.

The Artefacts Repository can also store both the persistent and temporary data. There are many examples of the temporary data such as the RCPs, SCPG, type of logistic problem, and types of methods found from ontologies. The Artefacts Repository includes two Database that are the Concept Path Database (CPDB) and the Web Services Database (WSDB). The *Concept Path Database (CPDB)* is responsible to store the RCPs that define each logistics problem. One can easily identify the type of logistics problem by comparing the resulting SCPG with those in CPDB.

The *Web Services Database (WSDB)* is responsible to store each web services associated to the optimization or logistics problem. So, it is possible to retrieve web services according to the logistic problem type, the optimization method

used, . . . *vice versa*.

4.2.6 The Web Services

The web services can be defined as any piece of software that makes itself available over the Internet and uses a standardized messaging system. There are two standard communications used in web services such as Web Services Description Language (WSDL) and JavaScript Object Notation (JSON). The WSDL is used as specification interface of web services by sending XML messages. The JSON is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs[55].

There are many web services available over Internet, which are developed by the community to solve optimization problems. Each web service may be built to solve a logistics problem. The web services may use one or more optimization methods. These web services can be classified into two types: the commercial web services and the personal. The commercial web services are developed by specialized companies to solve the optimization problems. The personal web services are developed by PhD students or research teams as a mean to validate their resolution methods and these are mostly available free of cost. For example, there are many web services over Internet to solve the Vehicle Routing Problems. We mention some these web services, as follows :

- The OptimoRoute¹ web service, is a commercial web service, it suggests the allocation of orders to vehicle drivers on the basis of the road network. It may additionally solve the problem with add constraints such that driver work time and vehicle loading capacity.
- Another web service which is available on internet is TrackRoad², it is also a commercial web service. The proposed service includes the management of the driver and orders on the basis of the road network.
- The VRP solution [52] is an open source web service developed to be used for research proposes. It was developed by using the hybrid optimization methods. It first uses local search to arrange the orders, then it uses first

1. OptimoRoute:<http://www.optimoroute.com>

2. TrackRoad:<http://doc.trackroad.com>

order split methods to calculate the routes. It has been used to solve CLVRP and TSP problem. It has one objective function that is minimum distance.

4.3 Conclusion

In this chapter, we discuss the global architecture of the reasoning system, that we develop as a prototype to validate the proposed approach. The reasoning system is comprised of logistics and optimization ontology repositories, along with the artefact repository and integrated query engine. The reasoning system provide the means for a logistic expert to interrogate the system in a usable manner. It allows its users to query the exact type of their logistic problem and provides a list of optimization solution methods. The reasoning system also indexes the available web services in relevance to the proposed optimal solution methods.

The contents in this chapter give general idea the current work such as, who may benefit from the reasoning system; which are the defined ontologies type; how can a user interact and use the web services, etc. We continue to explain further the each part of Reasoning System in the following chapters. The chapter 5 explains the logistics ontology and the chapter 6 explains optimization ontology. Likewise, the chapter 7 discusses the algorithms to identify and solve the type of logistic problem such as to find the Resources Concepts Paths (RCP) and the general logistic problem. It helps to map the logistics ontologies and optimization ontology.

The Logistics Ontologies

5.1 Introduction

The logistics connect vast amounts of events, activities, and actors. The involved complexity can generate difficult situations to be analyzed for process improvement [57]. We believe ontologies can help in this matter. These potentially contribute in solving integration problems in information systems [35]. These can extend the research work for the harmonization of heterogeneous information sources for knowledge discovery [17, 34].

We explained in chapter 2, there are many individuals of ontologies. These can be adopted on logistics. The existing ones are mostly focused on supply chain management [46]. These often does not consider the complex logistic problems, like the optimization problems [40] in logistics. However, the current logistic ontologies has not yet achieved a consensual acceptance and maturity.

We define and specify logistic ontologies to solve complex logistic problems. These are further extended to cope with the logistics optimization problems, which are encountered due to the heterogeneity of involved elements. These ontologies assists the end users in exchanging semantic.

The ontologies describe the shared information related to the knowledge domains and the instances for optimization. It requires, formerly, to investigate the optimization related knowledge and artefacts, while laterly, to establish common vocabularies, nomenclatures, and taxonomies among them.

An artefacts represents an activity, event, actor, resource, or any document that can influence the other related element. A relation is transient dependence that signifies the strength to influence the involved elements, although it evaluates the nature of dependence.

We further define the interoperability, integration, reuse of artefacts, and their optimization with the help of ontologies. It may support the development of algorithms, models, libraries, and simulation tools to minimize the logistics problems.

5.2 The top level of logistics ontologies

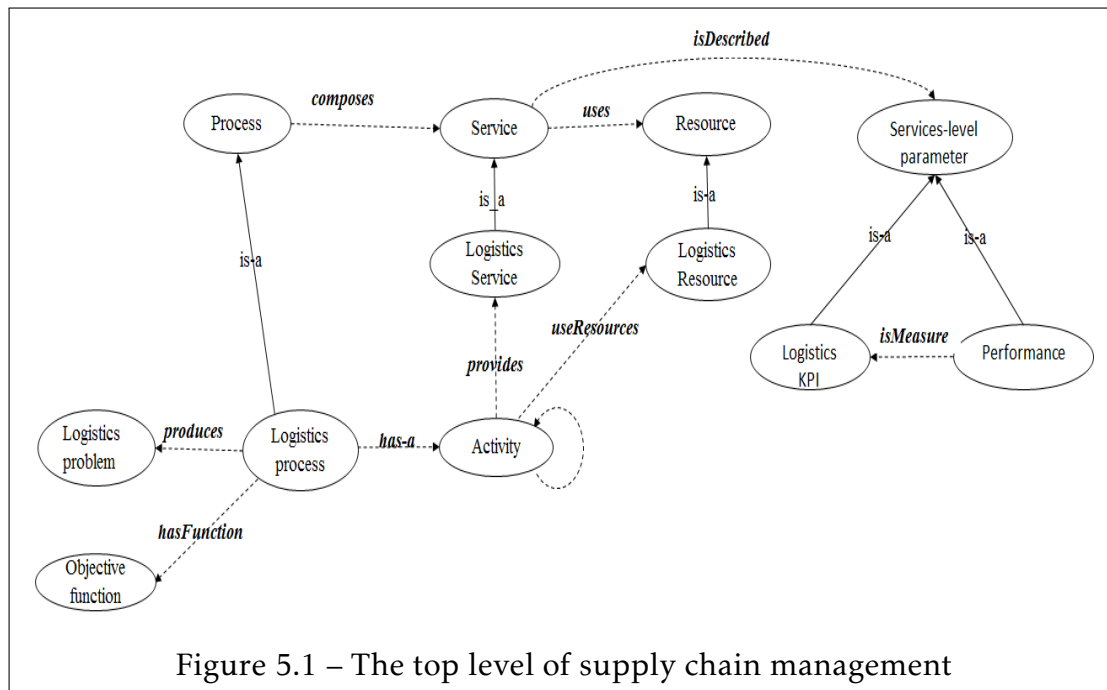
The construction of logistic ontologies depend on three things the Concepts or Classes, Relationships or Roles, and finally Axioms. In this chapter, we focus on modeling the general concepts of supply chain management (SCM), also the logistic problems generated by it, which may often be linked together to form a process with resources, services, and activities.

The basic top-level concepts (classes) include *Process*, *Service*, *Resource*, *Activity*, *Service Level Parameter*, and *Actor*, as shown in Figure 5.1.

The *Logistics Process* concept is a specialization of the *Process* class. In the same way, we define sub-classes for all top level classes such as *Logistics Process*, *Logistics Service*, *Logistics Resource*, *Performance*, and *Logistics KPI* are sub-classes of *Service Level Parameter* class.

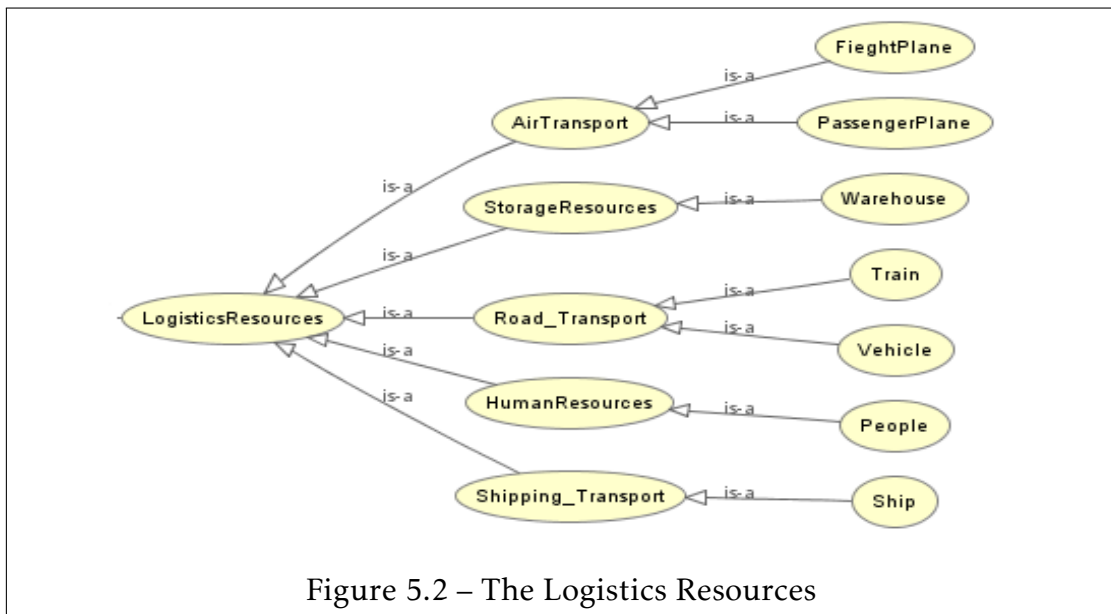
We further define the Roles (Relationships) or connecting classes along with data. The Roles are generally called properties. We distinguish the following property categories :

- The object property *composes* between the *Process* class and the *Service* class.
- The object property *uses* is *via* between the *Service* class and the *Resource* class.
- The object property *isDescribed* connects the *Service* class and the *Service-level parameter* class.
- The object property *produces* is *via* between the *Logistics Process* class and the *Logistics Problem* class.



- The object property *hasFunction* connects the *Logistics Process* class and the *Objective Function* class.
- The object property *hasActivity* connects the *Logistics Process* class and the *Activity* class.
- The object property *provides* is connects the *activity* class and the *Logistics Service* class.
- The object property *useResources* connects the *activity* class and the *logistics Resource* class.
- The object property *isMeasure* connects the *Performance* class and the *Logistics KPI* class.
- Data properties that links a class with data (string, Integer, Real or Boolean).
- The Axioms that provide the formal model correspondences using Description Logic (DL).

In the following sections, we describe formally each concept using DL.



5.2.1 The Logistic Resource concepts

The *Logistics Resource* concepts may appear on many categories such as *Air Transport*, *Storage Resources*, *Road transport*, *Shipping Transport*, and *Human Resources* as shown in figure 5.2.

These classes may contains one or more sub classes. For instance :

- The sub-classes of *Air Transport* are *Freight Plane* and *Passenger Plane*.
- The *Warehouse* is a sub-class of *Storage Resources*.
- The sub-classes of *Road Transport* are *Train* and *Vehicle*.
- The *Ship* is a sub-class of *Shipping Transport*.
- The *People* is a sub-class of *Human Resources*.

Each problem can use one or more resources concepts. In fact, the identification of a problem depends on the used resources, such as the vehicle routing problems can be identified by means resources like *Vehicle* and *Warehouse*.

5.2.2 The Actors

The figure 5.3 shows the specification of actors involved in various logistics processes. The *Actor* may be a *Service Provider*, *Service Requester*, *Company*, or *Participant* in the logistics chain.

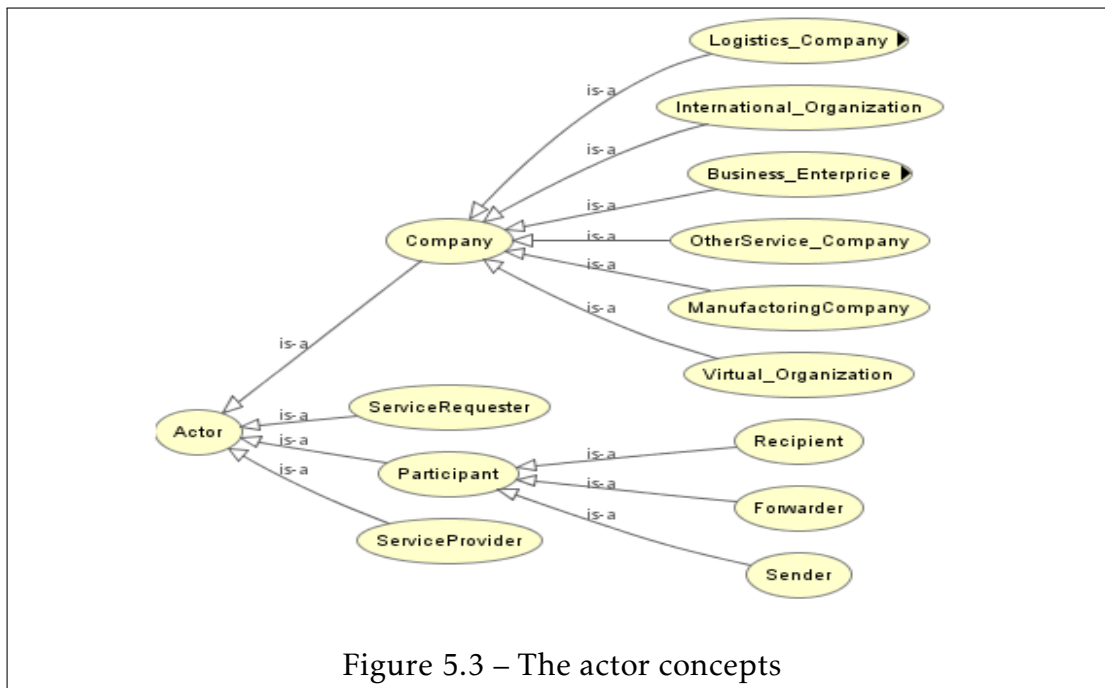


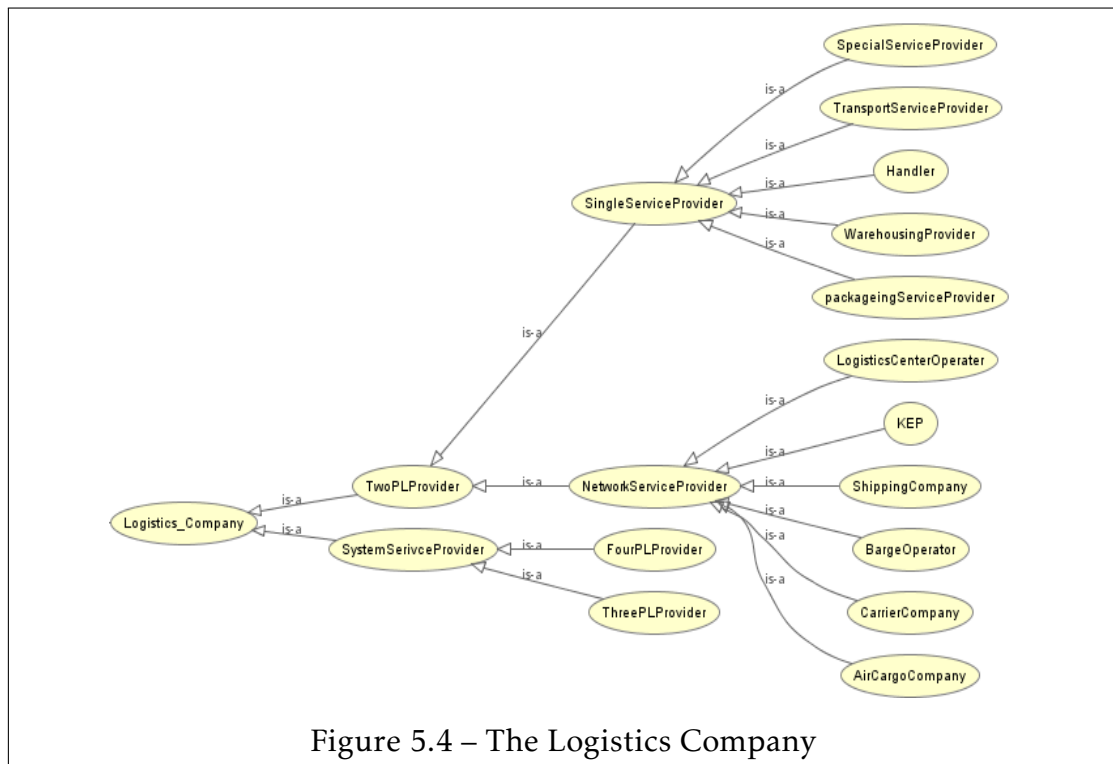
Figure 5.3 – The actor concepts

The *Recipient*, *Sender*, or *Forwarder* are sub-classes of the *Participant*. Moreover, every actor has a particular *Company Profile*, ranging from *Logistics Company* to *Business Enterprise*, *Manufacturing Company*, *Virtual Organization*, *International Organization*, or *Other Services Company*.

The figure 5.4 shows the variant of logistics companies. It includes the *Two PL Provider* and the *System Services Provider*. The *Two PL Provider* may contain the sub-classes *Network Service Provider* and *Single Service Provider*.

- The *Network Service Provider* includes the *Shipping Company*, *Carrier Company*, *KEP*, *Logistics Center Operator*, *Barge Operator*, and the *Air Cargo company*.
- The *Single Service Provider* may include *Warehousing Provider*, *Handler*, *Special Service Provider*, *package Service Provider* and *Transport Service Provider*.

The *System Services Provider* includes *Four PL Provider* and *Three PL Provider*.

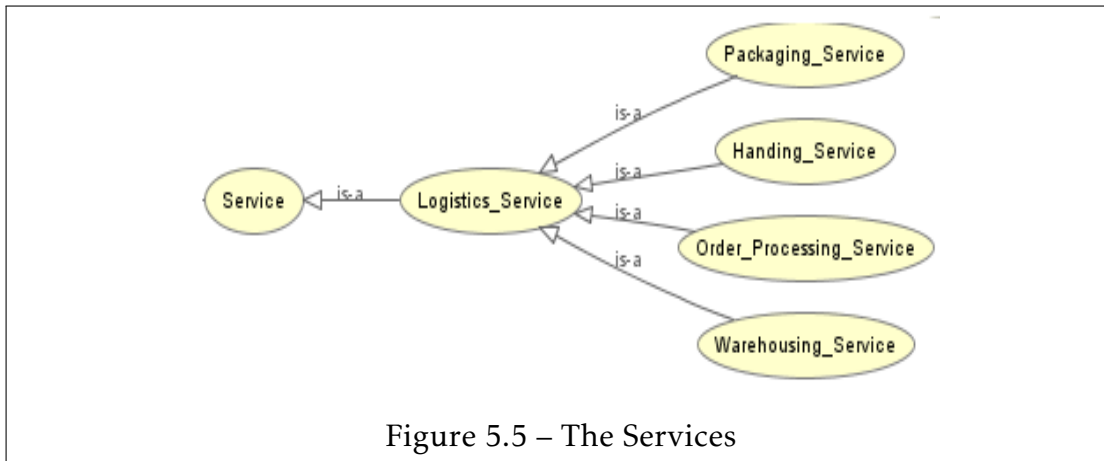


5.2.3 The Service Concepts

The *Service* is refined by *Logistics Service* class which is further refined by many sub-classes including *Package Service*, *Handing Service*, *Order Processing Service*, and *Warehousing Service*, etc. as shown in figure 5.5

Likewise, there are many object properties or roles connecting these concepts (classes). These roles include :

- The *composes* role has the domain *process* class and range is the *Service* class.
- The *hasProvider* role has the domain *Service provider* and the range is *Service* class.
- The *hasRequester* role has the domain *Service Requester* and the range is *Service* class.
- The *isDescribed* role has the domain *Service* and the range is *Service Level parameter* class.
- The *usesResource* role has the domain *Service* and the range is *Resource*



class.

In the following, we present the axioms defined by using these concepts (with the help of Description Logic :

Axiom 5.1: The *Logistics Services* is a sub-class of *Services* , which has the role *ProvidedBy* of *Activity*. It can be formulated as follows :

$$\begin{aligned}
 \text{LogisticsServices} \subseteq & \text{Services} \cap \exists \text{providedBy}.\text{Activity} \\
 & \cap \exists \text{hasRequester}.\text{ServiceRequester} \cap \exists \text{hasProvider}.\text{ServiceProvider}
 \end{aligned}$$

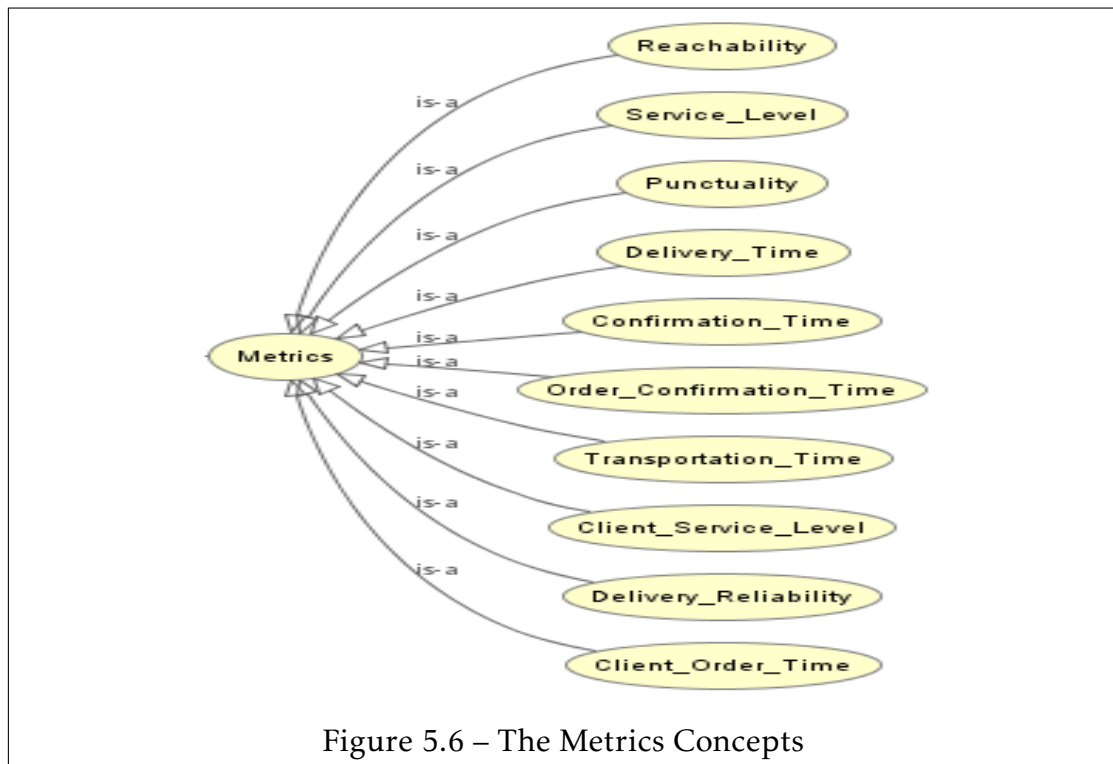
5.2.4 The Services Level Parameter

The services level parameter concepts include the sub-classes that are *Performance* and *Logistics KPI Metrics*. Likewise, there is an object property *isMeasureBy* between the Performance class and KPI Metrics class. For instance the performance is Measured by KPI Metrics.

Axiom 5.2: The *Performance* is a sub-class of *The Services Level Parameter*, and *isMeasureBy* of *KPI Metrics*. It can be formulated as follows :

$$\text{Performance} \subseteq \text{TheServicesLevelParameter} \cap \forall \text{isMeasureBy}.\text{KPI} \text{Metrics}$$

The figure 5.6 shows the Metrics Concepts (*Logistics KPI Metrics*). It contains various metrics that are used to measure the performance such as *Reachability*,

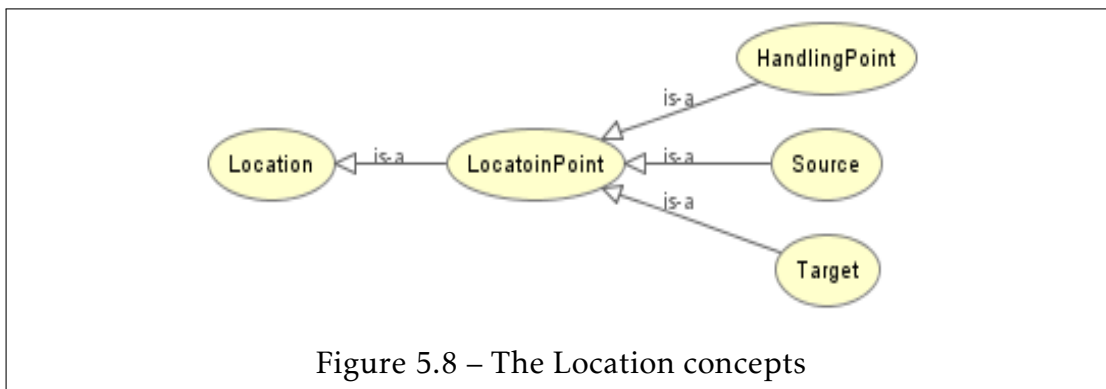
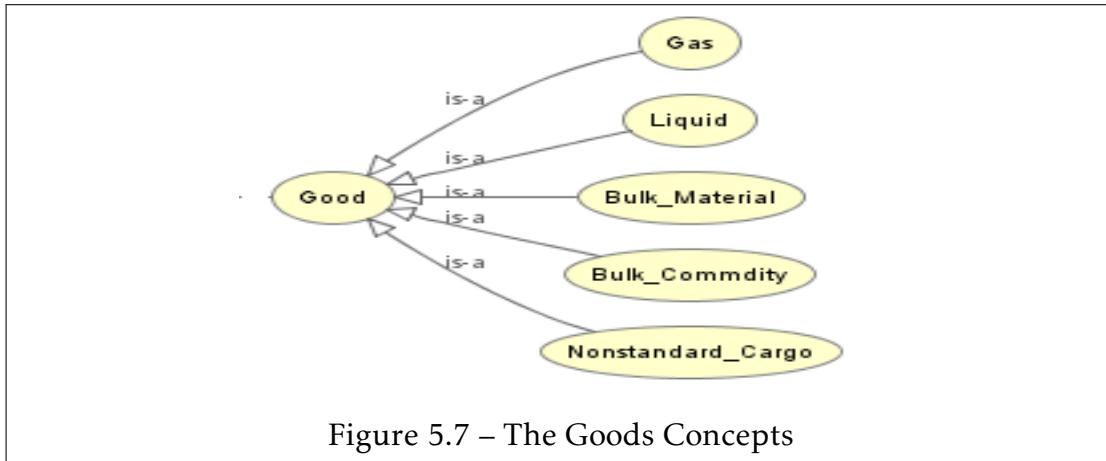


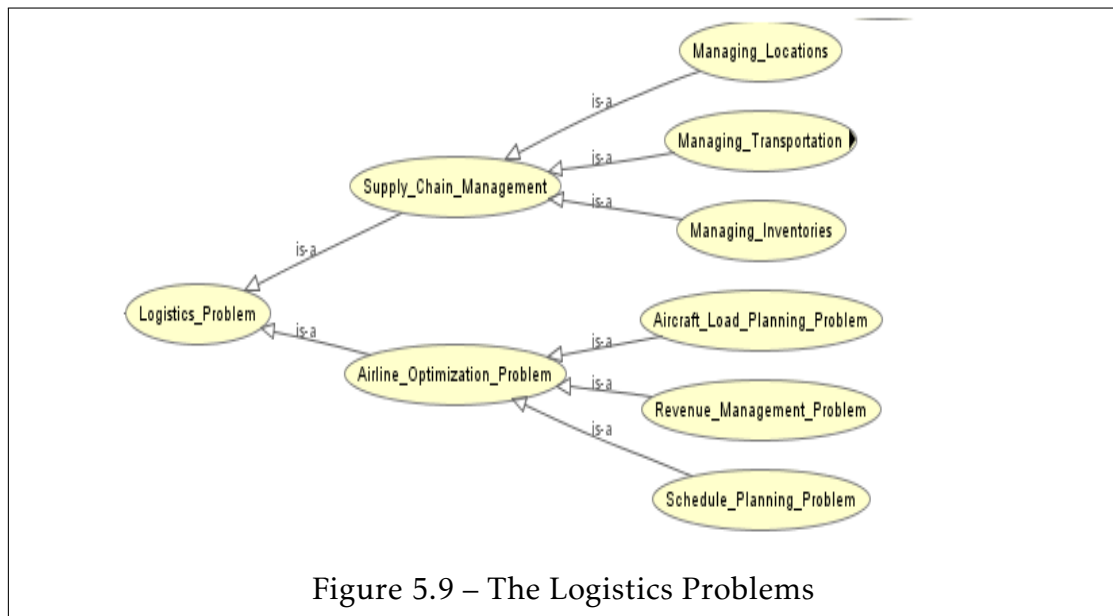
Services Level, Punctuality, Delivery Time, Confirmation Time, Order Confirmation Time, Transportation Time, Client Services Level, Delivery Reliability, and Client Order Time.

5.2.5 Other Logistic concepts

We also define some logistic ontologies, which are not necessarily part of the top level logistic ontologies, but still are useful in our subsequent work. We define the concepts to exhaustively capture the information of logistic domain. They vary from the different *types of carriers, goods, logistics documents, and logistics standards*. The figure 5.7 shows the *Goods* class, which include sub-classes such as *Gas, Liquid, Bulk Material, Bulk Commodity, and Non Standard Cargo*.

The figure 5.8 shows the *Location* class. It has the sub-classes *Source location, Target location, and Handling Point*.





5.3 The Logistic Optimization Problems

The logistics system has many resources that provide the services for both the customers and suppliers. The logistics system may manage these resources in an optimal way. Therefore, these can be subjected to produce the optimization problems. These problems mainly requires to perform the optimized activities for resource management and services provisions, with respect to the minimum cost and time.

Among others, the wide spread logistic problems can be categorized as the *Transport Problems* and *airline schedule problems*. These further integrate the resource of logistic optimization problems such as the *supply-chain management* and *people management*, etc. Similarly, the logistic optimization problems can be classified into three major categories depending on the type of problem in [49], as shown in figure 5.9. **Axiom 5.3:** The *Logistics Optimization Problem* includes three sub-classes : *Airline Optimization Problems* , *People Management Problems*,

and *Supply Chain Management Problems*. It can be formulated as follows :

$$\begin{aligned} \text{LogisticsOptimizationProblems} &\equiv \text{AirlineOptimizationProblems} \\ &\cup \text{PeopleManagementProblems} \\ &\cup \text{SupplyChainManagementProblems} \end{aligned}$$

The People management problem consists of allocating human resources to activities with minimal cost. The *Airline optimization problems* may be categorized in different fields such as *Aircraft Load Planning Problem*, *Revenue Management Problem* and *Schedule Planning problems*. For instance, *Aircraft Load Planning* consists of planning the transportation of equipment and persons. Additionally, the person transportation must involve a seat availability control system [49].

Axiom 5.4: The *Airline Optimization Problems* includes three sub-classes: *AirCRAFT Load Planning Problems*, *Revenue Management Problems*, and *Schedule Planning Problems*.

$$\begin{aligned} \text{AirlineOptimizationProblems} &\equiv \text{AirCRAFTLoadPlanningProblems} \\ &\cup \text{RevenueManagementProblems} \\ &\cup \text{SchedulePlanningProblems} \end{aligned}$$

The supply chain management is a vast domain covering problems related to *transport, location, inventories, vehicle routing*, etc.

Axiom 5.5: The *Supply Chain Mangement Problems* includes three sub-classes: *location Mangement Problems*, *Transportation Problems* and *Inverentories Mangement Problems*. It can be formulated as follows :

$$\begin{aligned} \text{SupplyChainMangementProblems} &\equiv \text{LocationMangementProblems} \\ &\cup \text{TransportationProblems} \\ &\cup \text{InverentoriesMangementProblems} \end{aligned}$$

In this work, we focus on the *Transportation Problems*. These can include many variant such as the *Railways Transportation Problems*, the *Road Transportation Problems*, and the *Marine Transportation Problems*. We choose three types of

transportation problems to validate the proposed model. These are the *Vehicle Routing Problem (VRP)*, the *Passenger Train Problems (PTP)* and the *Container terminal Problems (CTP)*.

In the following sections, we discuss the details of each of these problems. In addition, we define some more precise concepts, roles and data properties to better illustrate each problem. Finally, we define the various axioms concerning each problem.

5.3.1 The Vehicle Routing Problems

The *Vehicle Routing Problem (VRP)* attempts to determine the optimal set of routes to be adopted by a fleet of vehicles in order to serve a given set of customers constrained by the cost and time. The VRP is a combinatorial optimization problem [64].

The general idea for vehicle routing problem is how to deliver freights to many locations for different clients with least cost. For instance, the logistic companies receive one or more request orders from each client and more than thousand locations. The order may define a fixed time window for loading (sources) and for delivery (destination) such as $[t_s^1, t_s^2]$. The t_s^1 is the early time to Load the goods and t_s^2 is the lately time to lead the goods. The order is to be collected from source location $l_s \in L$ to destination location $l_d \in L$. The order must arrive within a time window $[t_d^1, t_d^2]$. The order has a volume V which is assumed to be an integer multiple of the capacity of a swap body.

The equation 5.1 describes the parameters of order.

$$o = \langle l_s, l_d, [t_s^1, t_s^2], [t_d^1, t_d^2], v \rangle \quad (5.1)$$

There is a planning process called *Tour Process*. All orders are intern in this process, and the result can be defined as a set of tours denoted as X . The single tour x may be defined as in equation 5.2.

$$x = \langle l_s, l_d, t^1, t^2, \underline{b}, \underline{q} \rangle \quad (5.2)$$

l_s and l_d are the start and destination locations and t^1 and t^2 are the departure

and arrival time of the vehicle. Also, each tour can include a set of freight $\underline{b} = \{b_1, b_2, \dots\}$ of swap bodies, and it may contain the orders $\underline{o} = \{o_1, o_2, \dots\}$. It is assumed that each vehicle can have at least one driver. The *Tour* are the smallest units of freight transportation. Usually, multiple tours are combined for a delivery. The VRP refers to an optimized use of logistics resources. The major relevant constraints oblige to respect the minimum cost and time consumption.

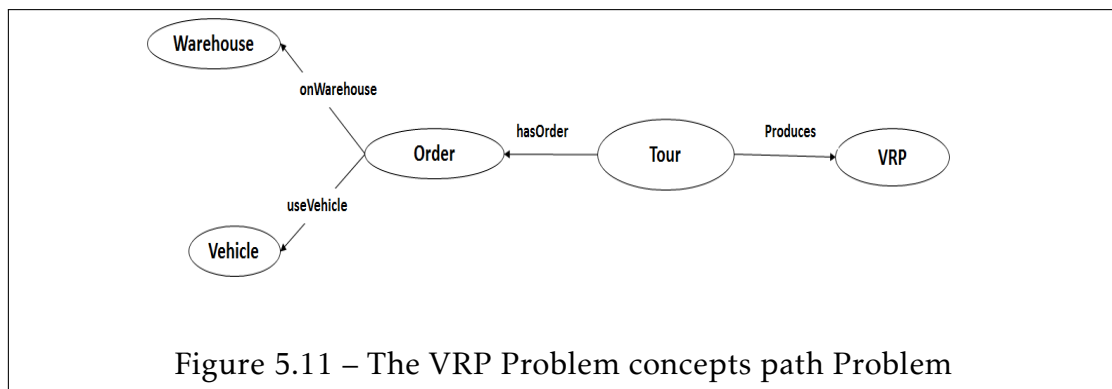
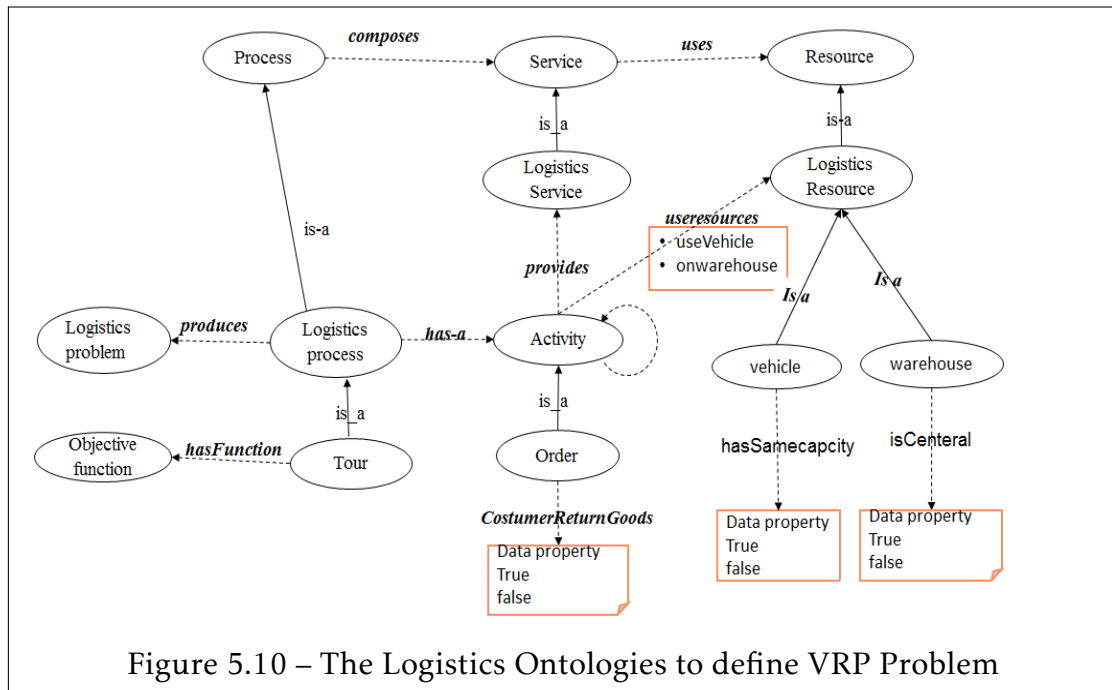
In the following, we discuss the variants of VRP [70]. It may give better understanding of the shared knowledge domain of logistics and optimization.

- The *Capacity Vehicle Routing Problem (CVRP)* is a VRP where a fixed fleet of vehicles respect the uniform capacity. In this problem, customer demands are served from a common depot at minimum transit cost.
- The *Vehicle Routing Problem with Time windows (VRPTW)* is a VRP with the restriction that every order must be delivered in a fixed time interval.
- The *Distance Vehicle Routing Problem (DVRP)* is a VRP with two objective functions :
 1. minimizing the cumulative travel distance of all vehicles and
 2. maximizing the travel distance of each vehicle.
- The *Multiple Depot Vehicle Routing Problems (MDVRP)* is a VRP where commodities can be supplied to customers from several depots.
- The *Vehicle Routing Problem with Pick-up and Delivering (VRPPD)* is a CVRP where each customer delivered some commodities with the possibility to return other commodities to the deliverer [50].

Initially, we may add more concepts and roles to logistics ontologies such as shown in figure 5.10, these can specify various Vehicle Routing Problems (VRP) such as *Order*, *Tour*, and *Objective Function*.

Likewise, As shown in figure 5.11. we define some roles that connect the concepts of VRP. These are as follows :

- The *requestedBy* between the *Order* and *ServicesRequester*.
- The *onWarehouse* between the *Warehouse* class and *Order* class.
- The *useVehicle* role between the *Vehicle* class and *Order* class.
- The *hasOrder* role between the *Order* class and *Tour* process.
- The *produces* role between the *Tour* class and *VRP* problem.



— Finally, the *hasObjectiveFunction* between *Tour* and *ObjectiveFunction* class.

Therefore, we define the axioms between related concepts, such as the following: **Axiom 5.6:** The *Order* is a sub-class of *Activity* class. Every *order* has a *requestedBy* role that is of the *ServicesRequester* class. It can be formulated, as follows :

$$Order \subseteq Activity \cap \exists requestedBy.ServicesRequester$$

Axiom 5.7: The *ServicesRequester* is a sub-class of *Actor* with *request* role referring some *Orders*. It can be formulated as follows :

$$ServicesRequester \subseteq Actor \cap \exists request.Order$$

Axiom 5.8: The *Tour* is a sub-class of *Logistics Process* class. Every *Tour* has a role *hasOrder* referring some *orders* and a role *hasObjectiveFunction* of class *Objective Function* and a role *produce* that is a collection of *Logistics Optimization Problems*. It can be formulated as follows :

$$\begin{aligned} Tour \subseteq & LogisticsProcess \cap \exists hasOrder.Order \\ & \cap \exists hasObjectiveFunction.ObjectiveFunction \\ & \cap \exists produce.LogisticsOptimizationProblems \end{aligned}$$

Axiom 5.9: The *Warehouse* is sub-set of *Logistics Resource* and has a role *contains-Good* of the class *Good*. It can be formulated as follows :

$$Warehouse \subseteq LogisticsResource \cap \exists containsGood.Good$$

Axiom 5.10: *Vehicle Routing Problems* is a sub-class of *Transportation Problems* class. It can be formulated, as follows :

$$VehicleRoutingProblems \subseteq TransportationProblems$$

We need to define some more data properties to further define the variants of the VRP. These are as follows :

- The *isCentral* define the fact that the *Warehouse* is *Central* (only one warehouse is used or not).
- The *sameCapacity* define the fact that the vehicles has the *same capacity* or not.
- *customerReturnGoods* is associated to *Orders* to define the fact that *Customers* can or can not return goods following a delivery order.

In the following, We show axioms defining the different variants of the Vehicle Routing Problem.

Axiom 5.11: The *Classical Vehicle Routing Problem (CLVRP)* is a *Vehicle Routing Problem* produced by the *Tour* process. The *Tour* has some *Order* and an *Objective function* which is to minimize the cumulative distance. Moreover, each *Order* uses a *Vehicle* and a central *Warehouse*. It can be formulated as follows :

$$\begin{aligned}
 CLVRP \subseteq & VehicleRoutingProblem \\
 & \cap \exists producedby.(Tour \\
 & \cap \exists hasOrder.(Order \\
 & \cap \exists useVehicle.Vehicle \\
 & \cap \exists onWarehouse.(Warehouse \\
 & \cap isCentral.\tau)) \\
 & \cap \exists hasObjectiveFuction.minDistance)
 \end{aligned}$$

Axiom 5.12: The *Capacity Vehicle Routing Problem (CVRP)* is a *Vehicle Routing Problem* having the same characteristics as the classical VRP with an additional constraint concerning the *vehicles* which must have the *same capacity*. It can be

formulated as follows :

$$\begin{aligned}
CVRP \subseteq & VehicleRoutingProblem \\
& \cap \exists producedby.(Tour \\
& \cap \exists hasOrder.(Order \\
& \cap \exists useVehicle.(Vehicle \\
& \cap sameCapacity.\tau) \\
& \cap \exists onWarehouse.(Warehouse \\
& \cap isCentral.\tau)) \\
& \cap \exists hasObjectiveFuction.minDistance)
\end{aligned}$$

Axiom 5.13: The *Multiple Depot Vehicle Routing Problem (MDVRP)* is a *Vehicle Routing Problem*. It is produced by the *Tour* process. The *Tour* has some *Orders* and an *Objective function* that is to minimize the distance. Each *Order* uses *Vehicles* and *Warehouses*. It can be formulated as follows :

$$\begin{aligned}
MDVRP \subseteq & VehicleRoutingProblem \\
& \cap \exists producedby.(Tour \\
& \cap \exists hasOrder.(Order \\
& \cap \exists useVehicle.Vehicle \\
& \cap \exists onWarehouse.Warehouse) \\
& \cap \exists hasObjectiveFuction.minDistance)
\end{aligned}$$

Axiom 5.14: The *Vehicle Routing Problem Pickup and Delivery (VRPPD)* is a *Capacity Vehicle Routing Problem* along with those orders which allow costumers to

return some Goods. It can be formulated as follows :

$$\begin{aligned}
 VRPPD \subseteq & VehicleRoutingProblem \\
 & \cap \exists producedby.(Tour \\
 & \cap \exists hasOrder.(Order \\
 & \cap \exists costumerReturnGoods.\tau \\
 & \cap \exists useVehicle.(Vehicle \\
 & \cap sameCapacity.\tau) \\
 & \cap \exists onWarehouse.Warehouse) \\
 & \cap \exists hasObjectiveFuction.minDistance)
 \end{aligned}$$

5.3.2 The Passenger Train Problems

The railway planning is a complex activity. It is usually succession of stages such as traditional network design, line design, time-tabling, rolling stock and finally crew management [6]. The railway system and the operational processes are rich in challenging Combinatorial Optimization problems. Many existing optimization methods can be used to solve these problems [10].

In general, the Railway transportation Problems can be classified, into the following categories :

- Passenger Train Problems
- Cargo (Freight) Transportation Problems

We focus to analyze the Passenger Train Problems (PTP) for the sake of illustration. The complexity of passenger railway systems leads to decompose it into sequential phrases, such as :

- Line Planning : it helps to decide the trips for the passenger trains. It may also help to decide the types and frequencies of the trains on each trip.
- Time-tabling : it helps to fix the timetable for each train.
- Rolling Stock Circulation : it helps to decide the assignments of train units (locomotives and train carriages) to the trains. Each train has his own timetable and platforms.
- Crew Planning : it helps to allocate the workload of train drivers and

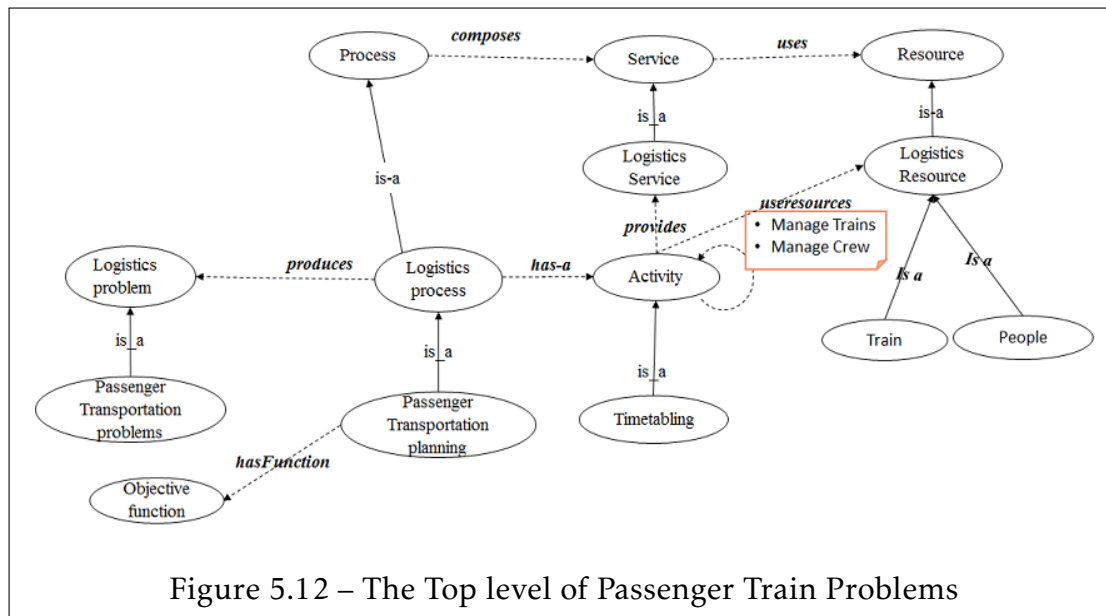


Figure 5.12 – The Top level of Passenger Train Problems

conductors to operate a given timetable.

The figure 5.12 shows the concepts of PTP problems [24], as follows :

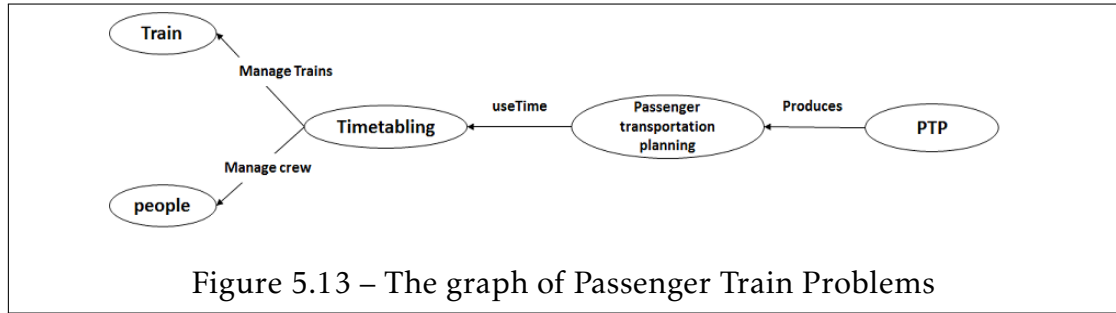
- The *timetabling* class is subset of the *activity* class.
- The *Passenger Transportation pLanning* (*PTL*) class is sub set of the *logistics process*
- The *Passenger Train Problems* (*PTP*) class is subset of the *logistics problems*.

We continue to define relations between these concepts, as follows :

- The *ManageTrain* is sub-role of *UseResources* that connects the *Timetabling* activity and the *Train* classes.
- The *ManageCrew* is sub-role of *UseResources* that connects the *People* and the *Timetable* classes.
- The *hasTime* is sub-role of *has-a* role that connects the *Passenger transportation planning* process and the *Timetable* activity.
- The *hasObjectiveFunction* is a role that connects the *Passenger Transportation Planning* and the *Objective Function* classes.

Finally, we define axioms between the related concepts and variants of PTP such as the followings :

Axiom 5.15: The *Train* is sub-set of *Logistics Resource Concepts* (*RC*) and it



exits *TravelTime* in *Timetabling*. It can be formulated as follows :

$$Train \subseteq LogisticsResource \cap \exists TravelTime.Timetabling$$

Axiom 5.16: *People* is a sub-set of *Logistics Resource Concepts* and it exits *WorkTime* in *Timetabling*. It can be formulated as follows :

$$People \subseteq LogisticsResource \cap \exists WorkTime.Timetabling$$

Axiom 5.17 *Passenger Train Problems* is a sub-class of *Railway Transportation Problems* class. It can be formulated as follows :

$$PTP \subseteq RailwayTransportationProblems$$

Axiom 5.18: *Passenger Transportation planning* is a sub-class of *Logistics Process*. Every PTL has a role *hasTime* into the *Timetabling* activity. It has the role *hasObjectiveFunction* to *Objective Function* class. It also has a role *produces* that is a collection of *Passenger Train Problems*, as shown figure 5.13. It can be formulated as follows :

$$PTL \subseteq LogisticsProcess \cap \exists hasTime.Timetabling \\ \cap \exists hasObjectiveFunction.ObjectiveFunction \\ \cap \exists produces.PTP$$

We may identify some variants of Passenger Train Problems, as follows :

- Rolling Stock Problem

- Crew Scheduling Problem
- Passenger Train Timetabling Problems
- Line Train Planning Problem

In the following, we further define these variants individually.

Axiom 5.19: The *Rolling Stock Problem (RSP)* is a sub-set of *Passenger Train Problems (PTP)*. It is produced by the *Passenger Transportation planning (PTL)*. The *Passenger Transportation planning (PTL)* manages the time into *Timetabling* activity. It has the *Objective functions* to *minimize the total cost and maximize the number of passengers*. It can be formulated as follows :

$$\begin{aligned}
 RSP \subseteq & \text{PassengerTransportationProblems}(PTP) \\
 & \cap \exists \text{producedby.}(PTL) \\
 & \cap \exists \text{hasTime.}(Timetabling) \\
 & \cap \exists \text{ManageTrain.}(Train) \\
 & \cap \exists \text{ManagePeople.}(People) \\
 & \cap \exists \text{hasObjectiveFunction.}(MinTravelCost \\
 & \cap \text{MaxNumberPassenger))
 \end{aligned}$$

Axiom 5.20: The *Crew Scheduling Problem (CSP)* is a sub-set of *Passenger Train Problems (PTP)*. It is produced by the *Passenger Transportation planning (PTL)*. The *Passenger Transportation planning* manages the time into *Timetabling* activity. It has the *Objective function* to *minimize the number of staff*. It can be formulated as

follows :

$$\begin{aligned}
CSP \subseteq & PassengerTransportationProblems(PTP) \\
& \cap \exists producedby.(PTL \\
& \cap \exists hasTime.(Timetabling \\
& \cap \exists ManageTrain.Train \\
& \cap \exists ManagePoepple.People) \\
& \cap \exists hasObjectiveFunction.MinNumberStaff)
\end{aligned}$$

Axiom 5.21: The *Passenger Train Timetabling Problem (PTTP)* is a sub-set of *Passenger Train Problems (PTP)*. It is produced by the *Passenger Transportation planning (PTL)*. The *Passenger Transportation planning (PTL)* manages the time into *Timetabling* activity. It has the *Objective function* to minimize wait time for passengers. It can be formulated as follows :

$$\begin{aligned}
PTTP \subseteq & PassengerTransportationProblems(PTP) \\
& \cap \exists producedby.(PTL \\
& \cap \exists hasTime.(Timetabling \\
& \cap \exists ManageTrain.Train \\
& \cap \exists ManagePoepple.People) \\
& \cap \exists hasObjectiveFunction.MinWaitTimePassenger)
\end{aligned}$$

Axiom 5.22: The *Line Train Planning Problem (LTPP)* is a sub set of *Passenger Train Problems (PTP)*. It is produced by the *Passenger Transportation planning (PTL)*. The *Passenger Transportation planning (PTL)* has to manage the time into *Timetabling* activity. It has the *Objective function* to minimize travel cost. It can be formulated as follows :

$$\begin{aligned}
<TPP \subseteq PTP) \cap \exists \textit{producedby}.(PTL \\
&\quad \cap \exists \textit{hasTime}.(\textit{Timetabling} \\
&\quad \cap \exists \textit{ManageTrain}.Train \\
&\quad \cap \exists \textit{ManagePoeple}.People) \\
&\quad \cap \exists \textit{hasObjectiveFunction}.MinTravelCost)
\end{aligned}$$

5.3.3 The Container Terminal Problems

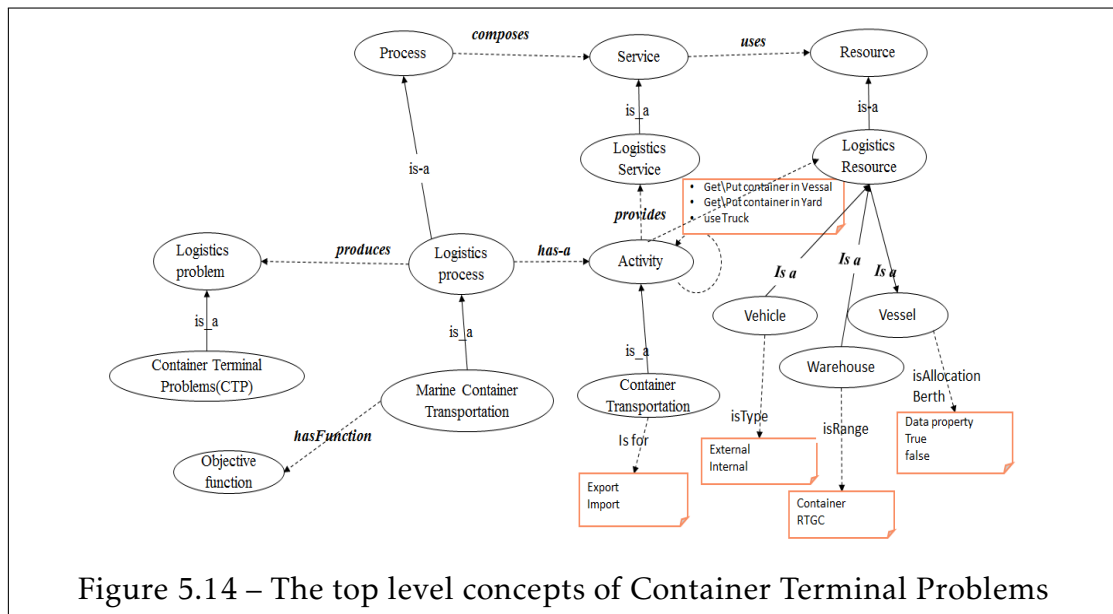
The container terminals include three crucial resources such as the *Yard*, *Vessel berths*, and the *Trucks*. The main objective of port terminals is the efficient use of these resources while performing different operations. The yard is a temporary container warehouse where containers are stocked until their transport to their next location by trucks, trains, or vessels. The Containers are stacked on top of each other, in order to utilize the yard space efficiently [21].

However, the stacking crane can only directly access the containers at the top of the stack. In order to extract another container stored underneath, it need to make a reshuffle/shift. Such resuffly/shift occurs are defined as the unproductive moves of the containers.

For instance, we focus to increase the efficiency of the yard *via* the consideration of the container stacking optimization problem for trans-shipment, inbound, and outbound containers at a container terminal. The objective of the problem is to minimize container storage and retrieval times through avoidance of reshuffles. It may result in more efficient loading/unloading operations, and in turn minimize the dwell time of containers. The main inputs are the type, weight, discharge port/location, destined vessel/vehicle of the container, and the expected departure time [21].

The figure 5.14 shows the set of classes (or concepts), relationships (roles), and the axioms concerning the CTP [25]. We define concepts of CTP, as follows:

- The *Container Transportation* is a sub-class of *Activity* class.
- The *Marine Container Transportation* is a sub-class of *Logistics Process*.



— The *Container Terminal Problems (CTP)* is a sub-class of *Logistics Problems*.

Similarly, we define the following relationships (or roles) :

- The *Get/Put Container in vessel* is a sub-role of *UseResources* that connects the *Container Transportation* activity and the *Vessel* class.
- The *Get/Put Container in Yard* is a sub-role of *UseResources*) that connects the *Warehouse*.
- The *useTruck* is a sub role of *UseResources* that connects the *Vehicle* class and *Container Transportation* activity.
- The *hasContainer* is a sub-role of *has-a* that connects the *Marine Container Transportation* process and the *Container Transportation* activity.
- The *hasObjectiveFunction* is a role between *Marine Container Transportation* and *Objective Function*.

Furthermore, we define some axioms to better understand the related concepts and variants of CTP, as follows :

Axiom 5.23: The *Berth and Quay Crane Allocation Problem (BQCAP)* is a *Container Terminal Problems (CTP)*. It is produced by the *Marine Container Transportation* process. The *Marine Container Transportation* has some *Container Transportation*. It has an *Objective function* that is to minimize the Loading Time

of vessel. Each *Container Transportation* uses the *Vehicle* and the *Warehouse*, therefore, the allocation *Vessel* is true. It can be formulated as follows :

$$\begin{aligned}
& BQCAP \subseteq CTP \\
& \cap \exists \text{producedby.}(\text{MarineContainerTransportation} \\
& \cap \exists \text{hasContainer.}(\text{ContainerTransportation} \\
& \cap \exists \text{useTruck.Vehicle} \\
& \cap \exists \text{Get/PutContainerInYard.Warehouse} \\
& \cap \exists \text{Get/PutContainerInVessel.}(\text{Ship} \\
& \cap \text{isAllocationInBerth.true})) \\
& \cap \exists \text{hasFuction.minLoadTime}
\end{aligned}$$

Axiom 5.24: The *Container Re-shuffling Problem (CRP)* is a *Container Terminal Problem (CTP)*. It is produced by the *Marine Container Transportation* process. The *Marine Container Transportation* has some *Container Transportation*. It has an *Objective function* that is to minimize the number of movement of the containers. Each *Container Transportation* uses the *Vehicle* and the *Vessel*. The global function of *Warehouse* is to arrange the Containers. It can be formulated as follows :

$$\begin{aligned}
& CRP \subseteq CTP \\
& \cap \exists \text{producedby.}(\text{MarineContainerTransportation} \\
& \cap \exists \text{hasContainer.}(\text{ContainerTransportation} \\
& \cap \exists \text{useTruck.Vehicle} \\
& \cap \exists \text{Get/PutContainerInYard.}(\text{Warehouse} \\
& \cap \text{isRange.Container})) \\
& \cap \exists \text{Get/PutContainerInVessel.Vessel} \\
& \cap \exists \text{hasFuction.minReshufflingOfContainer}
\end{aligned}$$

Axiom 5.25: The *Rubber Tyred Gantry Crane Problem (RTGCP)* is a *Container Terminal Problem (CTP)*. It is produced by the *Marine Container Transportation* process. The *Marine Container Transportation* has some *Container Transportation*.

It has an *Objective function* that is to minimize Vessel Wait Time. Each *Container Transportation* uses the *Vehicle* and *Vessel*. The global function of *Warehouse* is to arrange RTGCs. It can be formulated as follows :

$$\begin{aligned}
 RTGCP \subseteq CTP & \\
 & \cap \exists \text{producedby.}(\text{MarineContainerTransportation} \\
 & \cap \exists \text{hasContainer.}(\text{ContainerTransportation} \\
 & \cap \exists \text{useTruck.} \text{Vehicle} \\
 & \cap \exists \text{Get/PutContainerInYard.}(\text{Warehouse} \\
 & \cap \text{isRange.RTGC})) \\
 & \cap \exists \text{Get/PutContainerInVessel.} \text{Vessel} \\
 & \cap \exists \text{hasFuction.minVesselWaitTime}
 \end{aligned}$$

Axiom 5.26: The *Scheduling and Routing Vehicle Problem (SRVP)* is a *Container Terminal Problems (CTP)*. It is produced by the *Marine Container Transportation* process. The *Marine Container Transportation* has some *Container Transportation*. It has an *Objective function* that is to minimize the Wait Time For (QC and RTGCs). Each *Container Transportation* uses a *Vessel* and *Warehouse*. The *Vehicle* is *Internal* type. It can be formulated as follows :

$$\begin{aligned}
 SRVP \subseteq CTP & \\
 & \cap \exists \text{producedby.}(\text{MarineContainerTransportation} \\
 & \cap \exists \text{hasContainer.}(\text{ContainerTransportation} \\
 & \cap \exists \text{useTruck.}(\text{Vehicle} \\
 & \cap \text{isType.Internal})) \\
 & \cap \exists \text{Get/PutContainerInYard.} \text{Warehouse} \\
 & \cap \exists \text{Get/PutContainerInVessel.} \text{Vessel} \\
 & \cap \exists \text{hasFuction.minWaitTimeOfQCandRTGC}
 \end{aligned}$$

Axiom 5.27: The *Appointment Time for Extranal Trucks Problem(ATETP)* is a *Container Terminal Problems (CTP)*. It is produced by the *Marine Container Trans-*

portationprocess. The *Marine Container Transportation* has some *Container Transportation*. It has an *Objective function* that is to minimize the Wait Time For External Trucks. Each *Container Transportation* uses a *Vessel* and *Warehouse*. The *Vehicle* is External type. It can be formulated as follows :

$$\begin{aligned}
ATETP \subseteq CTP & \\
& \cap \exists \textit{producedby} . (\textit{MarineContainerTransportation} \\
& \cap \exists \textit{hasContainer} . (\textit{ContainerTransportation} \\
& \cap \textit{isFor.Import}) \\
& \cap \exists \textit{useTruck} . (\textit{Vehicle} \\
& \cap \textit{isType.External})) \\
& \cap \exists \textit{Get/PutContainerInYard.Warehouse} \\
& \cap \exists \textit{Get/PutContainerInVessel.Vessel} \\
& \cap \exists \textit{hasFuction.minWaitTimeForExternalTrucks}
\end{aligned}$$

5.4 Conclusion

In this chapter, we attempt to define the logistics ontologies in order to better understand the different logistic problems. We initially define the top level of logistics ontologies. The core of the logistic ontology is composed of some general classes and roles. It comprises the logistic resource concepts, the actors, and logistic services. We later on extend the top level of logistic ontologies, according to the specific type of logistic problem. It is for this reason, we define some logistic optimization problems. We logically formulate the *Vehicle Routing Problems*, *Passenger Train Problems*, and the *Container Terminal Problems* to better illustrate the logistic problems. We can similarly, enrich the logistic ontology to adopt more specific logistic problems. This logistic ontology building process is inherent and may follow some simple iterations for ontologies integration. Thus,

it is possible to extend such an ontology making it kind of an evaluative object.

In the next chapter, we continue the same approach to define the optimization ontology. Their development is intended to define more precisely the optimization problems. Whilst, we attempt to better the manage of logistic problems.

The Optimization Ontology

6.1 Introduction

We have been working on the definition of logistic ontologies. Their basic purpose is define logistic problem and find the optimal solution. It is therefore, necessary to define the ontological schema for the optimization. However, the definition of optimization ontologies is a challenging task, with reference to the available literature on this subject. The logistic ontologies defined previously in chapter 5 need to be complemented with the optimization ontologies to complete our framework. In the proposed system, we intend initially, to identify the logistic problem with its precise parameters, and later, we subject the findings to exploit the available optimization methods. These optimization methods are linked with their implementation in form of web-services. We can then transfer the found logistic problem data to the web-services.

In this regard, the optimization ontology may be seen as extension, or more precisely a refinement of the logistics ontology intended to cope with the various logistic problems accordingly, during the execution of the logistics processes. The optimization ontology has been defined to assist the end users, in this case the developers, to investigate the optimization related knowledge and artifacts. This is achieved by the definition of common vocabularies, nomenclature, and taxonomies.

In chapter 2, we discussed some individual optimization ontologies that

mostly focus to classify the optimization concepts which are generally used to solve optimization problem. In this chapter, we define the general concepts of optimization and the relationships among these concepts. We also define the axioms of related concepts. Finally, we develop the knowledge base, in order to sufficiently cope the different types of logistics problem and solution methods. For instance, we consider to build the knowledge in reference to some Transportation Problems such as *Vehicle Routing Problems (VRP)*, *Passenger Train Problems (PTP)* and *Container Terminal Problems (CTP)* to validate the proposed approach.

6.2 Conceptualization of optimization ontologies

The Optimization Ontologies include the definitions for typical optimization problems along with the descriptions of the methods applied to solve an optimization task. The basic structure of this ontology support optimization processes. It also focuses on how to select and apply an optimization method to solve an optimization problem. The ontology concepts must eventually cover all entities that concur in an optimization task. Accordingly, we categorize optimization concepts as follows :

- The *Optimization Problem* class represents the problem to be solved. It is further classified into different sub-classes.
- The *Optimization Method* class represents the method used to solve the problem.
- The *Optimization Component* class represents the various parameters of the optimization problem.

The figure 6.1 explains dependencies among the *Top Level Optimization Ontologies*. The Optimization class basically extended by the optimization component, optimization problem, and optimization method classes, whereas the logistics problem class is addressed by optimization method class, in this regard.

Axiom 6.1 : *Optimization* may be either *OptimizationProblem* or *Optimization-Method* or *Optimization Component*. It can be shown as follows :

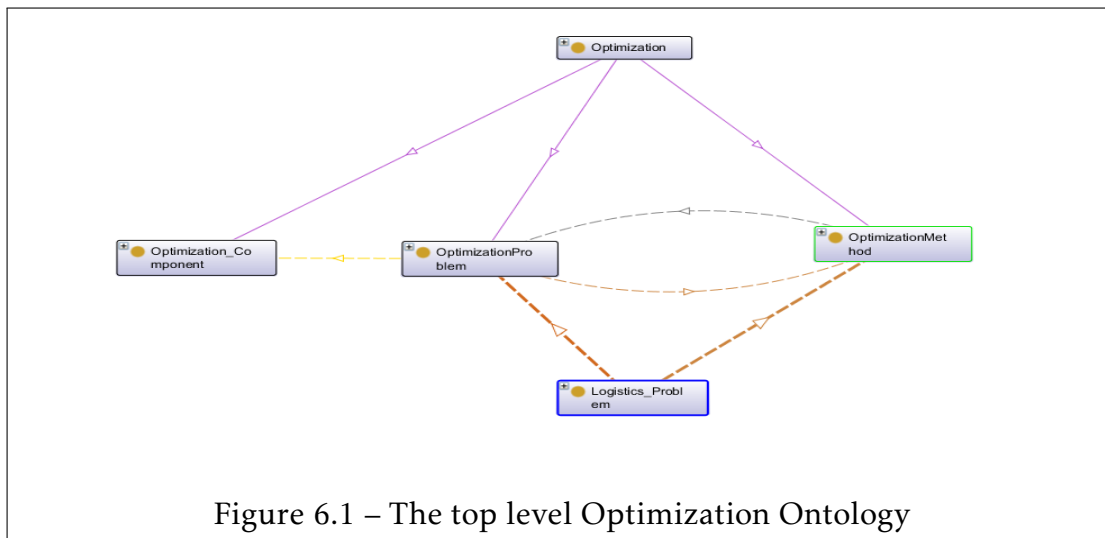


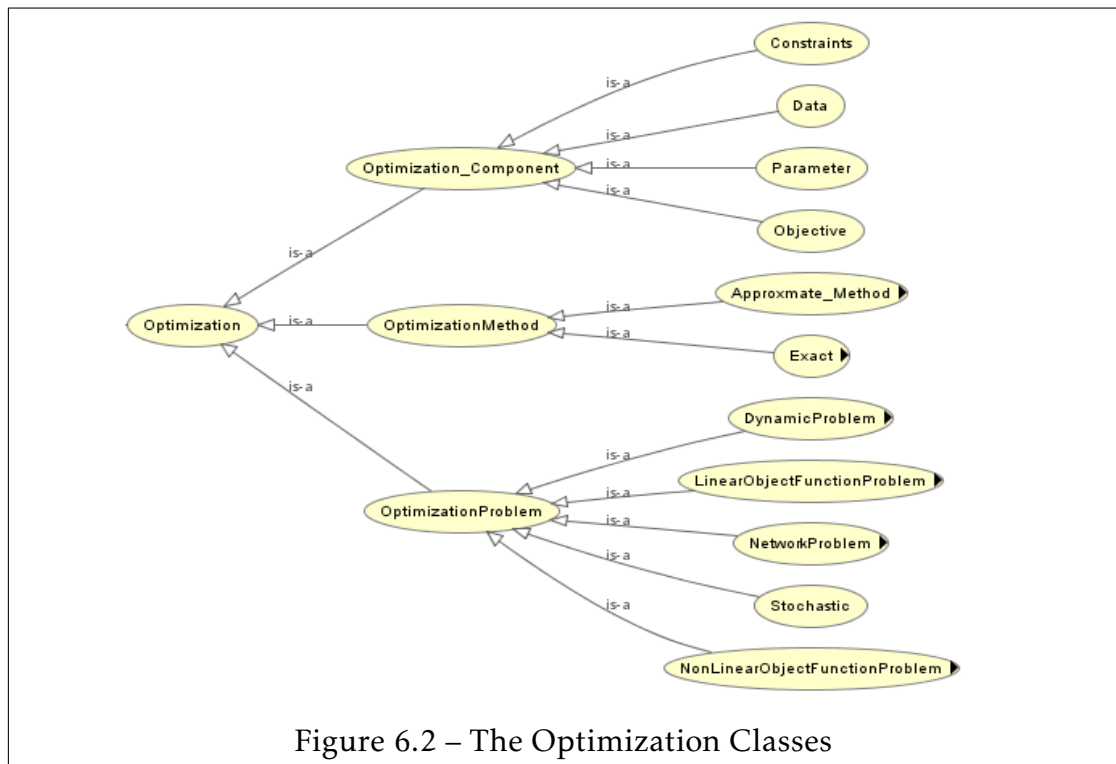
Figure 6.1 – The top level Optimization Ontology

$$\begin{aligned} \text{Optimization} \equiv & \text{OptimizationProblems} \cup \text{OptimizationMethods} \\ & \cup \text{OptimizationComponents} \end{aligned}$$

Axiom 6.2: The *OptimizationProblem* class is subset of *Optimization*. It has some components and there exists an *Optimization Method* to solve it. The *Logistics Optimization Problem* is subset of *Logistics Problem*. Its a type of *Optimization Problem* and there exists an *Optimization Method* to solve it. It can be formulated as follows :

$$\begin{aligned} \text{OptimizationProblem} \subseteq & \text{Optimization} \cap (\exists \text{SolveBy.} \text{OptimizationMethods}) \\ & \cap \exists \text{hasComponent.} \text{OptimizationComponent} \end{aligned}$$

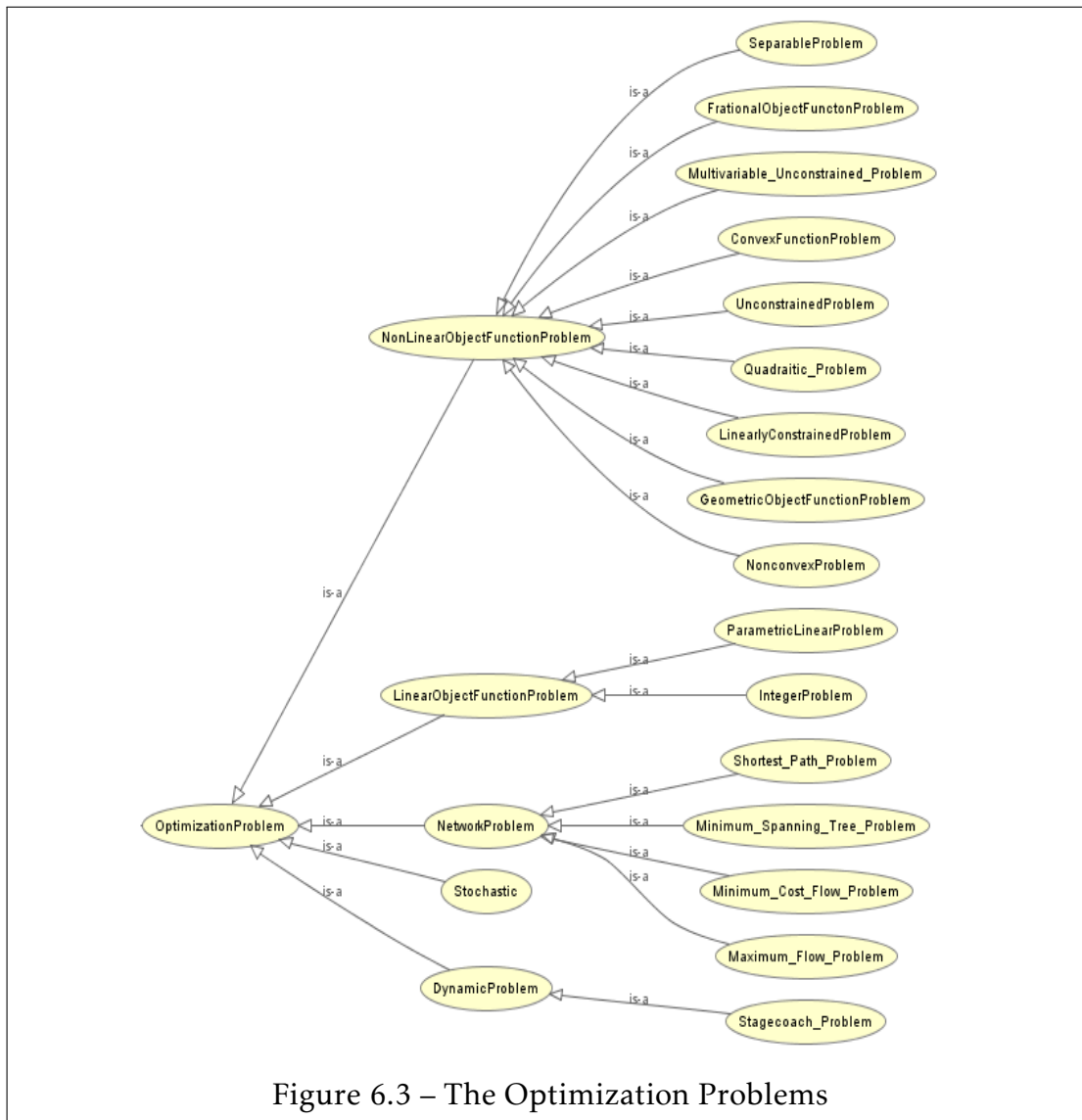
The figure 6.2 shows the general concepts of the optimization ontology. The root of the ontology is *Optimization* class. This root class has further sub-classes: *Optimization problem*, *Optimization Method*, and *Optimization Component*. In the sections below, we explain each of them.



6.3 The Optimization Components

The optimization problem is generally encompassed around multiple components. These components actually describe the nature of the optimization problem, depending on the component type. These components are notified as artifacts in our knowledge-base whereas, specifically their classification is as follows such as shown in figure 6.2:

- The *Data* contains some information needed by the optimization problem. For example the order for VRP problem or nodes for optimization methods, etc.
- The *Objective Function* is a mathematical function, it is used to find optimal solution. It depends on given data, actually the objective function consists maximum values or minimum values.
- Each optimization problem and methods has some *Constraints* for example the number of vehicles and the capacity each vehicle in VRP.
- The *Parameter* defines how to use resources to given optimal solving ways.



6.4 The Optimization Problems

The optimization problems may be classified into several categories depending on some factors. These may be :

- Continuous or Discrete,
- Unconstrained or Constrained, and
- Deterministic or Stochastic.

These problems may also be classified following the number of objective functions. The classification given in [27] inspired us to develop the optimization ontology presented in current work and in the underlying knowledge-base. The main concepts of this classification are shown in figure 6.3. These are :

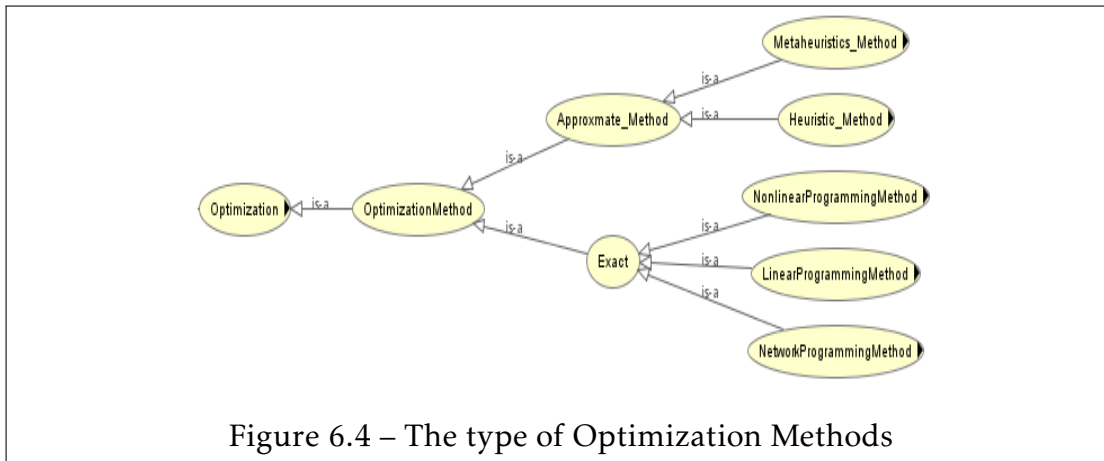
- Dynamic Problem
- Linear Objective Function Problem
- Non Linear Objective Function Problem
- Network Problem
- Stochastic problem

Each concept or class in top level includes other sub-classes. Such as :

- The *Dynamic Problem* is refined by the *Stagecoach Problem*.
- The *Linear Objective Function Problem* is refined by *Parametric Linear Problem and Integer*.
- The *Non Linear Objective Function Problem* can be classified into *Separable Problem, Fractional Object Function Problem, Multi-variable Unconstrained Problem, Unconstrained Problem, Convex Function Problem, Non Convex Problem Quadratics Problem, Linearly Constrained Problem, and Geometric Object Function Problem*.
- The *Network Problem* can be refined by *Shortest Path Problem, Minimum Spanning Tree Problem, Minimum Cost Flow Problem, and Maximum Flow Problem*.

The Logistics Optimization Problem can address the type of an Optimization Problem and also it can be solved by an Optimization Method.

Axiom 6.3 : The *Logistics optimization Problem* may be *Logistics Problems* and it is type of *Optimization Problem* solved by some *Optimization Method*. It can be formulated as follows :



$LogisticsOptimizationProblem \subseteq LogisticsProblem$

$\cap \exists isTypeOf.OptimizationProblem$

$\cap \exists SolveBy.OptimizationMethod$

6.5 The Optimization Methods

The optimization methods are the mathematical algorithms that can solve an optimization problem to produce a solution. This solution can be optimal or approximate (nearly optimal). The optimal solution, might take a lot of work/time to be found and some problems has limited time of execution. Therefore, we can use optimization method to find an approximate solution, in order to reduce time/work.

The optimization methods can be classified into exact or approximate methods, such as shown in figure 6.4.

Axiom 6.4: The *Optimization methods* may be *exact* or *approximated* ones. It can be shown, as follows :

$Optimizationmethods(OM) \equiv ExactMethods \cup ApproximatedMethods$

Axiom 6.5: An *approximate method* may be a *heuristic method* or a *meta-heuristic* one. It can be shown, as follows :

$$\text{ApproximatedMethods} \equiv \text{HeuristicMethods} \cup \text{Meta-heuristicMethods}$$

Axiom 6.6: A *heuristic method* may be an *improvement method* or a *construction method*. It can be shown, as follows :

$$\text{HeuristicMethods} \equiv \text{ImprovementMethods} \cup \text{ConstructionMethods}$$

The *Exact Methods* are shown in figure 6.5. They can have some type of constraints and these are classified into sub-classes, such as follows :

- Linear programming methods,
- Network programming Methods, and
- Nonlinear programming methods.

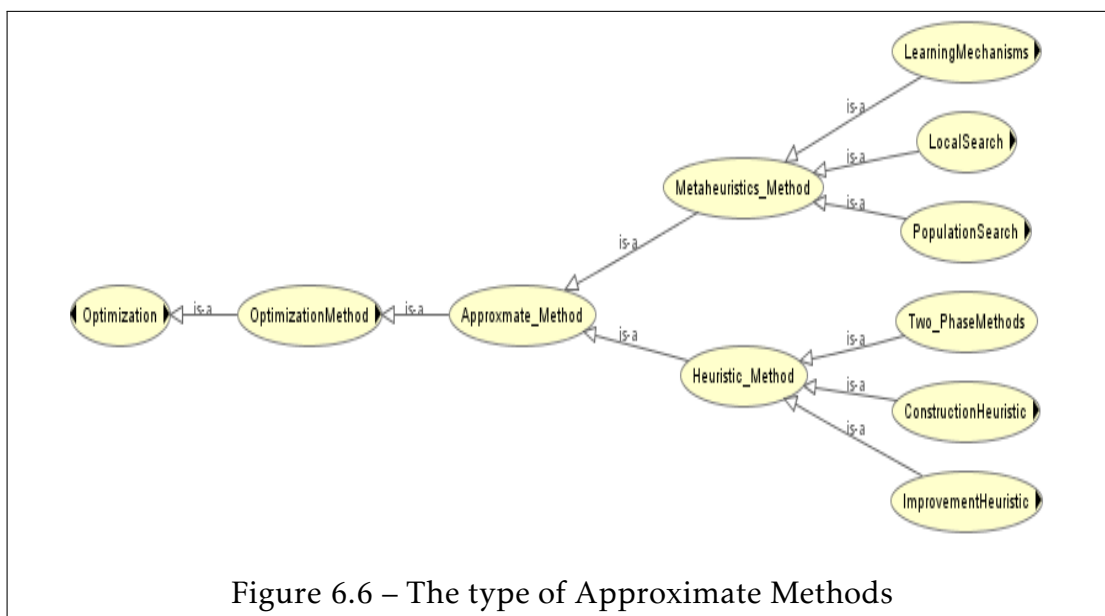
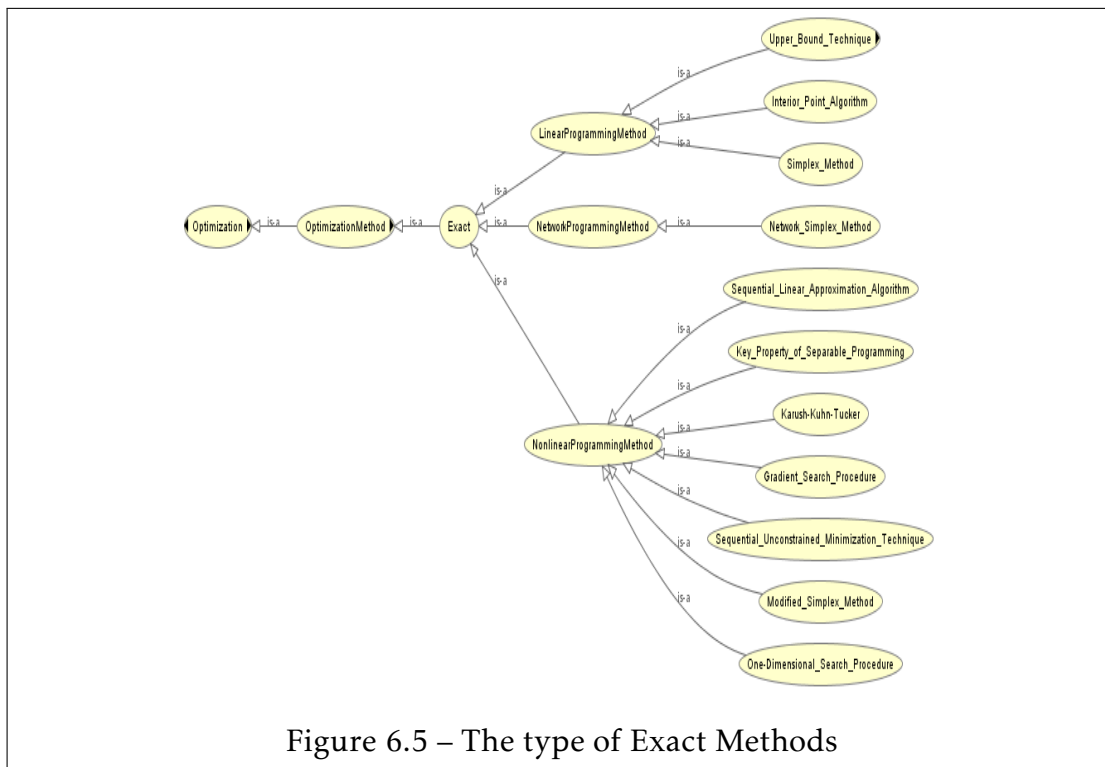
Each one of the classes above, may contain further sub-classes such as :

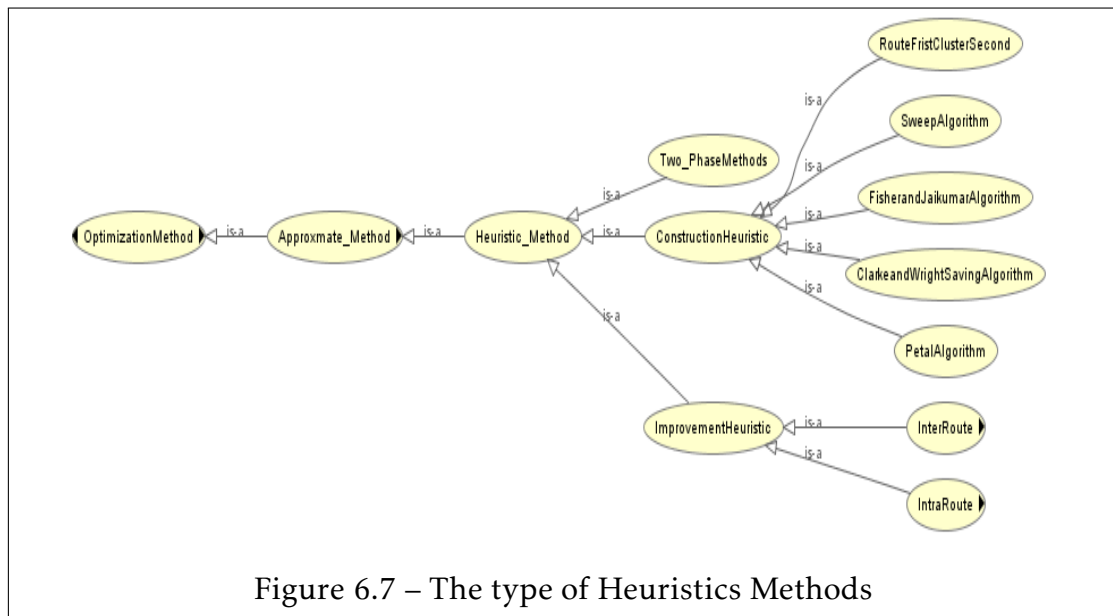
- The *Linear programming methods* include *Upper Bound Technique*, *Interior Point Algorithm*, and *Simplex Method*.
- The *Network programming Methods* may contain *Network Simplex Methods*.
- The *Nonlinear programming Method* can include the *Sequential Linear Approximation Algorithm*, *Key Property of Separable programming*, *Karush kuhn tucker*, *Gradient Search Procedure*, *Sequential Unconstrained Minimization Technique*, *Modified Simplex Method*, and *One Dimensional Search Procedure*.

The *Approximate Methods* can include two different types *Heuristic* and *Meta-Heuristic* methods, such as shown in figure 6.6.

In figure 6.7, we show the *Heuristic Methods*. The different types of sub-classes included in *Heuristic Methods* can be classified, as follows :

- Construction Heuristic
- Improvement Heuristic methods
- Two Phase Methods





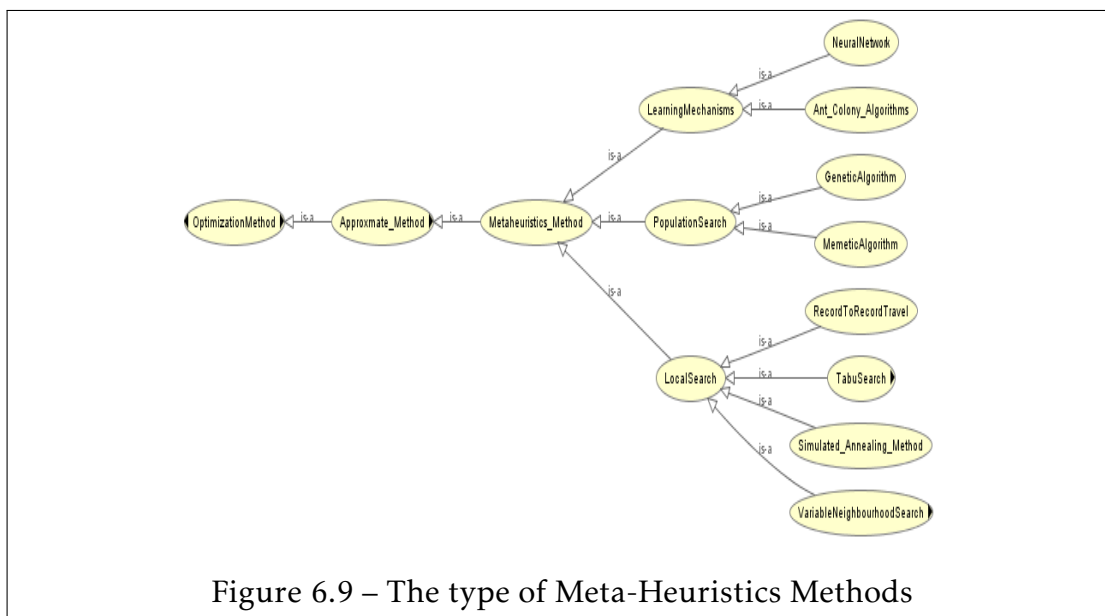
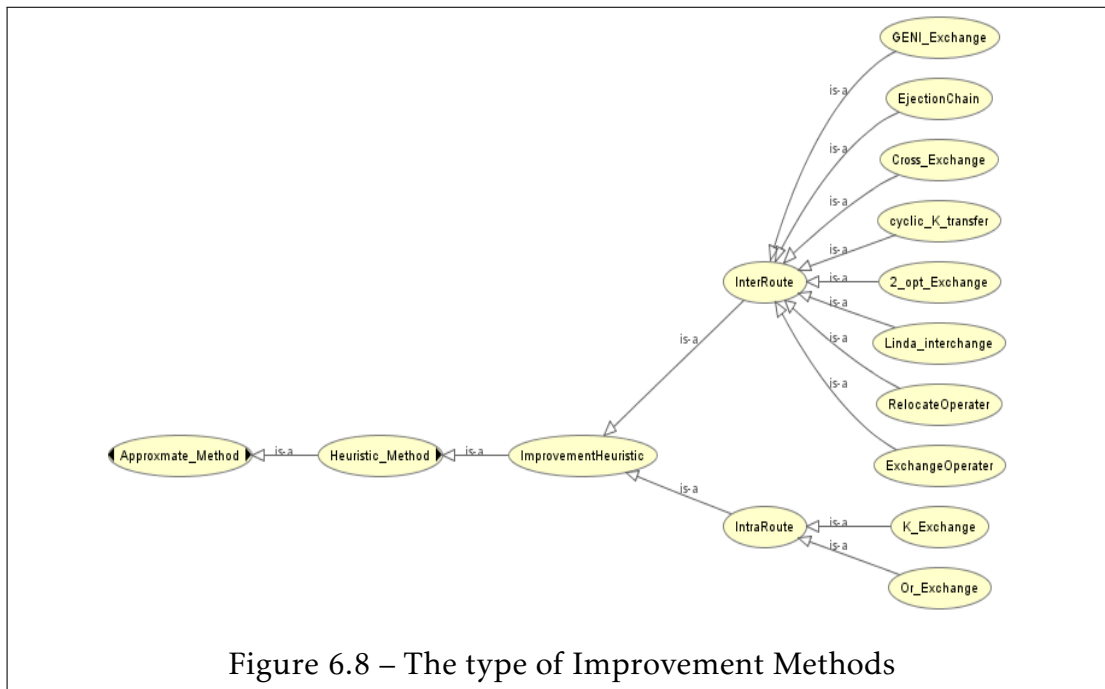
The *Improvement Heuristic* class can be classified into two sub-classes *Inter-Route* and *Intra-route*, such as shown in figure 6.8.

- The *Inter-route* can contain many type of methods such as *2opt Exchange*, *Cross Exchange*, *Relocate Operater*, *Cyclic-K-transfer*, *Ejection Chain*, *Exchange Operator*, *GENI exchange*, and *Relocate Operater*.
- Likewise,the *Intra-route* may include two methods, which are the *K-Exchange* and the *OR-Exchange*.

In figure 6.7 shows the *Construction Heuristic* class. It can include the methods such as the *Clarke and Wright Saving Algorithm*, *Fisher and Jaikumar Algorithm*, *Petal Algorithm*, *Route First Cluster Second*, *Sweep Algorithm*, and *Route First Cluster Second*.

Finally, Meta heuristics can be classified into three sub-classes, such as shown in figure 6.9.

- The *Learning Mechanisms* methods can be classified into the sub-classes such as *Ant Colony Algorithms* and *Neural network*.
- The *Local Search* methods can include the sub-classes such as *Record To Record Travel*, *Simulated Annealing Method*, *Variable Neighbourhoods Search* and *Tabu Search*. We continuously classify the *Tabu search*, and *Variable Neighbourhoods Search* into some other sub-classes.



- The *Population Search* methods can include two sub-classes Genetic algorithms and Mimetic algorithm.

6.6 The Reasoning system for Logistics Problems

We design and develop a reasoning system in respect of the logistics problem and the optimization methods. We define the different types of logistics problem in the designed reasoning system. It is capable to index the solution methods, defined as individual ontologies. For instance, we specify three types of logistic problems: The *Vehicle Routing Problems (VRP)*, *Passenger Train Transportation Problems (PTP)*, and *Container Terminal Problems (CTP)*.

The Vehicle Routing Problems (VRP) may further include various types, each one can be defined as individual ontology. We define the existing optimization methods as individuals in optimization ontology, to solve the respective problem. Some of them are as follows:

- The Capacity Vehicle Routing Problem (CVRP) is a type of Vehicle Routing Problem (VRP). There exist many methods that can solve such a problem. Such as *branch and cut method*, *2-opt-methods*, *Routing First cluster Second Methods*, *OR-opt-Method*, *3-Opt-Method*, *Clarke and wright Method*, *Cluster First Route Second Method*, and *Branch and Bound Method*, as shown figure 6.10.
- There exist many optimization methods to solve the *Vehicle Routing Problem with Time windows (VRPTW)*. Such as *Memetic Algorithm* [39], *meta-heuristics* [8], etc.
- Many existing methods can solve the *Vehicle Routing Problem backup (VRPB)* such as *exact methods*, *heuristic methods* [32], *meta-heuristics Memetic Algorithm* [58].
- The *Vehicle Routing Problem Pickup and Delivery (VRPPD)* problem can be solved by many methods such as *meta-heuristics tabu search* [59], and *heuristics* [7].
- The *Vehicle Routing Problem Multi-Depot (VRPMD)* problem can be solved by many methods such as *exact methods* [14], *heuristic methods* [11] and *meta heuristics Genetic Algorithms* [63], etc.

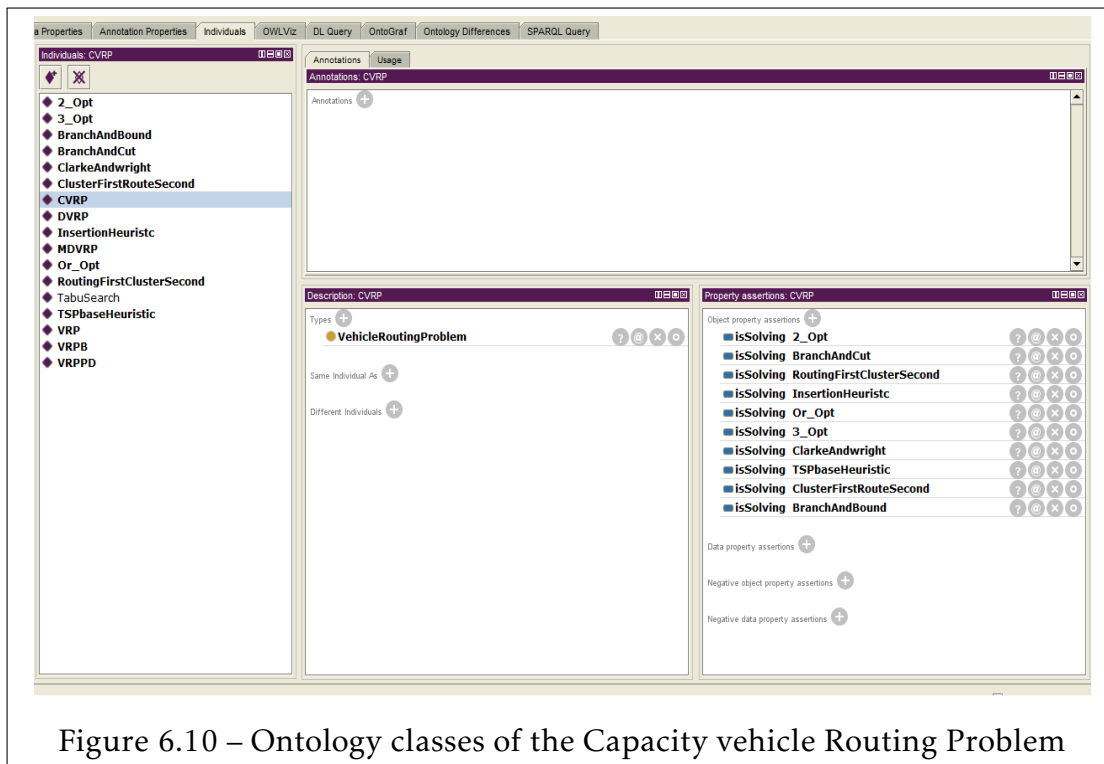


Figure 6.10 – Ontology classes of the Capacity vehicle Routing Problem

The Passenger Train Problems (PTP) includes many variant types depending on objective function.

There are several optimization methods that can be applied to find the solution of PTP. We discuss some variants of Passenger Trains Problems (PTP), as follows :

- A lot of research work, in the literature addresses the *Passenger Train Timetabling Problem (PTTP)*. In [6], Eva presents two optimization methods to solve the problem. She describes the *Integer Linear programming* and the *Large Neighborhood heuristic* and compares the results of these methods. In [5], Zahra proposes to use the Cellular Automate (CA) with genetic algorithm (GA) to solve the PTTP.
- Similarly, we find many solution methods for the *Rolling Stock Problem (RSP)* in the literature. They propose to use different optimization methods. For example, in [66], yasutaka proposes ant colony optimization (ACO). In [54], Reuther presents a heuristic method (generic hyper-graph) with mixed integer linear program (MILP). In [68], Jorgen compares two

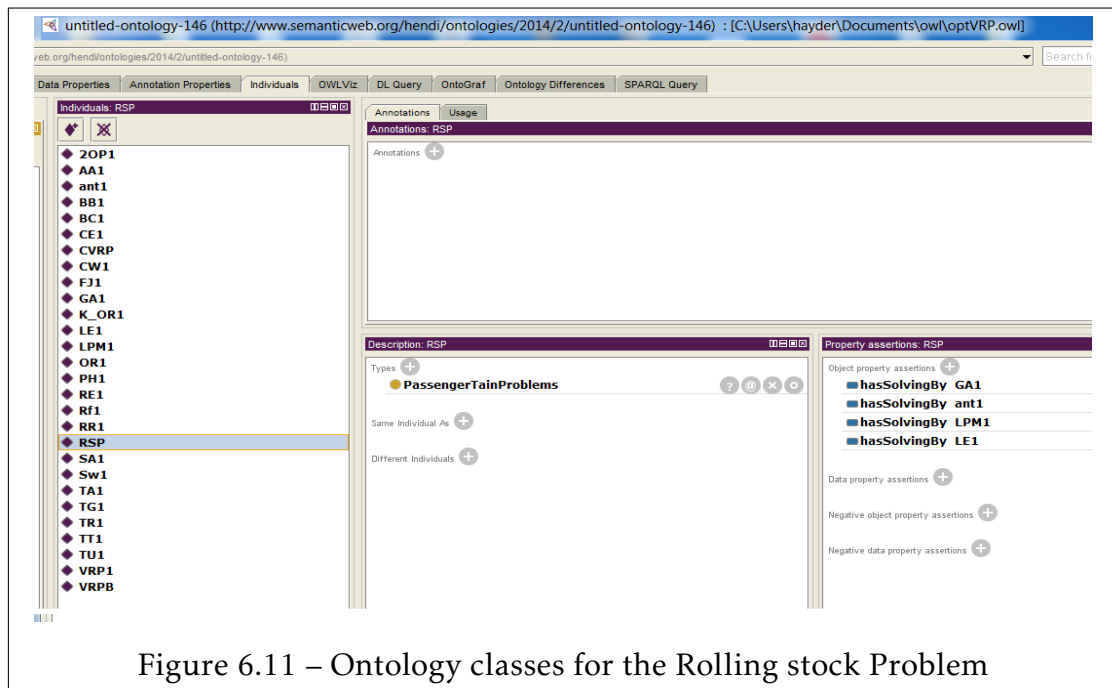


Figure 6.11 – Ontology classes for the Rolling stock Problem

approaches, the first one use CPLEX to solve Mixed Integer Linear Program (MILP), and the second algorithm uses the column generation.

- The *Crew Scheduling Problem (CSP)* is common to other problems but it has different constraints. Such as cargo railways system, in [12], Cavique proposes sub graph ejection and tabu search. In [1], Erwin proposes iterative partitioning method to solve the CSP.
- The figure 6.11 shows an example of the *Rolling stock Problem (RSP)* that is a type of the *Passenger Train Problems (PTP)*. It can be solved by the methods such as *Linear programming Methods, Genetics algorithm, Linda exchange Method, and Ant Colony algorithms*.

The *Container Terminals Problem (CTP)* can be solved by several optimization methods that can be applied to find a solution. Some of these methods, can be cited as follows :

- In [28], authors propose the *genetic algorithms* to solve the *Container Reshuffling Problem (CRP)*.
- In [19], authors propose the *heuristics method* to solve *Container Relocation(reshuffling) Problem* for export terminal.

- In [38], authors propose the *stochastic dynamic programming* to find min reshuffling with departure time windows for import terminal.
- In [21], authors proposes stacking policies to consider the departure time as parameters to stocking the container in yard.
- For *Berth and Quay Crane Allocation Problem (BQCAP)*, there exist the solution methods, such as:
 - In [41], authors propose the *branch-and-price algorithm* to solve it.
 - In [15], authors propose the *dynamic deployment scenario*.
- The *Appointment Time for External Trucks Problem (ATETP)*, there exist the solution methods, such as:
 - In [43], authors propose to model the integrated vehicle scheduling and container storage in Container Terminals.
 - In [13], authors propose Stacking policy to solve the container stacking problem for departing container from terminal.

6.7 Conclusion

In this chapter, we present the optimization ontology to integrate the optimization methods to solve a particular logistic problem. It is made possible by defining the optimization problems, the optimization methods, and the optimization in components, to interact with the already defined logistics ontologies. We take an in-depth review of the optimization methods for the respective logistic problems. We preview their integration for the development of a reasoning system.

We present the abstract optimization knowledge models. It facilitates the development of inherent optimization models, which may solve the optimization problems. The optimization ontology contains the general concepts of optimization. It also define some individuals of logistic optimization problem. The ontology, we develop can integrate some other optimization ontologies. It is capable to contain the concepts to make it more general in order to cope with the majority of the existing optimization problems and the methods. They are also capable to classify these concepts, depending on their nature, such that the Continuous versus Discrete Optimization, Unconstrained versus Constrained,

or Deterministic versus Stochastic, etc.

We briefly present the reasoning system, initially based on the three types of logistics problem such as Vehicle Routing Problems (VRP), Passenger Train Problems (PTP), and Container Terminal Problems (CTP). In the next chapter, we discuss in detail the design and development of the reasoning system to validate the proposed strategy.

Part III

The Implementation

Implementation prototype of validation

7.1 Introduction

The logistics and optimization ontologies can be managed interactively by a reasoning system. The proposed system integrates the ontologies as the repositories. It can be exploited by a user along with other logistics data with the help of usable interfaces. It allows to launch user-defined queries to index a particular logistic problem, and its respective optimization methods. The reasoning system also hosts the web-services data, which implements the optimization methods.

The implementation of Ontology-Based Reasoning System inherently use several languages. It uses OWL to built the different ontologies described in the previous chapters. The SPARQL is used to construct the queries intended to extract the concepts needed to identify the logistics problems. We also use the languages dedicated to define Web Services, such as WSDL and those used for data exchange like JSON, etc. The reasoning system has been developed and deployed in form a dynamic web Java EE application.

In this chapter, explore the execution sequences of designed reasoning system. We also present the algorithms to find the exact types of logistic problems and later on to find the optimal web-services. Finally, we discover the functionality of the reasoning system with the help screen captures.

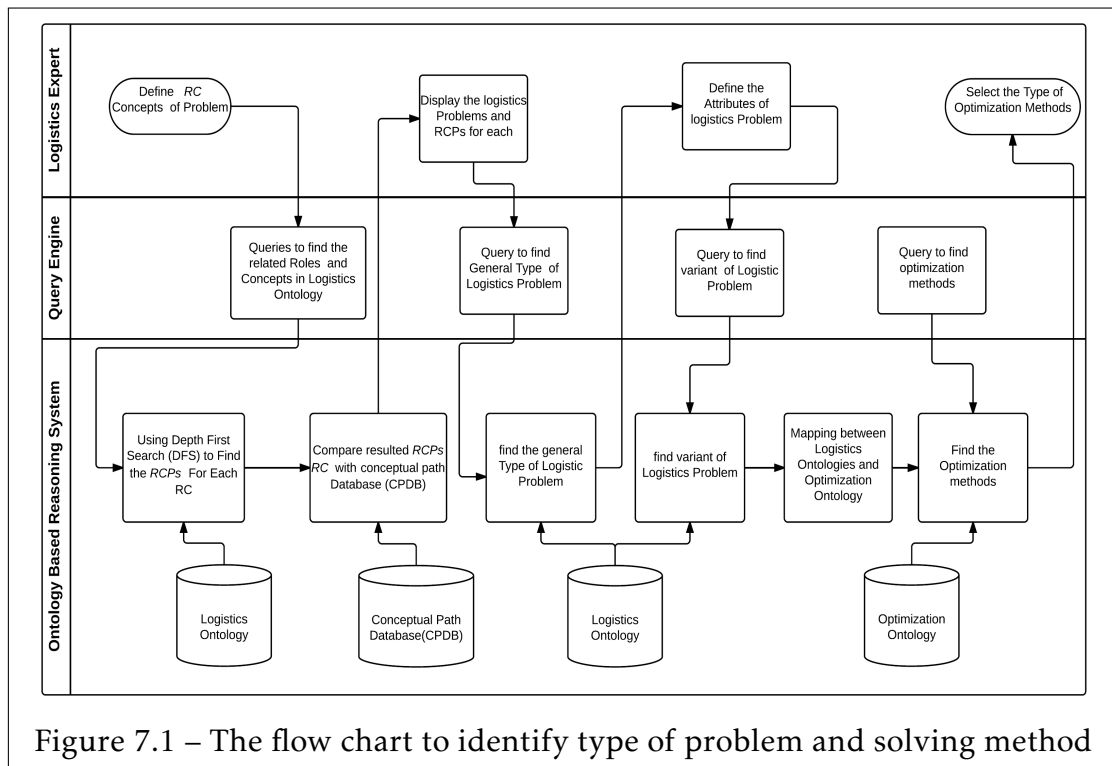


Figure 7.1 – The flow chart to identify type of problem and solving method

7.2 The flow chart of Reasoning System

This section discusses the set of processes composing the Ontology-Based Reasoning System. These can be categorized into two parts :

- The processes to find the type of logistics problem and index the solution methods
- The processes to find web services which execute the underlying implementation of the solution methods and presents the result of such an execution.

In the following, we describe the details of these processes.

7.2.1 Finding the type of logistic problem and its solution method

The figure 7.1 shows the general processes to identify logistics problem and to specify the optimization methods. These processes are organized on three layer : the logistics expert, the query engine, and ontology based reasoning system.

When the logistics expert interacts with process builder, (s)he identifies the concepts defining her problem. Therefore, (s)he identifies the type of her logistic problem. The interaction between the system and the logistics expert includes two steps. Finding first the type of logistics problem and then the corresponding optimization methods that solve the found problem.

Find the type of Logistics Problems

The identification as what we call finding the logistics problem is mainly based on the identification of the Resource Concepts (RCs) that are used by the problem. So, the expert begins the interaction with reasoning System by choosing the Resource Concepts from a category of RCs defined in the logistics ontology.

We assume the fact that RCs are at the core of every logistic problem. Inversely, the logistic problems depend on the involved Resources concepts (RCs).

For instance the *Vehicle Routing Problem* depends on the *Vehicle* and *Warehouse* resources. Likewise the *Passenger Train Problem* depends on the *Train* and *Passenger* resources. Similarly, *Container Terminal Problem* depends on the *Ship* and *Warehouse* resources.

Once, the resource concepts are specified, the Ontology-Based Reasoning System extracts the *Resource Concepts Path (RCP)* by means of queries addressed to the logistics ontology. The RCPs represent the interconnections of the resources by means of the role or relationship. The Ontology-Based Reasoning System uses the *Depth First Search (DFS)* (algorithm 1) to identify the Resource Concept Path(RCP). We can also use Breadth-First-Search (BFS) to identify the corresponding RCPs, but we believe the DFS is more convenient (in terms of memory and speed) than BFS because of the wider structure of inter-linked RCs.

Let us further illustrate this problem with the help of resource concepts of the Vehicle Routing problem (VRP). There are two major RCs (*Vehicle* and *Warehouse*). We observe that there exist the tree of interconnected concepts from RCs concepts to Logistics problem such as shown in figure 7.2. The DFS uses Resources Concepts (RCs) to find the RCPs which leads to the identification of two independent RCPs for each RC. Consequently, the system identifies a RCP

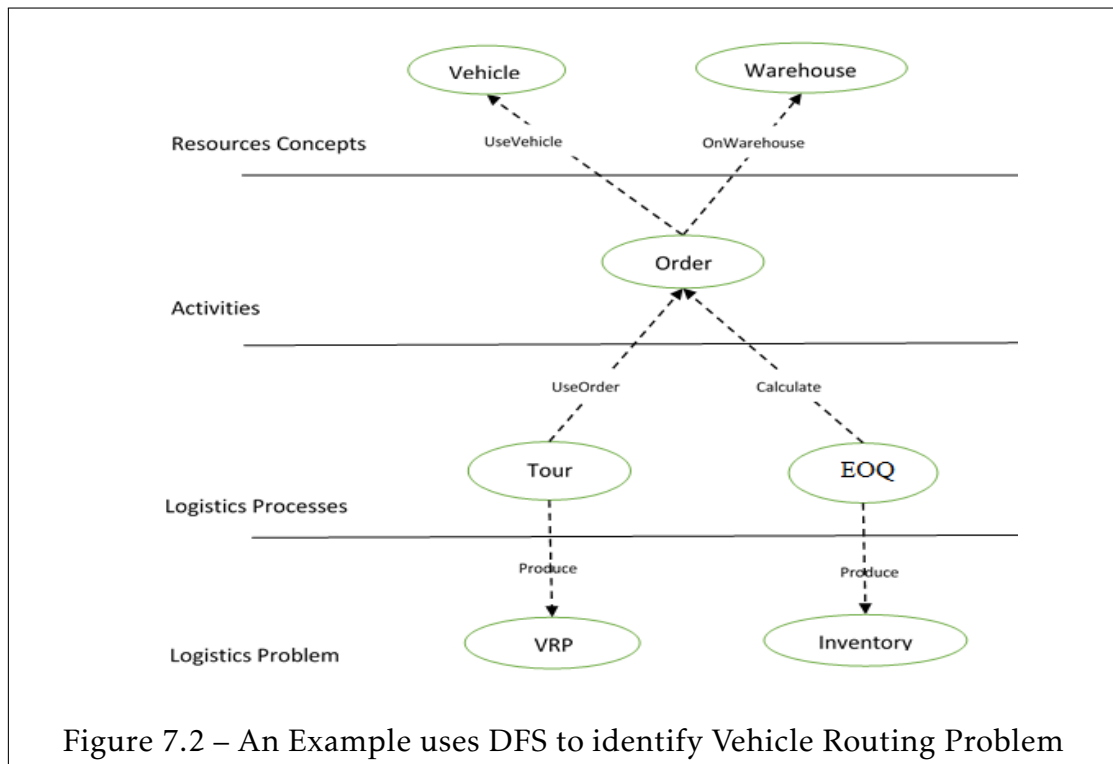


Figure 7.2 – An Example uses DFS to identify Vehicle Routing Problem

for *Vehicle* Resource and another RCPs for *Warehouse* Resource. Let us explain below the extractions of the DFS algorithm based on the the VRP problem.

1. The logistics expert can select the Resources Concepts of Problem (*Vehicle* and *Warehouse*).
2. The RCPs can be determined by following the inter-linking Roles between RC and the type of logistics problem. For instance, we focus to identify the Vehicle Routing Problem (VRP), along with the resulted RCPs as returned by DFS, for two RCs (which are *Vehicle* and *Warehouse*), as shown in Figure 7.3.

The DFS considers array lists such as *Related Concepts List (RCL)* including the related concepts, and *Visit List (VL)* including the visit concepts. It also takes into account a stack which is a *Path Stack (PS)*. As, the DFS identifies all the RCPs for each RC, so in this case, it first identifies the RCPs for *Vehicle* resource and then for *Warehouse* resource. We can explain the sequence processes of DFS, as follows :

Algorithm 1 Depth First Search

```

1: function DEPTH –FIRST –SEARCH(RC)
2:   Let array Path Stack(PS)
3:   Let array Visit List(VL)
4:   Let array Related Concepts List(RCL)
5:   Let array Resource Concepts path List(RCP)
6:    $i \leftarrow 0$ 
7:    $PS \leftarrow RC$ 
8:    $VL \leftarrow RC$ 
9:    $C \leftarrow RC$ 
10:   $stop \leftarrow false$ 
11:  while logisticsproblem  $\neq true$  do
12:     $RCL \leftarrow ResultsRangeConcepts$   $\triangleright$  Make query to find all roles and
    concepts that are Range of C
13:     $RCL \leftarrow resultsDomainConcepts$   $\triangleright$  Make query to find all roles and
    concepts that are Domain of C
14:    for each  $m \in RCL$  do
15:      for each  $s \in VL$  do
16:        if  $m = s$  then delete( $m$ )
17:        end if
18:      end for
19:      if  $m$  is ResourcesConcepts then delete( $m$ )
20:      end if
21:    end for
22:    for each  $m \in RCL$  do
23:      if  $m$  is LogisticsProblem then
24:         $RCP[i] \leftarrow PS$ 
25:         $i \leftarrow i + 1$ 
26:        delete( $m$ )
27:         $VL \leftarrow m$ 
28:      end if
29:    end for
30:    if RCL is not empty then
31:       $C \leftarrow dequeue(RCL)$ 
32:       $PS \leftarrow C$ 
33:       $VL \leftarrow C$ 
34:    end if
35:    if RCL is empty then
36:      if PS is not empty then
37:         $C \leftarrow pop(PS)$ 
38:      else
39:         $stop \leftarrow true$ 
40:      end if
41:    end if
42:  end while
43:  return RCP
44: end function

```

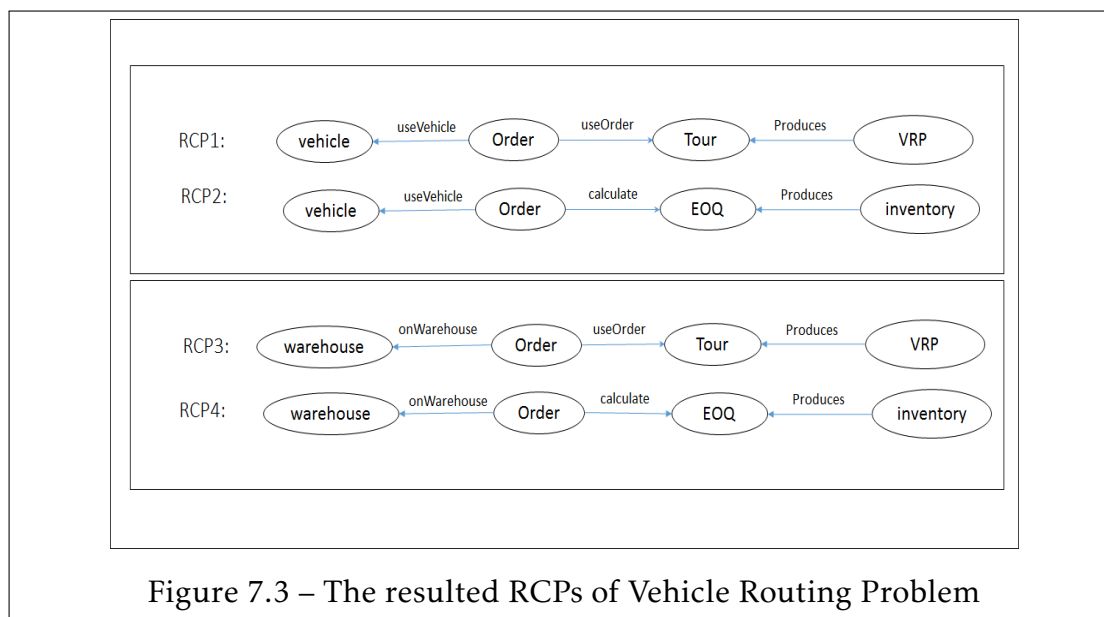
- The DFS add the *Vehicle* concept into *PS* and *VL* list. It can help to find the roles connected to *Vehicle*. We also find the related concepts. For instance, we find the roles and range concept. Such as the domain concept of *useVehicle* is the *Order* and the range concept is the *vehicle*. Hence, we add the related concepts *Order* to *RCL*.
- Compare the each concept found in *RCL* with the visit concepts list (*VL*) and Resource Concepts (*RCs*). If there exist a common item in *RCL* that is both in *VL* and *RCs* then it should be deleted. We compare the resulted items in *RCL* to find the Logistics Problem. For this purpose, we move the concepts in *RS* as *RCP* to *RCPs* list from *RCL*. Verify that the *RCL* is not empty, then choose the first element of *RCL* and add it in *PS* and *VL*. We continue to find related concepts for the whole process, as stated above. For instance, we consider the *RCL* that includes the *Order* concept. This concept is not in *VL* and *RCs*, and it is not a logistics problem. We add *Order* to *VL* and *RS* and continue for the three related concepts (*Warehouse*, *Tour*, and *EOQ*). We then verify, if the *Warehouse* concept is resource concept then we delete it. We can select the *Tour* and add it to *VL* and *RS*. We then find the related concepts of *Tour* that is *VRP*. The *VRP* is a logistics problem. Proceeding, in the same way, we delete the *VRP* from *RCL* and return the *RCP1* that includes *PS* and *VRP*.

RCP1: Vehicle → *Order* → *Tour* → and eventually *VRP*.

- Another condition is to verify if the *RCL* is empty. Then, we can pop an item from the Result Stack (*RS*). We attempt to find the related concepts until the result stack is empty. For instance, consider if there is no element in *RCL*, then we can pop(*Order*). We can continue to find the related concepts (*Warehouse*, *Tour*, and *EOQ*) and compare them with *VL* and *RC*. We can then select the *EOQ* and find the related concepts (*Inventory*) and compare if it is a logistic problem. Then we can return *RCP2*.

RCP2: Vehicle → *Order* → *EOQ* → and eventually *Inventory* : the stop condition is, if the *RS* and *RCL* are empty.

In the same way, we also find the *RCPs* of the *Warehouse* Resource, as



shown in figure 7.3.

3. To identify the general type of logistics problem; We devise the RCPs into Similar Concept Path Groups (SCPG). The groups are devised on the basis of most common premier concepts of the RCPs. To further illustrate, let us consider, once again, the example shown in Figure 7.3. We have RCP1 and RCP2 for *Vehicle* and RCP3 and RCP4 for *Warehouse*. We can identify two SCPGs; SCPG1 contains RCP1 and RCP3, whereas the SCPG2 contains RCP2 and RCP4. As a result, we can find the problem type *VRP* from SCPG1 and problem type *Inventory* from SCPG2.
4. Compare the resulted SCPGs with Concept Path Databases (CPDB), the CPDB can find the general type of logistics problem as *VRP*. It depends on the SCPGs, as found above. The CPDB includes the RCPs of each general type that are identified by ontologies, such as shown in Figure 7.4.
5. To identify specifically the exact type of logistic problem. We require logistics expert to specify some attributes of his logistics problem. This specification can lead us to find the exact type of problem with the help of logistic ontologies. For instance, the logistics expert can identify some data properties such as if *isCentral* is true, *Costumer Return Goods* is false,

Name of problem	Vehicle	Warehouse	Trains	Ship	Flight	people
Passenger Train Problems	Null	Null	RCP1	Null	Null	RCP2
Vehicle routing problems	RCP3	RCP4	Null	Null	Null	Null
Container Terminal Problem	RCP5	RCP6	Null	RCP7	Null	null

RCP3

- { Vehicle, Order, Tour, VRP }

RCP4

- { Warehouse, Order, Tour, VRP }

Figure 7.4 – An Example of CPDB

has some capacity is true and type of objective *Function* is min distance cost; that produces the variant of VRP as a CVRP. We use a SPARQL query to find variant of VRP, as shown in Algorithm 2.

The RCPs essentially begin with the problem type, because the found elements are stored in form of *stack* data-structure, as a result of DFS. This further justifies, our use of DFS (over BFS), as BFS stores the elements in form of *queue* data-structure. Hence, it is comparatively faster to compare the similar stacked elements from the top, between RCPs.

Find the list of Optimization Methods

The reasoning System maps the found logistics problem to the corresponding optimization ones by means of the optimization ontology. The aim is to assist the logistics expert to identify the optimization methods that are necessary to solve the logistics problem. For instance Capacity Vehicle Routing Problem (CVRP) can be mapped to some optimization methods that solve it, as shown in Algorithm 3.

7.2.2 Find Optimal Web Services

The figure 7.5 shows the global process leading to assist logistics experts to identify the optimal web services deployed in order to solve the optimization problems corresponding to initial logistics one.

Algorithm 2 Type of Logistics Problem

```

1: function TYPE-VRP(returnGood, CapcityVehicle, WarehouseCent, Objective-
   Function)
2:   connect to logistics ontology;
3:   launch query= SELECT ?problem
4:   WHERE { ?tour r:hasOrder ?order.
5:   ?order :useVehicle ?vehicle ;
6:   :onWarehouse ?warehouse;
7:   :CostumerReturnGoods "returnGood"
8:   ?warehouse :isCentral "WarehouseCent" .
9:   ?vehicle :hassomecapcity "CapcityVehicle".
10:  ?tour r:produces ?problem;
11:  :hasFunction : "ObjectiveFunction"};
12:  typeproblem ← execute(query);
13:  return type problem
14: end function

```

Algorithm 3 Find List Optimization Methods

```

1: function TypeofMethods(TypeProblem)
2:   connect to ontimization ontology;
3:   replace ← Mapping(Typeproblem)
4:   launch query= SELECT ?method
5:   WHERE { : "replace" :isSolving ?method.
6:   ?method rdf:type ?class.
7:   ?class rdfs:subClassOf ?class1.}
8:   methods[] ← execute(query);
9:   return methods[]
10: end function

```

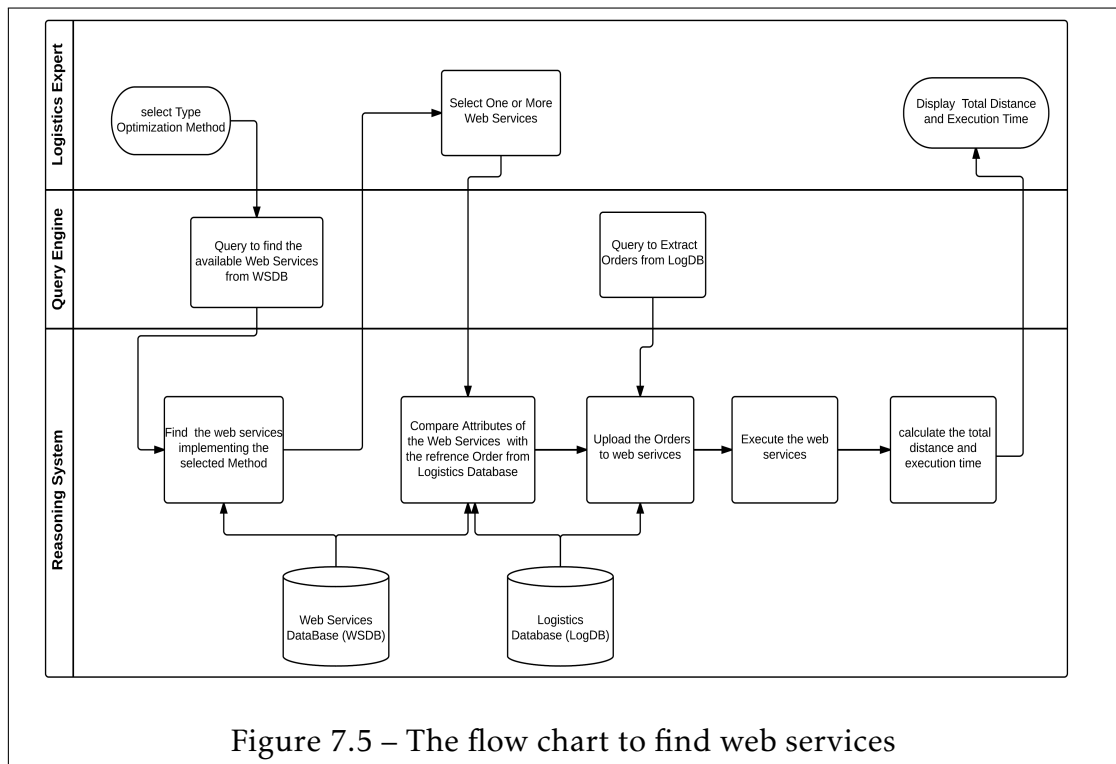


Figure 7.5 – The flow chart to find web services

The logistic expert selects an optimization method from a list of optimization's methods found by the reasoning system. The Reasoning System also recognizes available web-services from Web Services Database (WSDB). This recognition depends on some criteria like the type of logistics problem and implementation of the selected optimization method, etc. The system builds a kind of parameterized SQL query addressed to WSDB (which is a relational database).

Likewise, the web-services are also characterized by different attributes concerning the performance, the size, and the algorithm efficiency.

These may also involve the problem complexity, the number of nodes, time constraints. The choice of these attributes and to qualify their importance is strictly restrained by the logistics expert.

Therefore, the Reasoning System must compare the attributes of the web services with logistics data needed to solve the problem. For example for the CVRP problem there are many web services with limitation conditions such as the number of order (nodes) that can be executed, etc.

Web Service	Total Distance (Km)	Time (s)
optimoRoute	26.145	22670
TrackRoad	17.727	2170

Table 7.1 – The results of execution web services

Finally, the system may upload logistics data to the web services for execution and then gets the results. In fact, the data extraction is achieved by means of the queries, addressed to the logistics database. We used two techniques to transfer the logistics data to web-services that are explained in section 4.2.6.

We experiment the same data (involving 20 orders) on different web-services (for instance: TrackRoad¹, OptimoRoute²). Each web-service uses different optimization method. Evidently, we obtain different results according to the capability of the web-service to solve the logistics problem. The developed plate-form provides a means to store the user experience for verification. The results of this experiment for TrackRoad and OptimoRoute (as both offers to solve the same set of problem types) as shown in Table 7.1.

7.3 The prototype of reasoning system

The reasoning system, we develop is composed of two main parts. The first part deals with assisting the logistics experts to well identify their problem and then solve the problem. The second part consists of some functions leading to change or modify the background of knowledge of the system by adding problem descriptions, or uploading optimization methods or web-services.

The figure 7.6 shows initial window interface of the ontology based reasoning system. The two main icons concern the two part of the system discussed above.

When the Logistics Expert check the START icon, the system identify a list of the Resources Concepts (RC) available in the Logistic ontology by means of a SPARQL query such as shown in figure 7.7. The logistics expert may select some resources from this list and then the system identifies the corresponding

1. TrackRoad:<http://doc.trackroad.com>

2. OptimoRoute:<http://www.optimoroute.com>

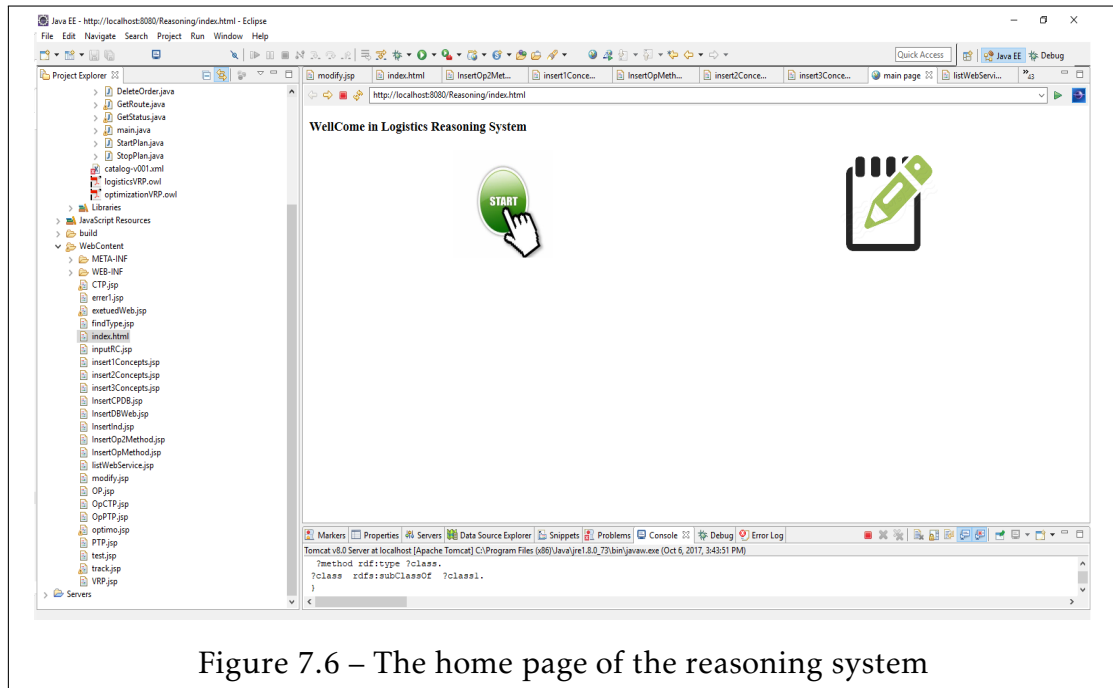


Figure 7.6 – The home page of the reasoning system

Resources Concepts Paths (RCPs).

For instance, the logistic expert may select couple of resources (*Vehicle* and *Warehouse*). The system identifies the Resource Concept Paths (RCP) for each Resource Concept (RC) by means of the DFS algorithm. The system compares the resulted conceptual paths into Similar Concept Path Groups (SCPG), and compare their list with Conceptual Paths Database (CPDB) to find general type of logistics problem.

Hence, the system finds and return the general logistics problem. It may return a list of general logistic problems, and the logistic expert must choose one of them.

The system uses the attributes of general logistics problem concepts in order to assist the logistics expert to specify more precisely the variant of the logistics problem.

The figure 7.8 shows an example of attributes concerning the *Vehicle Routing Problem (VRP)*. This problem may have several variants depending of the attributes like : the vehicle has some capacity (true or false), the goods are in a central warehouse, etc.



Figure 7.7 – The interface that are shown the Resources Concepts

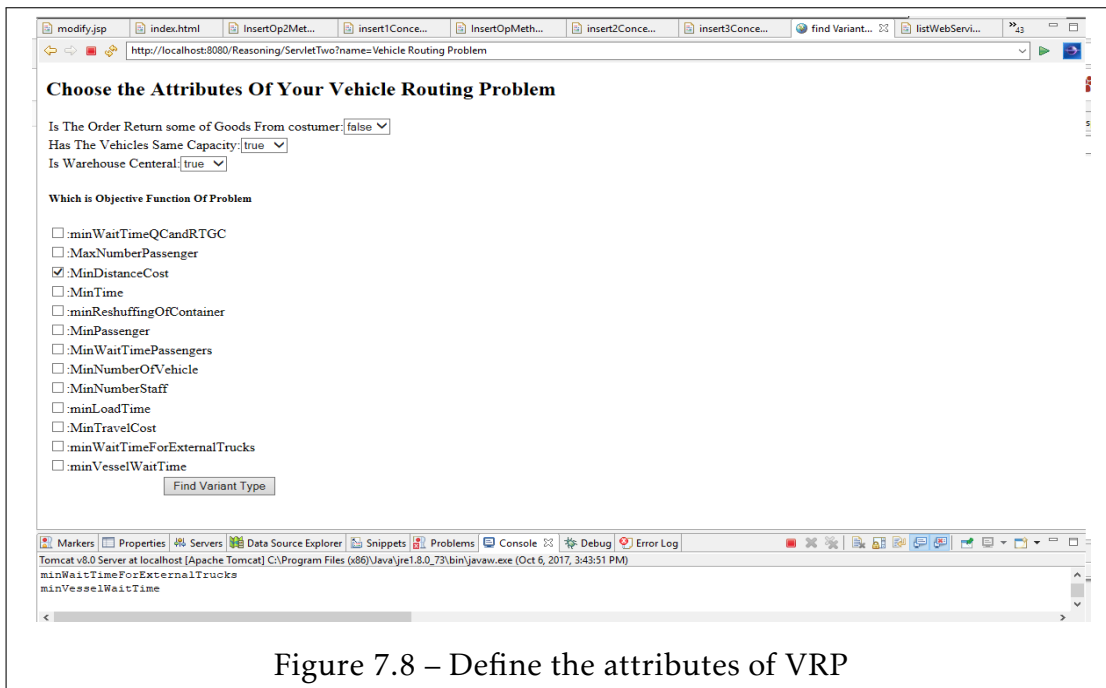


Figure 7.8 – Define the attributes of VRP

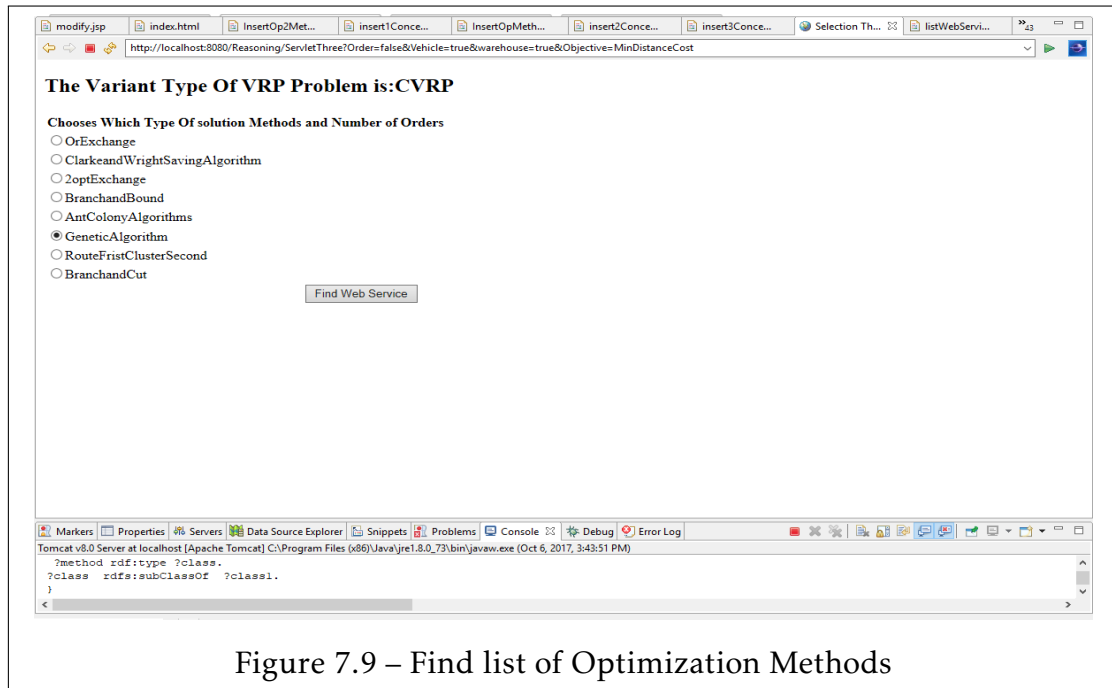


Figure 7.9 – Find list of Optimization Methods

For instance, if the logistics expert specify the value "true" for attribute "vehicle have some capacity ", the "false" value for attribute "costumer can return goods", and the value "true" for attribute "Warehouse is Central" and choose objective function "minimum distances", then the system identify a variant of the VRP as the *Capacity Vehicle Routing Problem (CVRP)*.

Likewise, the system make the mappings between individuals of CVRP or instance of the CVRP concepts, and concepts specified in the optimization ontology. The system also build queries in order to extract the optimization methods to solve the CVRP. The logistics expert is then invited to choose one of these methods, as shown in figure 7.9.

When logistic expert chooses an optimization method like the Genetics Algorithm (GA). The reasoning system might make a query to Web Services database (WSBD) to find web services which can solve the CVRP by means of Genetics Algorithm (GA). Such web services can be TrackRoud and OptimoRoute, for instance (Figure 7.10). When the logistic expert select one or more web services such as TrackRoud, the system transfers the orders of correct problem from logistics database to web service for execution.

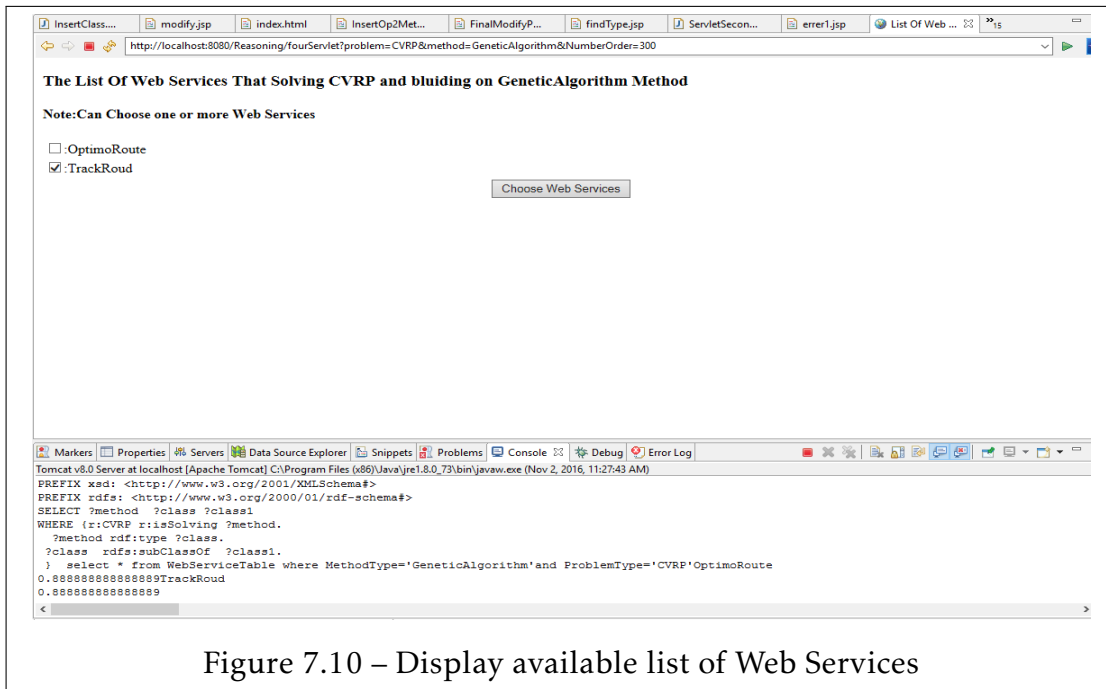


Figure 7.10 – Display available list of Web Services

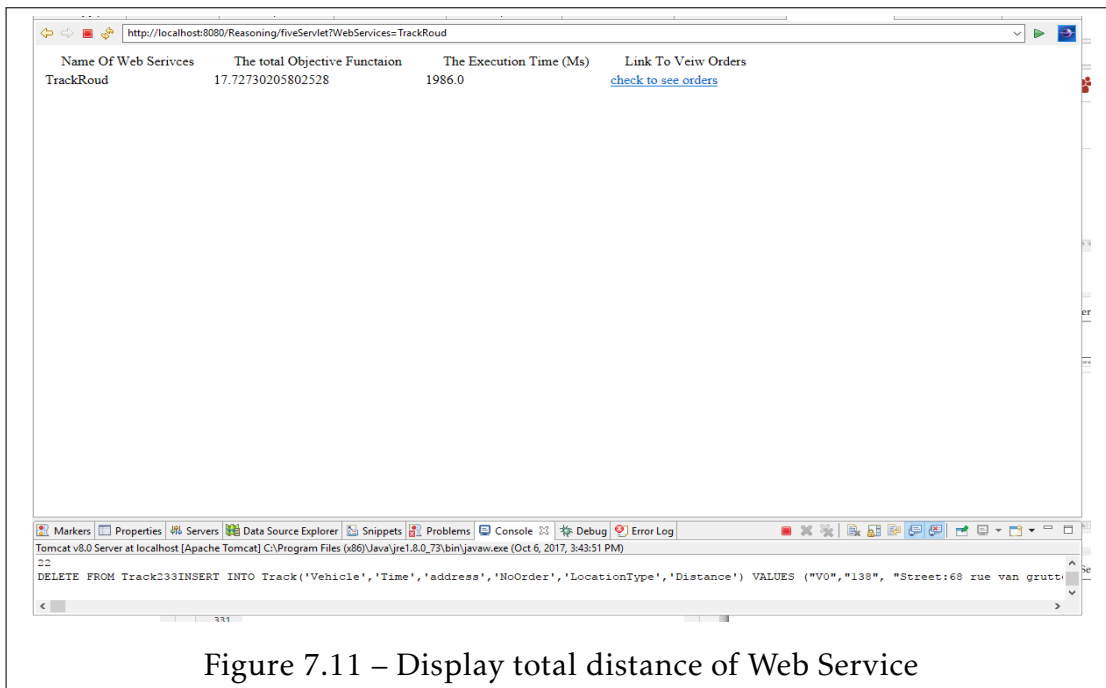


Figure 7.11 – Display total distance of Web Service

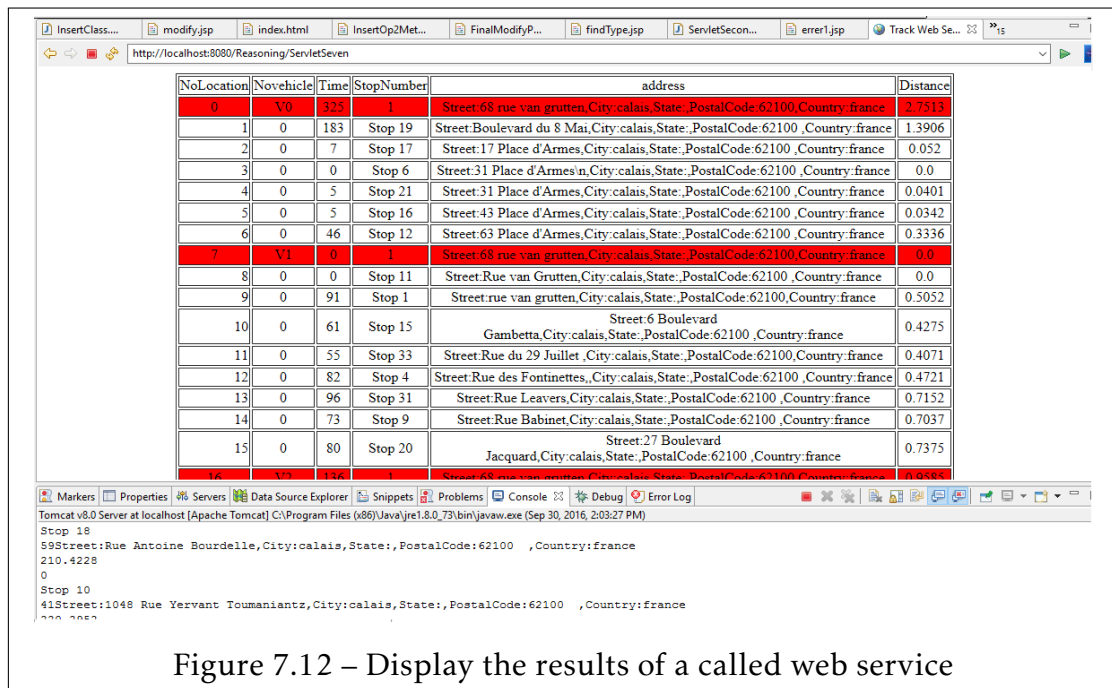


Figure 7.12 – Display the results of a called web service

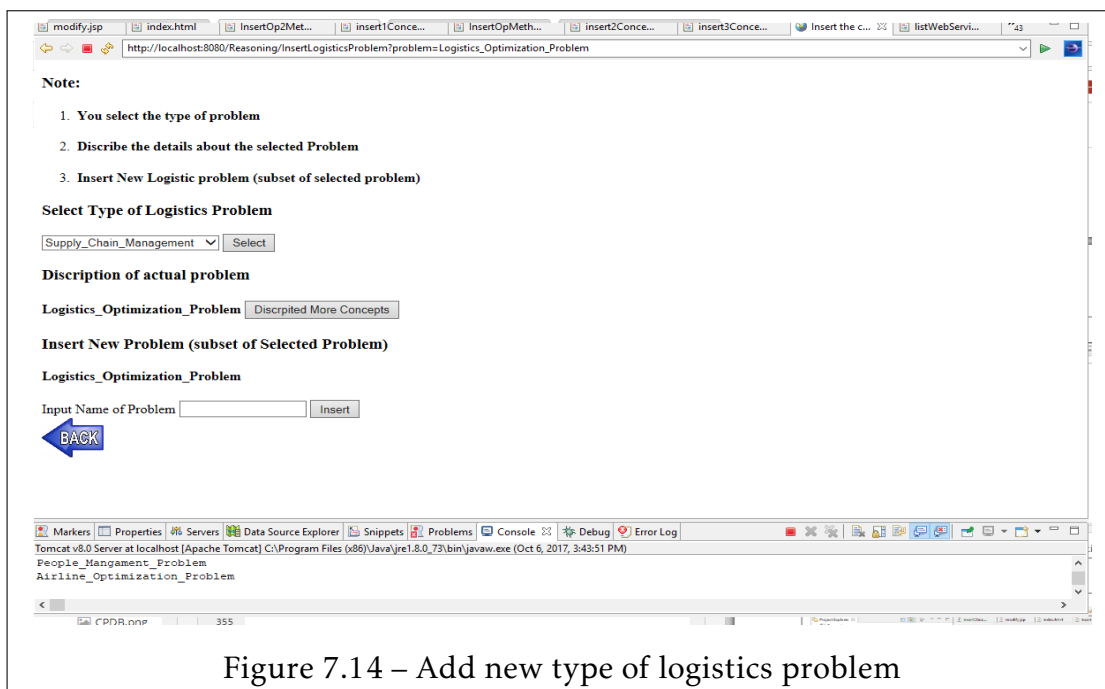
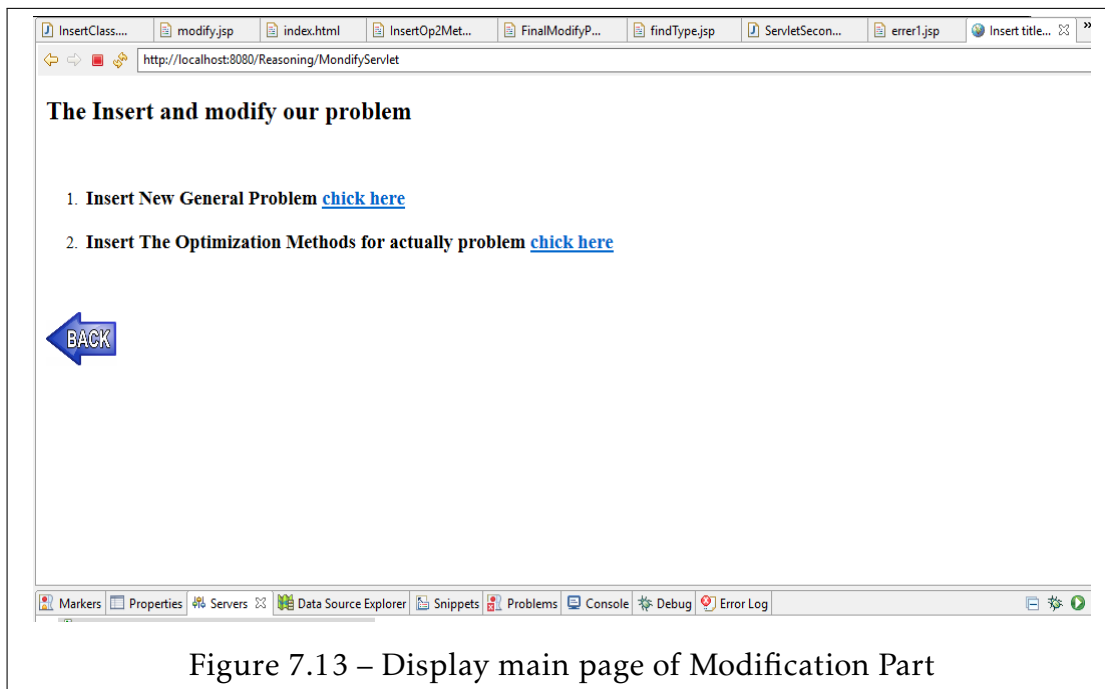
The figure 7.11 shows the name of the chosen web service, total distances, and execution time. It also show the total steps of vehicles such as shown in Figure 7.12. We obtain a list of stops of each vehicle along with the order details and web-service performance attributes. The logistic expert can choose and call the web-service. In return, (s)he may obtain the results.

The insert and modification part

In this part, we present the means through which a logistic expert can insert or modify the concepts of actual logistics problem in logistics ontologies, or add/modify the optimization methods in optimization ontology, or insert the new web services in WSBD.

The Figure 7.13 shows the interface, including the two main parts to insert the concepts for new general logistics problem, and/or to add new optimization method or modification methods in reasoning system.

When we choose first one to insert general logistics problem. There are three buttons such as shown in figure 7.14. The first must be selected which is the type of available general logistic problems in logistics ontologies. It can provide



The screenshot shows a web browser window with the URL `http://localhost:8080/Reasoning/ServletNight?problem=VehicleRoutingProblem`. The page content is organized into five main sections:

- Problem Name Concept:** A text input field containing "VehicleRoutingProblem".
- Process Concept:** A dropdown menu currently showing "Tour".
- Active Concept:** A dropdown menu currently showing "Order".
- Resource Concept:** A list of checkboxes for "Train", "Ship", "People", "PassengerPlane", "FieghtPlane", "Warehouse" (checked), and "Vehicle" (checked). Below this list is a "Goto Role" button.
- Objective Function:** A list of bullet points: "MinTime", "MinDistanceCost", "MinPassenger", and "MinNumberOfVehicle".

At the bottom of the form, there are three input fields for adding new concepts:

- "Input new Process" with an "Insert" button.
- "Input New Activity" with an "Insert" button.
- "Input New Objective Function" with an "Insert" button.

A blue "BACK" button with a left-pointing arrow is positioned on the left side of the form.

Figure 7.15 – Define the Concepts of logistics problem

the details of the concepts of logistics problem by choosing the second button. Otherwise, we have the option to insert new type of problems by using the third button.

The reasoning system provides the option to describe the concepts of logistic problem, such as shown in figure 7.15. It actually includes four concept fields, which are the process concept field, the activity concept field, the resource concepts, and the objective functions. The processes concepts can be used to select the actual logistics process or it may allow to add new process if it does not exist in logistics ontologies. The activity concept is same as the process concepts. The resource concepts, can help to select one or more resource concepts of the problem. Finally, the objective function can be selected if it already exist otherwise, it may allow to add the objective function.

The next step, allows to insert the roles of logistics problem such as shown in figure 7.16. It is same the insert concepts, we can select the existing roles or add the new ones if theses are never described. It does allow the possibility to add the general logistics problem. The reasoning system then automatically generate the Resource Concept Path (RCP) of the problem and insert it into Concept Path

http://localhost:8080/Reasoning/ServletEight?process=Tour&activity=Order&problem=VehicleRoutingProblem&RC=Warehouse&RC=Vehicle

Input the Role of Problem

VehicleRoutingProblem produces Insert Role

Tour hasOrder Insert Role

Order onWarehouse Insert Role

Warehouse Insert Role

useVehicle Insert Role

Vehicle Insert Role

Insert the Problem in Conceptual Path Database

Name of Insert CPDB

BACK

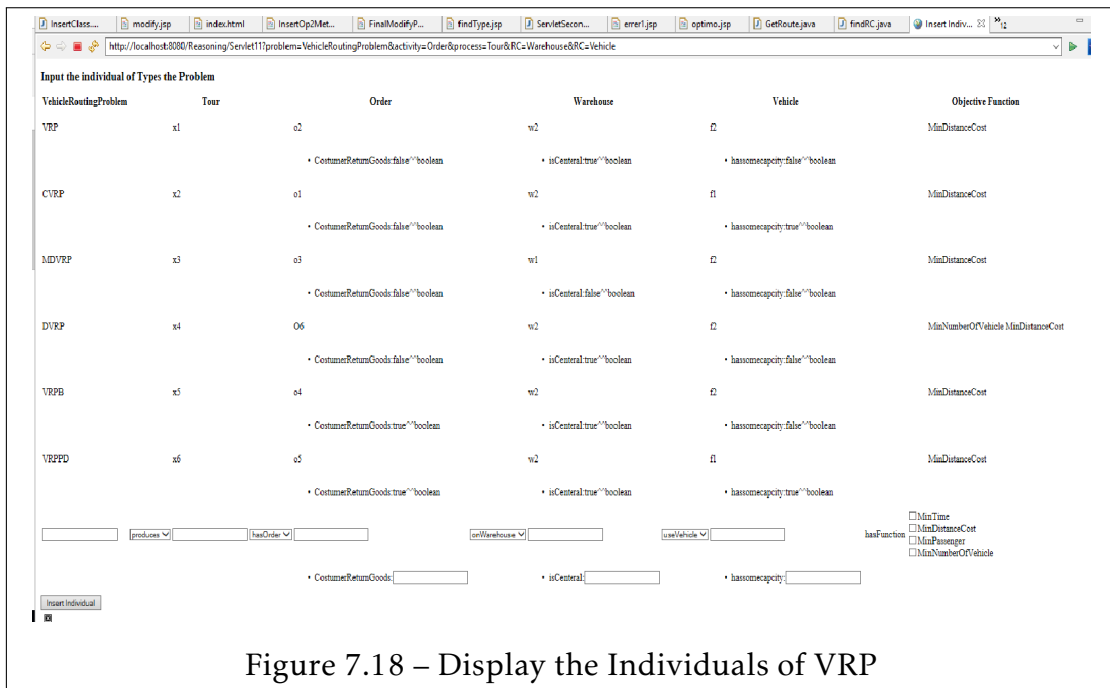
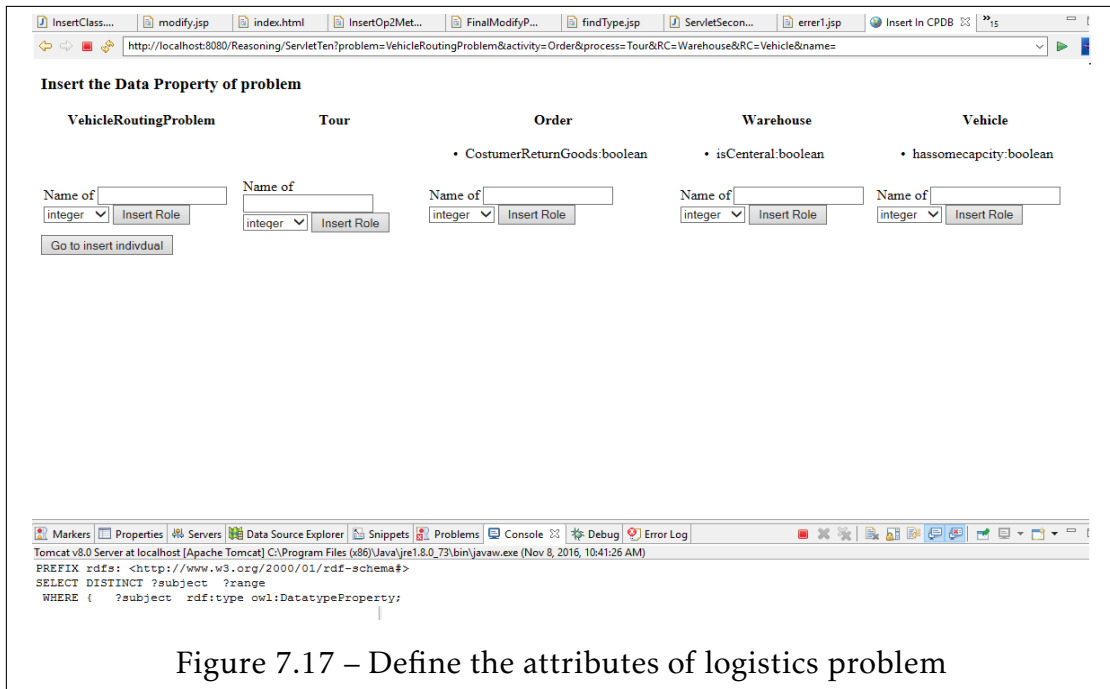
Figure 7.16 – Define the Objective Function

Database (CPDB).

Hence, we can define the attributes of the logistics problem such as shown in figure 7.17. For instance, we define the attributes of *Vehicle Routing Problem (VRP)*. We can also insert the data properties for each concept of the problem, these attributes help to define the type of logistics problem. Once the attributes are defined, we can then insert the individual variant type of logistics problem.

Hence, we can find all individual variants of *Vehicle Routing Problem (VRP)* to modify or add a new individual such as shown in figure 7.18. We can define new variant of logistics problem by inserting the concepts and attributes along with its objective function. These individuals are added in the logistic ontologies which in turn defines logistic problems.

In order to add the optimization methods into the optimization ontology repository, we select the option to add the optimization method, as shown in figure 7.13. Henceforth, we select the type of logistics problem to associate its solution method. The reasoning system is then capable to provide the individuals of logistics problem as defined earlier, such as shown in figure 7.19.



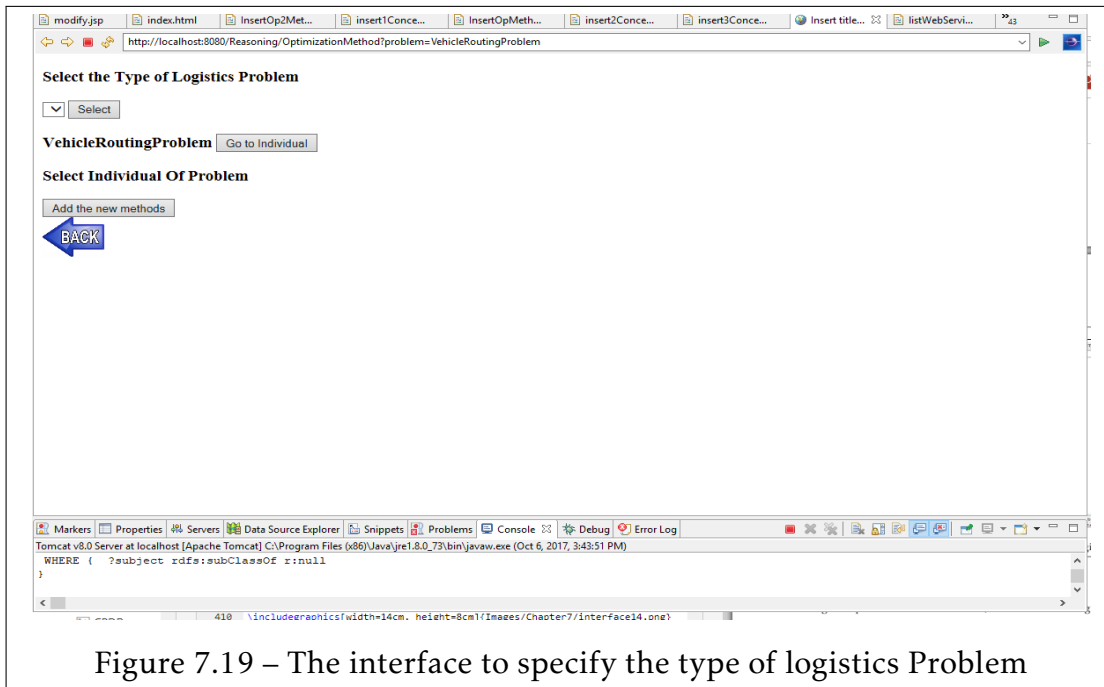


Figure 7.19 – The interface to specify the type of logistics Problem

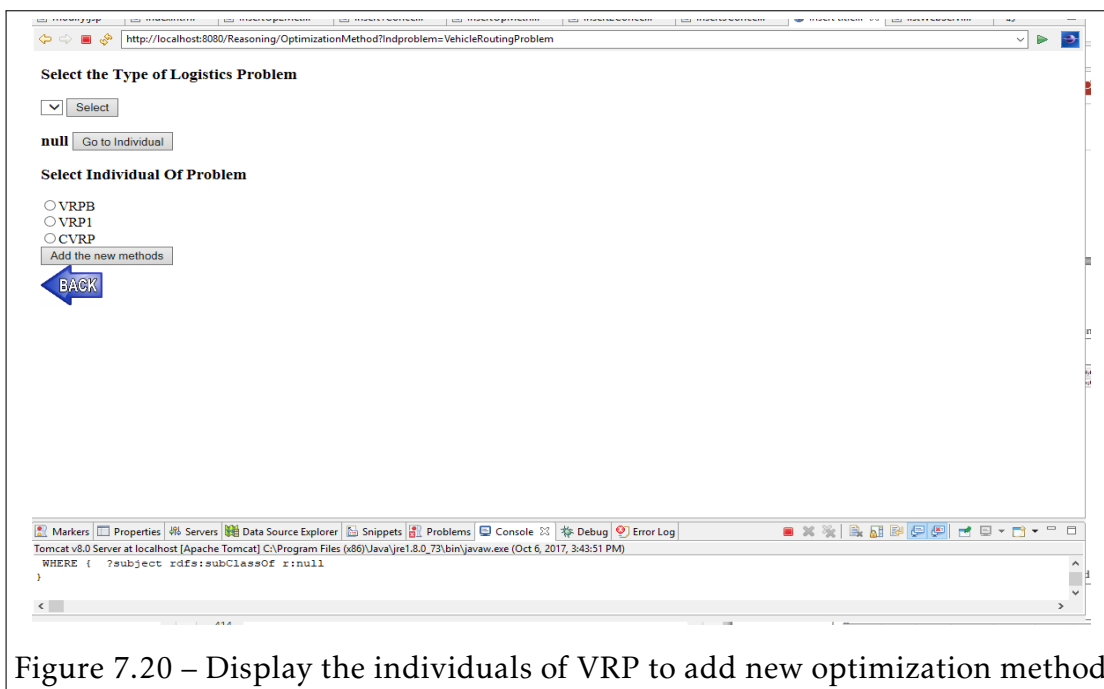


Figure 7.20 – Display the individuals of VRP to add new optimization method

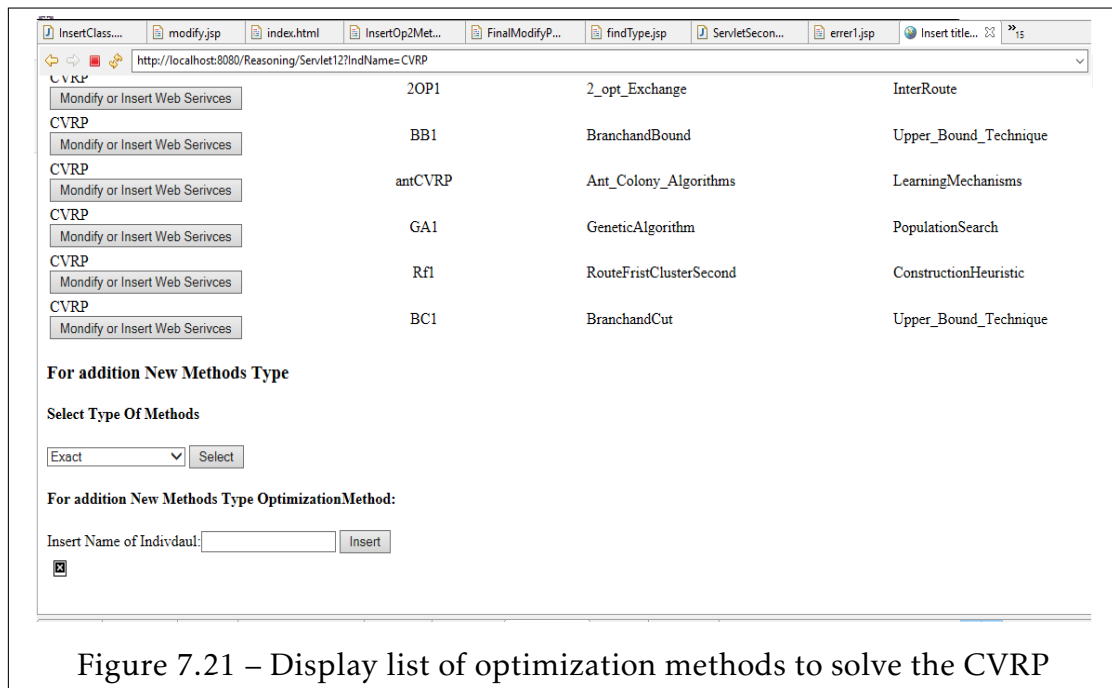


Figure 7.21 – Display list of optimization methods to solve the CVRP

For instance, when we select the type of logistics problem, as vehicle routing problem such as shown in figure 7.20. The reasoning system display the individuals of VRP. We can select an individual such as CVRP and add the new method. Hence, the reasoning system has the repository that contains the defined optimization methods to solve the CVRP, such as shown in figure 7.21. We can add new method by selecting the type of methods from the list, and we can add the individuals of the chosen method.

Likewise, we can add the web service for an existing method. We can select the option to add new webservice, from below the individual type of method. The added web-services are stored in the *Web-Service Database (WSDB)*. For example, we can add a web service for individuals of Genetic Algorithms method (GA1). Then the reasoning system asks the name of web services and the maximum nodes, that can be executed, such as shown in figure 7.22.

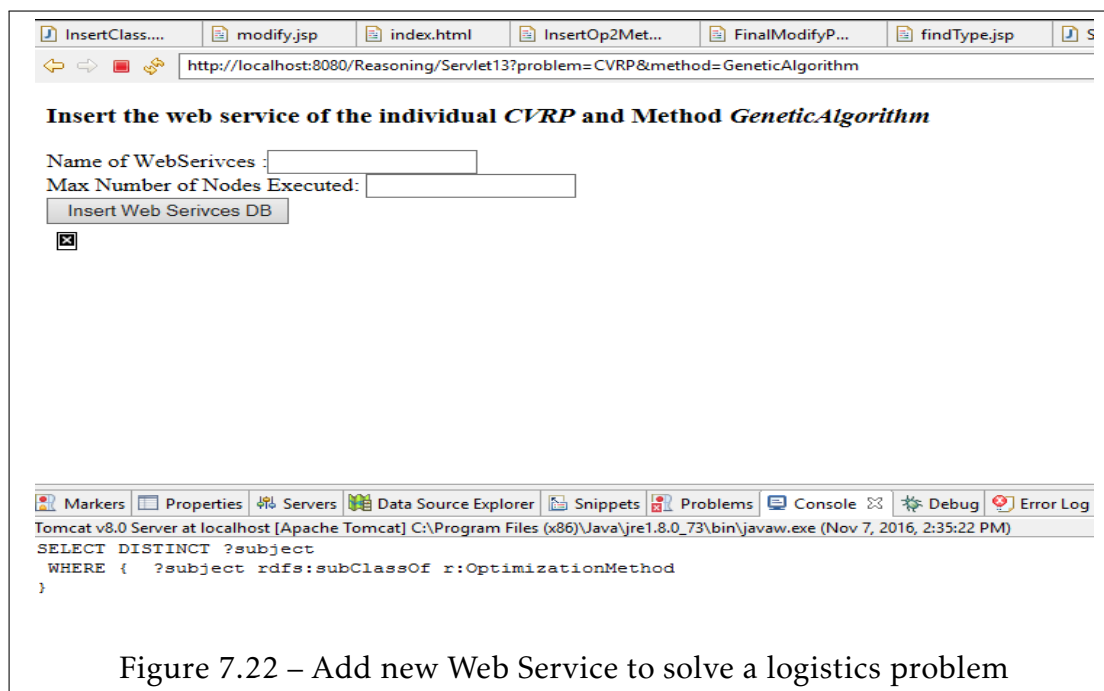


Figure 7.22 – Add new Web Service to solve a logistics problem

7.4 Conclusion

In the chapter, we discuss in detail the validity of proposed approach. For this purpose, we design and develop a software tool based on the logic's of a reasoning system. It has been developed, using the JEE technology, with a web-interface. We used OWL to define the ontologies, which can be interrogated with the help of SPARQL queries. The system, inherently manages the interaction of different repositories, such the repository of logistic ontologies, repository of optimization methods, repository of web-services, etc. The reasoning system incorporates the logistics processes and implements the optimization algorithms. It uses the Depth-First-Search algorithm to identify the Resource Concept Paths, in logistic ontologies.

We, initially explain the data flow among different constituents of the prototype system, and then describe its functionalities. The reasoning system provides the usable means to add or identify an existing logistic problem, and associate it with a newly added or existing optimization method. The optimization methods may have their implementation details, linked with the help of available web-services, which can be called. The prototype system can communicate with the

web-services and retrieve the processed data. At the end, we illustrate different functions of reasoning system with the help of screen captures.

Part IV

Conclusion

The Conclusion and Perspectives

8.1 Conclusion

A logistic process can be viewed as a complex amalgamate of activities. These activities are realized by different actors such the manufacturers, suppliers, and customers, which are often far apart. These distinct actors are implicated to manufacture, transport, and distribute the products. Each of these actors has their own, well-defined objectives. These objectives can many-a-times generate difficult conflicting situations.

An efficient logistic process, requires the better management of the production, storage, and transport of goods. It inherently requires better management of production chains, routing delivery details, inventory controls, and overall information processing. It is for this reason, we need to explicitly define the problem domain.

In this thesis, we consider the optimization problems of logistic transportation, such as the Vehicle Routing Problem and the Passenger Train Problem, to better illustrate our approach. The proposed approach is extensible, and may allow to include various other logistic problems. Hence, we further validate the proposed approach, with the help of its extension to cope with Container Terminal Problem.

In chapter 1, we aim to introduce the different concepts of a logistic network. We explore the problems, which are most susceptible to be encountered by a

logistic expert. We focus on the supply chain management. We explore the key issues at the strategic, tactical, and operational levels. Various logistic processes requires to be defined explicitly, in order to better understand the encountered problems. We observe that the logistic problems are complex combinatorial problems, which need to be optimized. The heterogeneity of involved problem domains, urges us to bridge the communication gap. The careful observation of the characteristics of a logistic problem reveals it to be an optimization problem. It is for this reason, we adopt an ontology-based approach, to define the common vocabulary to better define the logistic problems and index the associated optimization methods.

It is for this reason, in chapter 2, we introduce the ontologies and their capabilities to accommodate the logistic data. We explore the application scenarios of ontology, to find their relevance to logistics. The ontologies at the core are defined with the help of Description Logic. For the sake of clarity, we describe the description logic and its concept, as defined in the literature. We further explores the possibilities of its definition and interaction by the point of view of a user. It is therefore, we also describe the semantic web techniques from the existing literature. Such techniques provide the possibility to extend the World Wide Web, with the help of inter-linked ontology concepts. The Internet, initially had been passive, with mostly read-only contents. With the arrival of Web 2.0, it has been made possible to integrate user responds, consequently generating huge data bulks. These can be efficiently managed using the Semantic Web languages, like the RDF, and OWL, etc. These ontology schema can be interrogated with the help of structure query languages e.g., SPARQL. We explore the means to define the logistic and optimization concepts with the help of ontologies and semantic web techniques.

To pursue this objective, we attempt to define the explicit descriptions of logistics and optimization ontologies in chapter 3. We describe in details the concepts of supply chain management. We review the existing ontologies in the literature. The state of the art research presents some interesting approaches in this regard. The most inspirational among them have been SCOR (Supply Chain Operations Model), OrGoLo (Organizational Innovations *via* Good Governance in Logistics Networks), Semantic services application framework, ON-

TOP (Ontology for Optimization), and GOO (General Optimization Ontology). We infer the significance of the existing work in respect of logistic problems and their optimization methods. We extend this work to explicitly define the logistics and optimization ontologies in form of the repository schema. Which can be exploited with the help of queries.

In order to define the logistic problems and subsequently find their solutions with the help of ontologies, we proposed to design a reasoning system in chapter 4. We present the global architecture of the reasoning system to present its different functionalities at an abstract level. The proposed reasoning has been used as a prototype of validation for the defined strategy to deal with different logistic processes. It manages the interaction among the repositories that store the logistics and optimization ontologies, along with the artifact repository that represent the various artefacts related to logistics problems and optimization methods. The system allows a user to launch the queries with the help of a query engine, and later on to transfer the extracted information to the available web-services. Finally, it is capable to retrieve the processed information from the web-services.

We subsequently, define the logistic ontologies in chapter 5. The top level of logistic ontologies considers the general concepts of supply chain management. We specialized these concepts progressively with help of definitions of resource concepts, like the sub-classes of *Air Transport*, *Warehouse*, *Vehicle*, *Ship*, and *People*. In this regard, we identify the different actors like *Network Service Provider*, *Single Service Provider*, *System Service Provider*, etc. The implicated service concepts can be distinguished as different Logistic Services like *Packaging*, *Handling*, *Order Processing*, *Warehousing*, etc. We further define the sub-classes of *Goods*, and *Locations* to leave no room for confusion to define the logistic problems. It is therefore, we subsequently define the *Vehicle Routing Problems*, *Passenger Train Problems*, and *Container Terminal Problems*, using the defined concepts and inferred axioms.

The logistics problems, which have been characterized as optimization problems requires the optimization methods for their resolutions. In chapter 6, we explicitly define the optimization concepts and their associations to different individual variants of logistic problems. We consider the optimization as a

composition of optimization problems, methods, and their components. Hence, we need to provide the respective definitions of optimization problems such as the dynamic, linear-objective function, non-linear objective function, network, or stochastic problem. We need also to define the optimization methods as either exact or approximated (classified as either heuristic (further classified as either improvement or construction) or meta-heuristic) methods, etc. These ontology repositories are managed with the help of a reasoning system.

Moreover, in this regard, we opt to develop a prototype of validation for the presented approach. Hence, we discuss the further details in chapter 7 concerning the different aspect of the reasoning system. The reasoning system allows to enrich the logistic and optimization ontologies by a software developer. From the point of view of a logistic expert, it may provide the possibilities to identify and index the logistic problems. We show its execution scenario, in respect to identify the type of logistic problems and furthermore to find the optimization methods. These optimization methods can solve the identified problems. We test the reasoning system and show the results for the Capacity Vehicle Routing Problem (CVRP). We retrieve and compare the results for CVRP from two different web-services (optimoRoute and TrackRoad).

8.2 Perspectives

The logistics is a complex network of heterogeneous resources. We define an ontology-based approach for the definition logistic problems, and similarly, for their association with optimization methods, that can be used to solve these problems. For instance, we deal with three major problem sets, which are the *Vehicle Routing Problem (VRP)*, the *Passenger Train Problems (PTP)* and the *Container terminal Problems (CTP)*. In the short-term, we wish to complete some more individuals of logistics problem such as Airline Optimization Problems (Aircraft Load Planning Problems, Revenue Management Problems, and Schedule Planning Problems), the Supply Chain Management Problems (Location Management Problems, Inventories Management Problems) and also the People Management Problems.

In the middle-term perspectives, we wish to continue the enhancement of

optimization. We may develop and add more web services that can be used to solve the considered logistics problem, and for which either the web-services are not available or they does not sufficiently serve the purpose, such as: Rolling stack problem (RSP), Passenger Train Timetabling Problem (PTTP), etc. We intend to enlarge our working team with the collaboration of other research groups to develop the new web services for the optimization methods.

In the long-term, we intend to transform the reasoning system to be available online. It may permit an open access to integrate the web-services for other optimization methods. In this case, we can use WSDL and JSON to define the interface that connect the reasoning system with web services. Hence, young researchers can easily connect to these web services and alternatively to the reasoning system. It may reduce the time for them, to develop and/or compare the web services. We can also measure the ability of web services by using score values, it may use history of web services executions. In this regard, the reasoning system can use standard testing system for optimization web services.

Bibliography

- [1] Erwin Abbink. *Solving large scale crew scheduling problems by using iterative partitioning*. Tech. rep. Econometric Institute Research Papers, 2008.
- [2] Nilesh Anand, Ron van Duin, and Lori Tavasszy. “Ontology-based multi-agent system for urban freight transportation”. In: *International Journal of Urban Sciences* 18.2 (2014), pp. 133–153.
- [3] Nilesh Anand et al. “GenCLOn: An ontology for city logistics”. In: *Expert Systems with Applications* 39.15 (2012), pp. 11944–11960.
- [4] Franz Baader and Werner Nutt. “Basic description logics.” In: *Description logic handbook*. 2003, pp. 43–95.
- [5] Zahra Bahramian and Morteza Bagheri. “A simulation-based optimization approach for passenger train timetabling with periodic track maintenance and stops for praying”. In: *Journal of Modern Transportation* 23.2 (2015), pp. 148–157.
- [6] Eva Barrena et al. *A Fast and Efficient Adaptive Large Neighborhood Search Heuristic for the Passenger Train Timetabling Problem and Dynamic Demand*. CIRRELT, 2013.
- [7] Nicola Bianchessi and Giovanni Righini. “Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery”. In: *Computers & Operations Research* 34.2 (2007), pp. 578–594.
- [8] Olli Bräysy and Michel Gendreau. “Vehicle routing problem with time windows, Part II: Metaheuristics”. In: *Transportation science* 39.1 (2005), pp. 119–139.
- [9] Marco C Campi and Simone Garatti. “The exact feasibility of randomized solutions of uncertain convex programs”. In: *SIAM Journal on Optimization* 19.3 (2008), pp. 1211–1230.
- [10] Alberto Caprara, Leo Kroon, and Paolo Toth. “Optimization problems in passenger railway systems”. In: *Wiley Encyclopedia of Operations Research and Management Science* (2011).

- [11] John Carlsson et al. "Solving min-max multi-depot vehicle routing problem". In: *Fields Inst. Commun* 55 (2009), pp. 31–46.
- [12] Luís Cavique, César Rego, and Isabel Themido. "Subgraph ejection chains and tabu search for the crew scheduling problem". In: *Journal of the Operational Research Society* (1999), pp. 608–616.
- [13] Razouk Chafik, Youssef Benadada, and Jaouad Boukachour. "Stacking policy for solving the container stacking problem at a containers terminal." In: ().
- [14] Claudio Contardo and Rafael Martinelli. "A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints". In: *Discrete Optimization* 12 (2014), pp. 129–146.
- [15] Jam Dai et al. "Berth allocation planning optimization in container terminals". In: *Supply chain analysis*. Springer, 2008, pp. 69–104.
- [16] Mohamed Fayez, Luis Rabelo, and Mansooreh Mollaghasemi. "Ontologies for supply chain simulation modeling". In: *Simulation Conference, 2005 Proceedings of the Winter*. IEEE. 2005, 7–pp.
- [17] Dieter Fensel. "Ontologies". English. In: *Ontologies*. Springer Berlin Heidelberg, 2001, pp. 11–18. ISBN: 978-3-662-04398-1. DOI: 10.1007/978-3-662-04396-7_2.
- [18] William Gropp and Jorge J. Moré. "Optimization Environments and the NEOS Server". In: *Approximation Theory and Optimization*. Ed. by Martin D. Buhman and Arieh Iserles. Cambridge University Press, 1997, pp. 167–182.
- [19] Roberto Guerra-Olivares, Rosa G González-Ramírez, and Neale R Smith. "A Heuristic Procedure for the Outbound Container Relocation Problem during Export Loading Operations". In: *Mathematical Problems in Engineering* 2015 (2015).
- [20] Siddharth Gupta and Narina Thakur. "Semantic query optimisation with ontology simulation". In: *arXiv preprint arXiv:1011.0306* (2010).
- [21] Ceyhun Güven and Deniz Türsel Eliiyi. "Trip allocation and stacking policies at a container terminal". In: *Transportation Research Procedia* 3 (2014), pp. 565–573.
- [22] Jun Han, John A Miller, and Gregory A Silver. "SoPT: ontology for simulation optimization for scientific experiments". In: *Proceedings of the Winter Simulation Conference*. Winter Simulation Conference. 2011, pp. 2914–2925.

- [23] Jeff Heflin et al. "An Introduction to the OWL Web Ontology Language". In: *Lehigh University. National Science Foundation (NSF)* (2007).
- [24] Hayder I Hendi et al. "Ontology based reasoning for solving passenger train optimization problem". In: *Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA), Al-Sadeq International Conference on*. IEEE. 2016, pp. 1–6.
- [25] Hayder I Hendi et al. "Ontology Based Reasoning to Solve Container Terminals Problems". In: *The Fourth International Conference on Computer Science, Computer Engineering, and Education Technologies (CSCEET2017)*. IEEE. 2017, pp. 1–6.
- [26] Hayder Hendi et al. "LOGISTICS OPTIMIZATION USING ONTOLOGIES". In: *in Proceedings of ICCSA 2014, The 4th international Conference on Complex Systems and Applications, Normandy University - Le Havre, France June 23 - 26, 2014*. 2014.
- [27] Frederick S Hillier. *Introduction to operations research*. Tata McGraw-Hill Education, 1995.
- [28] André Hottung and Kevin Tierney. "A biased random-key genetic algorithm for the container pre-marshalling problem". In: *Computers & Operations Research* 75 (2016), pp. 83–102.
- [29] Julia Hoxha, Andreas Scheuermann, and Stephan Bloehdorn. "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*. 2010, pp. 73–78.
- [30] Dennis Huisman et al. "Operations research in passenger railway transportation". In: *Statistica Neerlandica* 59.4 (2005), pp. 467–497.
- [31] Dewan Md Zahurul Islam et al. "Logistics and supply chain management". In: *Research in Transportation Economics* 41.1 (2013), pp. 3–16.
- [32] Charlotte Diane Jacobs-Blecha and Marc Goetschalckx. *The Vehicle Routing Problem with Backhauls: Properties and Solution Algorithms*. Material Handling Research Center, Georgia Institute of Technology, 1992.
- [33] Robert Jasper, Mike Uschold, et al. "A framework for understanding and classifying ontology applications". In: *Proceedings 12th Int. Workshop on Knowledge Acquisition, Modelling, and Management KAW*. Vol. 99. 1999, pp. 16–21.

- [34] Yannis Kalfoglou and Marco Schorlemmer. "Ontology mapping: the state of the art". In: *The Knowledge Engineering Review* 18 (01 Jan. 2003), pp. 1–31. ISSN: 1469-8005.
- [35] Yasanur Kayikci and Helmut Zsifkovits. "Successful ICT Integration in Transport Collaboration". In: *Pioneering Solutions in Supply Chain Performance Management: Concepts, Technologies and Applications* (2013), p. 237.
- [36] Martin Kowalski et al. "Application of new techniques of artificial intelligence in logistics: an ontology-driven case-based reasoning approach". In: *Proceedings of European Simulation and Modelling Conference*. 2012, pp. 323–328.
- [37] Markus Krötzsch, Frantisek Simancik, and Ian Horrocks. "A description logic primer". In: *arXiv preprint arXiv:1201.4089* (2012).
- [38] Dusan Ku and Tiru S Arthanari. "Container relocation problem with time windows for container departure". In: *European Journal of Operational Research* 252.3 (2016), pp. 1031–1039.
- [39] Nacima Labadi, Christian Prins, and Mohamed Reghioui. "A memetic algorithm for the vehicle routing problem with time windows". In: *RAIRO-Operations research* 42.03 (2008), pp. 415–431.
- [40] Joerg Leukel and Stefan Kirn. "A supply chain management approach to logistics ontologies in information systems". In: *Business Information Systems*. Springer. 2008, pp. 95–105.
- [41] Ming Liu et al. "Framework branch-and-price algorithm for yard management problem at container terminals". In: *2016 IEEE 13th International Conference on Networking, Sensing, and Control (ICNSC)*. IEEE. 2016, pp. 1–6.
- [42] Rhonda R Lummus, Dennis W Krumwiede, and Robert J Vokurka. "The relationship of logistics to supply chain management: developing a common industry definition". In: *Industrial Management & Data Systems* 101.8 (2001), pp. 426–432.
- [43] Jiabin Luo, Yue Wu, and André Bergsten Mendes. "Modelling of integrated vehicle scheduling and container storage problems in unloading process at an automated container terminal". In: *Computers & Industrial Engineering* 94 (2016), pp. 32–44.
- [44] Azad M Madni, Weiwen Lin, and Carla C Madni. "IDEONTM: An extensible ontology for designing, integrating, and managing collaborative distributed enterprises". In: *Systems Engineering* 4.1 (2001), pp. 35–48.

- [45] Christopher Matheus et al. "Using SWRL and OWL to capture domain knowledge for a situation awareness application applied to a supply logistics scenario". In: *Rules and Rule Markup Languages for the Semantic Web* (2005), pp. 130–144.
- [46] VC Moussas, J Tsahalis, and HT Tsahalis. "design of an ontology for simulation workflow optimization". In: *Proceedings of 5th international conference on experiments/process/system modeling/simulation/optimization*. 2013.
- [47] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.
- [48] Jiff Ocenasek and Josef Schwarz. "Estimation of distribution algorithm for mixed continuous-discrete optimization problems". In: *2nd Euro-International Symposium on Computational Intelligence*. IOS Press Kosice, Slovakia. 2002, pp. 227–232.
- [49] Şule Onsel. "Operations research–applications, edited by A. Ravi Ravindran". In: *International Journal of Production Research* 47.23 (2009), pp. 6781–6782.
- [50] Sophie N Parragh, Karl F Doerner, and Richard F Hartl. "A survey on pickup and delivery problems". In: *Journal für Betriebswirtschaft* 58.1 (2008), pp. 21–51.
- [51] Dirk Pawlaszczyk et al. "Ontologies supporting cooperations in mass customization-a pragmatic approach". In: (2004).
- [52] Christian Prins, Philippe Lacomme, and Caroline Prodhon. "Order-first split-second methods for vehicle routing problems: A review". In: *Transportation Research Part C: Emerging Technologies* 40 (2014), pp. 179–200.
- [53] A Ravi Ravindran. *Operations Research Applications*. CRC Press, 2008.
- [54] Markus Reuther et al. "Integrated optimization of rolling stock rotations for intercity railways". In: (2012).
- [55] Leonard Richardson and Sam Ruby. *RESTful web services*. " O'Reilly Media, Inc.", 2008.
- [56] Sebastian Rudolph. "Foundations of description logics". In: *Reasoning Web. Semantic Technologies for the Web of Data*. Springer, 2011, pp. 76–136.
- [57] Alan Rushton, Phil Croucher, and Peter Baker. *The handbook of logistics and distribution management: Understanding the supply chain*. Kogan Page Publishers, 2014.

- [58] A Saremi, TY ElMekkawy, and GG Wang. “Tuning the parameters of a memetic algorithm to solve vehicle routing problem with backhauls using design of experiments”. In: *International Journal of Operations Research* 4.4 (2007), pp. 206–219.
- [59] Karl Sigurjonsson. “Taboo search based metaheuristic for solving multiple depot vrppd with intermediary depots”. PhD thesis. Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2008.
- [60] D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi. *Designing and Managing the Supply Chain: Concepts, Strategies, and Case Studies*. McGraw-Hill International Edition. McGraw-Hill/Irwin, 2008. ISBN: 9780071287142. URL: <https://books.google.fr/books?id=SURypwAACAAJ>.
- [61] Evren Sirin and Bijan Parsia. “SPARQL-DL: SPARQL Query for OWL-DL.” In: *OWLED*. Vol. 258. 2007.
- [62] Hartmut Stadtler. *Supply chain management—an overview*. Springer, 2002.
- [63] P Surekha and S Sumathi. “Solution to multi-depot vehicle routing problem using genetic algorithms”. In: *World Applied Programming* 1.3 (2011), pp. 118–131.
- [64] Paolo Toth and Daniele Vigo. *The vehicle routing problem*. SIAM monographs on discrete mathematics and applications. SIAM : Society for Industrial and Applied Mathematics. Philadelphia (Pa.): SIAM, 2002. ISBN: 0-89871-498-2.
- [65] Yung-yu Tseng, Wen Long Yue, Michael AP Taylor, et al. “The role of transportation in logistics chain”. In: Eastern Asia Society for Transportation Studies. 2005.
- [66] Yukihide Tsuji et al. “Ant Colony Optimization approach for solving rolling stock planning for passenger trains”. In: *System Integration (SII), 2012 IEEE/SICE International Symposium on*. IEEE. 2012, pp. 716–721.
- [67] Michael Uschold and Michael Gruninger. “Ontologies and semantics for seamless connectivity”. In: *ACM SIGMod Record* 33.4 (2004), pp. 58–64.
- [68] Joris Wagenaar, Lucas P Veelenturf, Leo G Kroon, et al. “A Comparison of Two Exact Methods for Passenger Railway Rolling Stock (Re) Scheduling”. In: (2015).
- [69] Yujun Wang et al. “Developing an ontology-based cold chain logistics monitoring and decision system”. In: *Journal of Sensors* 2015 (2015).

-
- [70] Thomas Weise, Alexander Podlich, and Christian Gorltd. “Solving real-world vehicle routing problems with evolutionary algorithms”. In: *Natural intelligence for scheduling, planning and packing problems*. Springer, 2009, pp. 29–53.
- [71] Thomas Weise, Alexander Podlich, and Christian Gorltd. “Solving real-world vehicle routing problems with evolutionary algorithms”. In: *Natural intelligence for scheduling, planning and packing problems*. Springer, 2009, pp. 29–53.
- [72] Paul Witherell, Sundar Krishnamurty, and Ian R Grosse. “Ontologies for supporting engineering design optimization”. In: *Journal of Computing and Information Science in Engineering* 7.2 (2007), pp. 141–150.

Abstract

Le domaine de la logistique implique souvent la résolution de problèmes combinatoires complexes. Ces derniers font également implicitement référence à des processus, acteurs, activités et méthodes concernant divers aspects qu'il faut considérer. Ainsi, un même problème peut faire intervenir des processus de vente/achat, transport/livraison et gestion de stock. Ces processus sont tellement divers et interconnectés qu'il est difficile pour un logisticien de tous les maîtriser. Dans cette thèse, nous proposons l'explicitation, par le biais d'ontologies, de connaissances conceptuelles et sémantiques concernant les processus logistiques. Ces connaissances explicites sont alors mises à contribution pour construire un système à base de connaissances permettant de guider les logisticiens dans la construction, de façon incrémentale et semi-automatique, de solutions informatiques à un problème qui leur est posé à un moment donné. Nous mettons en oeuvre une ontologie concernant le domaine de logistique connectée à une ontologie associée à la problématique de l'optimisation. Nous établissons ainsi un lien sémantique explicite entre le domaine de la logistique et celui de l'optimisation. Cela permet aux logisticiens d'identifier de façon précise et sans ambiguïté le problème logistique auquel il est confronté et les problèmes d'optimisation associés. L'identification des problèmes conduit alors à un processus de choix des solutions allant du choix du processus logistique précis à mettre en oeuvre à celui de la méthode de résolution du problème combinatoire et cela jusqu'à la découverte du composant informatique à invoquer et qui est matérialisé par un service web. L'approche que nous avons adoptée et mise en oeuvre a été expérimentée avec les problèmes de routage de véhicules, le problème de transport ferroviaire de passagers et le problème de terminaux de conteneurs.

Keywords: semantic web, ontology, logistic, optimization, web services, owl, description logic, vehicle routing problem, passenger train problem, container terminal problem

LISIC

Le Laboratoire d'Informatique Signal et Image de la Côte d'Opale – Maison de la Recherche Blaise Pascal – 50, rue Ferdinand Buisson – CS 80699 – 62228 Calais Cedex – France

Abstract

Logistic problems are often complex combinatorial problems. These may also implicitly refer to the processes, actors, activities, and methods concerning various aspects that need to be considered. Thus the same process may involve the processes of sale/purchase, transport/delivery, and stock management. These processes are so diverse and interconnected that it is difficult for a logistic expert to compete all of them. In this thesis, we propose the explications with the help of ontologies of conceptual and semantic knowledge concerning the logistic processes. This explicit knowledge is then used to develop a reasoning system to guide the logistic expert for an incremental and semi-automatic construction of a software solution to an instantly posed problem. We define an ontology concerning the inter-connected logistics and associated optimization problem. We, henceforth, establish an explicit semantic link between the domains of logistics and the optimization. It may allow the logistic expert to identify precisely and unambiguously the confronted logistic problem and the associated optimization problem. The identification of the problems then leads to a process to choose the solutions ranging from the choice of the precise logistic process to be implemented to that of the method to solve the combinatorial problem until the discovery of the software component to be invoked and which is implemented by a web service. The approach we have adopted and implemented has been experimented with the *Vehicle Routing Problems*, the *Passenger Train Problems*, and the *Container Terminal Problems*.

Keywords: semantic web, ontology, logistic, optimization, web services, owl, description logic, vehicle routing problem, passenger train problem, container terminal problem
