



HAL
open science

Ontology Based Reasoning for Solving Passenger Train Optimization Problem

Hayder I Hendi, Adeel Ahmad, Mourad Bouneffa, Cyril Fonlupt

► **To cite this version:**

Hayder I Hendi, Adeel Ahmad, Mourad Bouneffa, Cyril Fonlupt. Ontology Based Reasoning for Solving Passenger Train Optimization Problem. 2016 Al-Sadeq International Conference on Multi-disciplinary in IT and Communication Science and Applications (AIC-MITCSA), May 2016, Bagdad, Iraq. pp.9 - 10. hal-01651103

HAL Id: hal-01651103

<https://hal.science/hal-01651103v1>

Submitted on 28 Nov 2017

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.

Ontology Based Reasoning for Solving Passenger Train Optimization Problem

Hayder I. Hendi

University of Littoral Cote d'Opale (ULCO), Calais, France
hendi@lisic.univ-littoral.fr

Adeel Ahmad

University of Littoral Cote d'Opale (ULCO), Calais, France
ahmad@lisic.univ-littoral.fr

Mourad Bouneffa

University of Littoral Cote d'Opale (ULCO), Calais, France
bouneffa@lisic.univ-littoral.fr

Cyril Fonlupt

University of Littoral Cote d'Opale (ULCO), Calais, France
fonlupt@lisic.univ-littoral.fr

Abstract—The Logistic processes generally lead to complex physical flows dealing with various logistic elements. It has been widely observed that the quality of uncontrolled processes decline with the evolving complexity. It may make them incoherent and quasi obsolescent. Thus, the continuous optimization of logistic processes is essential for the consistent continuity of logistic activities, and henceforth, it supports their desired growth. In this paper, we propose a reasoning system that uses the conceptual domain of logistics and their optimization. The proposed approach is mainly based on the definition of logistic terminologies using ontology. We intend, that a logistic expert may use defined terms to specify a problem. These can be matched to extract the involved logistic processes. It may assist a logistics expert to identify and/or precisely specify the logistic problem. Furthermore, it may identify the logistic processes, that can be executed, to resolve the problem and consequently to resolve the inherent optimization problems. We have been experimenting the different solutions of the *Passenger Transportation Problem* and eventually built a software framework (based on the composition of web services), to semi-automatically assist the resolution procedure of identified optimization problems.

Keywords: *Passenger Transportation Problem, Logistics Ontology, Optimization Ontology, SPARQL, OWL.*

I. INTRODUCTION

The logistic processes specify various interconnected artifacts such as events, activities, and actors. The process improvement may generate difficult situations due to the intrinsic complexity of logistics systems [1]. Similarly, the physical or real world constituents of logistic processes must also evolve to meet the evolution of logistic artifacts.

This paper addresses the evolution of logistic processes using ontologies. The ontologies are defined in order to semantically specify the various logistic artifacts. It may allow the maintaining of the traceability relationships between logistics artifacts and software components implementing them. There exist a large variety of logistics problems including some important ones like the *Vehicle Routing Problems (VRP)* and the *Passenger Transportation Problems (PTP)*. In this paper, we focus on the *Passenger Transportation Problems*. The aim of this problem is to assist logisticians to determine the optimal schedule organizing the railway travels. The scheduling concerns both the trains and the staff. The optimization processes concerns both time and cost. PTP is considered as a combinatorial optimization problem [2].

The approach we develop is intended to solve several optimization problems by implementing different optimization methods such as exact algorithms (linear programming, etc.), heuristics (large neighborhood heuristic, etc.) [3], or meta-heuristic algorithms (ant colony algorithm, tabu search, genetic algorithm, etc.) [4]. The main objective of our work is to build a semi-automatic framework assisting logisticians to both specify and implement some logistics problems as well as the PTP. We specify and implement two ontologies defining the semantics of both logistics and optimization concepts. The ontologies serve as a knowledge base addressed by queries representing the expert's questions, such that which heuristic methods can solve the PTP etc.

Our implementation is mainly based on the use of the web semantic concepts and tools including the OWL (Ontology Web Language) [5] to specify the ontologies and SPARQL [6] to implement queries.

The rest of the paper is organized as follows: the section 2 describes the related work. The PTP is explained in the section 3. The section 4 shows the general activities of the proposed reasoning system. The sections 5 and 6 respectively show the logistics and the optimization ontologies using concepts of description logic. The section 7 shows the validation of our approach with the help of a query engine and describing its functionally aspects. In section 8 we conclude the paper by resuming the results and the perspectives of our work.

II. RELATED WORK

In the literature, a lot of work has been devoted to the definition of logistic ontologies. Initially, the literary work has been focused on the analysis of logistic problems from the simulation and modeling perspectives [7]. Several individual ontologies have been defined to represent logistics concepts. We consider three of them that use different strategies to define the general concepts concerning the supply chain management.

Leukel describes the logistics model by defining the five process types: *plan*, *source*, *make*, *deliver*, and *return*. The work also defines relationship types that links process types and the other concepts like *Company* and *Good* ones. But the work has been primarily limited to define the sequence of processing in logistics system, and may further require the definition of the logistics object [8].

Kowalski shows an ontology by measuring the similarity between knowledge collections written in natural languages. The work develops a logistics ontology based on some linguistic analysis tools. This ontology defines Object at the

root class and derives two sub-classes representing physical and abstract objects. Physical objects represent real world objects such as (mean of transportation, Good, etc.). The abstract objects include some success factors such as logistics KPI (Key Performance Indicators) [9].

Hoxha discusses different functionalities involved in the logistics domain. The work defines top level classes of logistics ontology concerning four concepts (*Process, Service, Resource, and logistics KPI*) and make simulations concerning the performance of train transportation network. It lacks the definition of the concepts for logistics problem [10].

Some other individual ontology has been devoted to define optimization concepts such as SoPT [11], ONTOP [12] and GOO [13].

SoPT (Simulation oPTimization) ontology includes concepts from both conventional mathematical programming and the simulation optimization. It is yet unclear to consider such ontology for complex logistic problems. It focuses to classify the optimization component into many subclasses but with limited number of problems and methods; therefore it lacks to classify the optimization problem and methods [11].

ONTOP (Ontology for Optimization) has been developed to facilitate engineering problems. The preliminary work began with the development of a Finite Element Model (FEM) knowledge-capturing tool. ONTOP's structure provides the means to identify feasible optimization techniques for a given design optimization problem, it included one concept in top level, which is optimization type. It classifies the optimization type concept depending on continuous or discrete problem (either constrained or unconstrained). This classification is useful for limited number of problems and develops general ontology but it has inconsistencies between concepts [12].

Likewise, GOO (General Optimization Ontology) has been designed and structured with the main focus on the optimization. The authors develop the top level of ontology by specifying the concepts concerning the optimization problems and methods. The optimization component and the problem class define only five problems [13].

III. PASSENGER TRANSPORTATION PROBLEM

The railway planning is a complex activity which is usually composed of a set of successive stages including network design, line design, timetabling, rolling stock and finally crew planning [3].

In this paper, we deal with the *Passenger Transportation*. Its planning is decomposed into four sequential phases:

- *Line planning*: deciding the trips for the passenger or cargo trains, as well as the types and frequencies of the trains for each trip.
- *Timetabling*: fixing the timetable for each train. This leads to a problem called the *Passenger Train Timetabling Problem (PTTP)*.
- *Rolling Stock Circulation*: defining train units (locomotives and train carriages) to be assigned to the trains. The subsequent problem is called the *Rolling Stock Problem (RSP)*.
- *Crew planning*: defining the workload of train drivers and conductors to operate a given timetable (*Crew Scheduling Problem or CSP*).

The passenger transportation Planning and operational processes are rich in challenging *Combinatorial Optimization Problems* [15]. There are several optimization methods that can be implied to solve various *PTP*. We discuss some of

these methods. For example, concerning the *Passenger Train Timetabling Problem (PTTP)*; Eva [3] shows two optimization methods to solve it. It uses the *Integer Linear programming* and *Large Neighborhood heuristic* and compares the results from each one of them [3]. Zahra [16] proposes the use of cellular automata (CA) with genetic algorithm (GA) to solving the PTTP. Many methods have been used to solve the *Rolling Stock Problem (RSP)*. In [4] Vasutaka use a method based on ant colonies (ACO: Ant Colonies Optimization) to solve this problem. In [14], Reuther shows a solution to the RSP problem using a heuristic method (generic hyper graph) with a mixed integer linear program (MILP). Jorgen [17] presents and compare two approaches to solve this problem. He primarily uses the CPLEX to solve mixed integer linear program (MILP) and then uses column generation algorithm.

The *Crew Scheduling Problem (CSP)* is common among other problems but with different constraints such as crew for airline and crew for vehicles. There are many methods solving this problem such as sub graph ejection and tabu search that are proposed by Cavique [18]. Erwin proposed an iterative partitioning method [19].

IV. THE LOGISTIC REASONING SYSTEM

We develop a reasoning system that implements logistic processes using specific terminology. The proposed system implies the semantic exploitation of logistics domain concepts using ontologies. These ontologies constitutes the departure point of a reasoning process allowing the logistics experts to express their requirements by means of some terms. Such terms are then matched to the ontologies leading to discover the associated, yet categorized, logistics problems. The system then initiates a dialog with logistic expert leading to identify the concerned processes and optimization methods. It eventually leads to software services that implement the identified methods.

The Fig 1 summarizes the global schema of the proposed logistic reasoning system. The process builder facilitates the identification of the logistics process concerned with the problem expressed by an expert. In fact, the expert just enumerates some terms (or words). These terms must be chosen from a catalog containing all the terms appearing in the logistics ontologies. In reality these terms represent the Resource Concepts (RC) of the problem. The process builder then matches these terms with their occurrence in the stored processes and extract processes, which are assumed to match the expert needs. An interactive process allows the expert to choose the corresponding process and then the desired optimization methods and software services implementing them.

The query engine interacts with a database that contains the various artefacts composing logistics processes and software constituents implementing them. The semantics of such artefacts are formalized by ontologies. We distinguish two kind of ontologies: the logistics ontologies that explicit the semantic of the various logistics artefacts (processes, actors, activities or tasks ...) and the optimization ontologies formalizing the semantics of the optimization methods, problems and tools (software). The query engine uses both SPARQL and SQL languages. The database is implemented using a relational DBMS (MySQL) while the ontologies are materialized by OWL files. The logistics and optimization ontologies are described in more detail in section V and section VI.

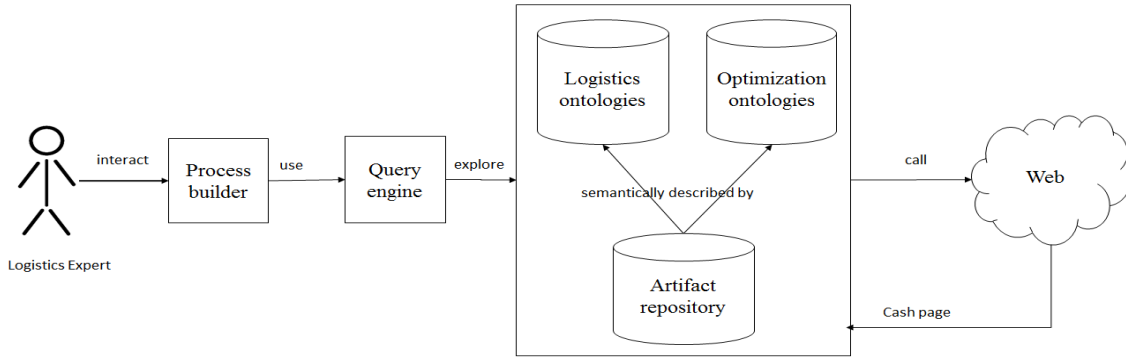


Fig 1. The global schema Logistic Reasoning System

The *Artifact Repository* represents both stored (persistent data) and temporary data. This repository is composed of two kinds of databases: the *Conceptual Path Database (CPDB)* and the *Article Database (ADB)*. The CPDB represents paths in which nodes represent terms or ontology concepts and edges represent relationships linking these terms. Each one of these paths represents a certain logistic or optimization problem. In fact, we represent the problem by the set of concepts (including resources) involved in that problem. The ADB represents article pages (by means of URLs) dealing with a specific problem.

V. LOGISTICS ONTOLOGIES

The core of the logistics ontologies is to know the essence of the logistics domain. This is achieved by means of semantic constructs using terms like concepts (or classes) and the relationships linking these concepts, roles and axioms. We define a logistics ontology including top level classes as *Process*, *Service*, *Resource*, *Performance*, *Activity*, and *Logistics Problem*. These are further extended into subsidiary classes e.g. *Logistic Process* is a sub-class of *Process*. Likewise, the *Logistics Service* is a sub-class of *Service*, etc. Hence, we have defined more than 50 classes [20].

There are many roles connecting the concepts, such as *compose* that connects the *Process* class to the *Service* class and the *uses* role connecting the *Service* class to the *Resource* class, etc. Some of the concepts presented here has yet been introduced in [18].

In this paper, we extend the ontologies defined in [20] with the concepts related to logistic problems and we focus the ones concerning the *PTP*. Therefore, we introduce new classes (or concepts), relationships (roles and axioms) concerning this problem. Furthermore, we define concepts like *Timetabling* class (that is a sub-class of *Activity*), *Passenger transportation planning (PTL)* class (sub-class of *Logistics Process*) and *Passenger Transportation Problems (PTP)* class (sub-class of *Logistics Problems*).

In Fig 2, we show relationships (or roles) between the newly defined concepts such as *ManageTrain* (sub-role of *UseResource*) that connects *Timetabling* activity and *Train* classes; *ManageCrew* (sub role of *UseResource*) that connects *People* and *Timetabling* activity, *hasTime* (sub-role of *has-a*) that connects *Passenger transportation planning process* and *Timetabling* activity classes and *hasObjectiveFunction* between *Passenger Transportation Planning (PTL)* and *Objective Function*.

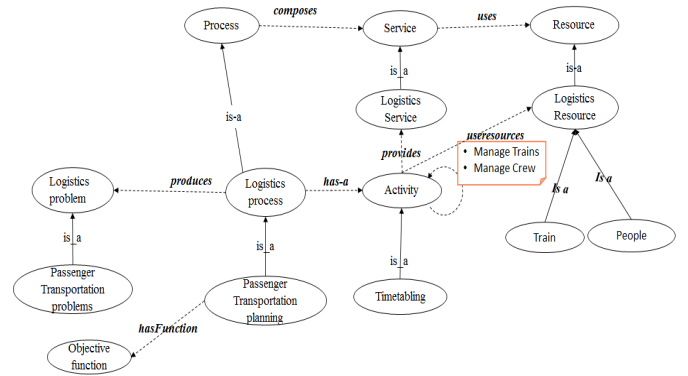


Fig 2. Top Level of PTP ontology

Axiom 1: The class (or concept) *Train* defines instances of *Logistics Resource* having a property *TravelTime* of class *Timetabling*. Notably, *People* is a sub-class of *Logistics Resource* with a property *WorkTime* of *Timetabling*; Similarly *PTP* is a sub-class of *Railway Transportation Problems* class.

This may be formalized by the following Description Logic assertions [21] :

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

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

$$PTP \sqsubseteq RailwayTransportationProblems$$

Axiom 2: *Passenger Transportation pLanning (PTL)* is a sub-class of *Logistics Process*. Every *PTL* has a role or property *hasTime* of *Timetabling* *Activity* class; a role *hasObjectiveFunction* of *Objective Function* class and a role *produces* that is a collection of *PTP*.

PTL

$$\sqsubseteq LogisticsProcess \cap \exists hasTime.Timetabling$$

$$\cap \exists hasObjectiveFunction.ObjectiveFunction \cap \exists produce.PTP$$

The *PTP* is composed of four other problems that are: the *Rolling Stock Problem (RSP)*, the *Crew Scheduling Problem (CSP)*, the *Passenger Train Timetabling Problems (PTTP)* and the *Line Train Planning Problem (LTPP)*. The following axioms define more precisely each one of these problems.

Axiom 3: The *Rolling Stock Problem (RSP)* is a sub-class of the *Passenger Transportation Problems (PTP)*, produced by a *Passenger Transportation pLanning (PTL)* having a property *hasTime* of a *Timetabling* managing *Trains* or *Peoples*. The

RSP has two *Objective functions* that are of *MinTravelCost* and *MaxNumberPassenger* classes.

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

Axiom 4: The *Crew Scheduling Problem (CSP)* is a sub-class of the *PTP*, produced by a *PTL* having a property *hasTime* of a *Timetabling* managing *Trains* or *Peoples*. The *CSP* has one *Objective function* that is of *MinNumberStaff* (Minimize the Number of Working Staff) class.

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

Axiom 5: *Passenger Train Timetabling Problem (PTTP)* a sub-class of the *PTP*, produced by a *PTL* having a property *hasTime* of a *Timetabling* managing *Trains* or *Peoples*. The *PTTP* has one *Objective function* that is of *MinWaitTimePassengers* (Minimize the waiting time for passengers) class.

$$\begin{aligned} PTTP \\ \subseteq PTP \\ \cap \exists \text{producedby.} (PTL \\ \cap \exists \text{hasTime.} (Timetabling \cap \exists \text{ManageTrain.} Train \\ \cap \exists \text{ManagePeople.} People) \\ \cap \exists \text{hasObjectiveFunction.} MinWaitTimePassengers) \end{aligned}$$

Axiom 6: The *Line Train Planning Problem (LTPP)* is a sub-class of the *PTP*, produced by a *PTL* having a property *hasTime* of a *Timetabling* managing *Trains* or *Peoples*. The *LTPP* has one *Objective function* that is of *MinTravelCost* (Minimize the travel cost) class.

$$\begin{aligned} LTPP \subseteq PTP \cap \exists \text{producedby.} (PTL \\ \cap \exists \text{hasTime.} (Timetabling \\ \cap \exists \text{ManageTrain.} Train) \\ \cap \exists \text{hasObjectiveFunction.} MinTravelCost) \end{aligned}$$

VI. OPTIMIZATION ONTOLOGY

The use of optimization ontology aims to identify typical optimization problems along with the descriptions of the methods applied to solve an optimization task. The basic structure of these ontologies should support optimization processes. The optimization can be categorized into many concepts as follows:

- *Optimization problem concepts,*
- *Optimization method concepts,*
- *Optimization component concepts.*

The ontology defined in [20] has been extended by the concepts of *Optimization components*, *Optimization problems*, and *Optimization methods*. The Fig 3 shows these concepts in terms of classes. The Optimization Ontology, in our proposed approach, contains more than 50 classes.

Axiom 7: The *Logistics optimization problem* is sub-class of *Optimization problem*, and it applies all roles as either *solve* or *hasComponent*.

$$\begin{aligned} Logistics\ optimization\ problem \\ \subseteq Optimization\ problem \end{aligned}$$

Axiom 8: An optimization method can be an *Exact Method* or an *Approximate Method*. An *Approximate Method* can be a *Heuristics Method* or a *Meta-Heuristics Method*. A *Heuristics*

Method can be an *Improvement Method* or a *Construction Method*.

$$\begin{aligned} Optimization\ Method \\ \equiv ExactMethods \cup ApproximateMethods \end{aligned}$$

$$\begin{aligned} Approximate\ methods \\ \equiv HeuristicsMethods \\ \cup MetaheuristicsMethods \end{aligned}$$

$$\begin{aligned} HeuristicsMethods \\ \equiv ImprovementMethods \\ \cup ConstructionMethods \end{aligned}$$

Many methods have been defined in optimization ontologies, but still the optimization methods to solve PTP are limited such as ant colony method, large neighborhoods heuristic, and branch and cut methods.

Therefore, we build the knowledge base that can have exhaustive number of methods, which can be applied to solve PTP.

Axiom 9: *PTP* is a sub-class of the *Logistics Optimization Problems (LP)* that can be solved by many *Optimization Methods (OM)* and has many *Optimization Component (OC)*.

$$\begin{aligned} PTP \subseteq LogisticsOptimizationProblems \\ \cap \exists \text{solveBy.} OptimizationMethods \\ \cap \exists \text{has.} OptimizationComponent \end{aligned}$$

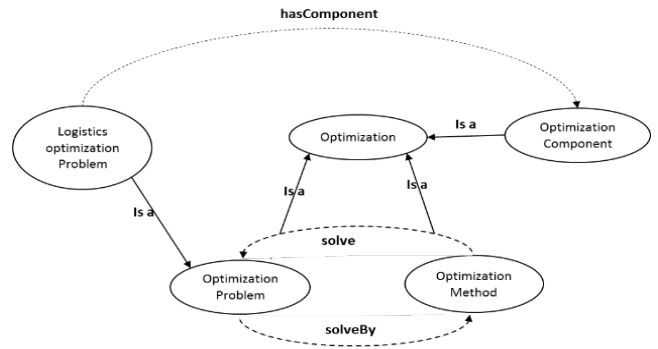


Fig 3. Top level optimization ontology

VII. THE ARTEFACT REPOSITORY

The main objective of the artefact repository is to store the useful data classified into temporary or transient data and the database or persistent data. The temporary data comprise the received data from the process builder such as *Resources Concepts (RC)*, the results *Resource Concepts Paths (RCPs)* and the results of the queries such as the various types of problems and associated resolution methods.

As narrated earlier, in section 4, the Problem Repository includes two databases:

- *CPDB* store all *RCPs* for a problem. It is, in fact, possible to find the type of *Logistics Problems* automatically, by using *Selected Logistics Resources* to find *RCPs* and therefore compare the resulted *RCPs* with *CPDB* to identify the type of a *logistics Problem*.
- *ADB* store some information related to the articles such as name of article, authors and URL. The major aim of such a database is to make it easy to find collections of articles dealing with a particular problem and corresponding resolution method.

VIII. THE VALIDATION OF LOGISTIC REASONING SYSTEM

The conceptualization of logistics and optimization domains has been used to build the Logistic Reasoning System in order to assist logistic experts to identify their particular logistic problems and related optimization problems and methods to solve them. The system also assists the users to find related article pages describing the identified problems. In order to more precisely show the functionalities provided by the system, let us explain its use through a sequence of tasks.

A. Finding a general logistics problem type

When a logistic expert interacts with the Logistic Reasoning System, (s)he explains his *Logistics RCs* and the system extracts the corresponding *Logistics RCPs*. To do this, the system uses either the Depth-First-Search (DFS) or Breadth-First-Search (BFS). In this particular case, the RCs are *Train* and *People* and the DFS is more convenient (in terms of memory and speed) than BFS because of the wider structure of inter-linked RCs.

The algorithm 1 can be used to show resulted RCPs of DFS for each RC, selected by the logistic expert. It also eliminates (delete) the Similar RCPs and temporarily stores the resulting RCPs.

Algorithm 1 Find RCPs

```

1: procedure SELECT-PATH
2:   connectedtologisticsontology;
3:   for each rc ∈ RC do
4:     RCP[i][j] ← FirstDepthSearch(rc[i])
5:   end for
6:   for each rc[i] ∈ RC do
7:     for each rep[i][j] ∈ RCP[i] do
8:       for k = j to N do
9:         if rep[i][j] == rep[i][k] then delete(rep[i][k])
10:        end if
11:      end for
12:    end for
13:  return RCPs
14: end procedure

```

As shown in Fig 4, the resulting RCPs, can be determined by following the inter-linking Roles between RC and the general type of logistics problem (generated automatically) by the logistic reasoning system .

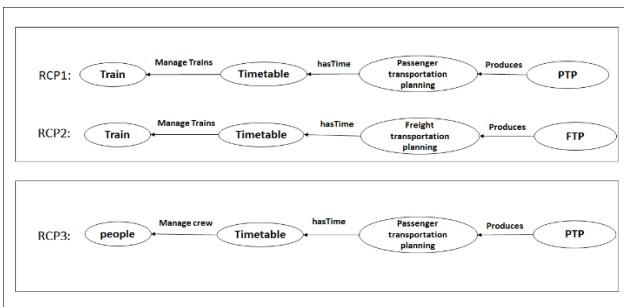


Fig 4. Example of RCPs

In this example, we use the Roles *ManageTrain*, *hasTime*, and *Produces* to identify the RCP, i.e:

RCP1: Train >> Timetable >> PTP and eventually PTP.

We divide the RCPs into Similar Concept Path Groups (SCPG). The groups are divided on the basis of most common premier concepts of the RCPs. 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.

For more illustration, let us consider, once again, the example shown in Figure 4.

Name of problem	Vehicle	Warehouse	Trains	Ship	Flight	people
Passenger Train Problems	Null	Null	Str1	Null	Null	Str2
Vehicle routing problems	Str3	Str4	Null	Null	Null	Null
Frigh trains problems	Null	Null	Str5	Null	Null	null

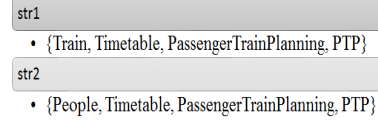


Fig 5. Example of CPDB

We have RCP1 and RCP2 for Train and RCP3 for People. We can identify SCPG that contains RCP1 and RCP3, and leave out RCP2, therefore compare the resulted SCPG with CPDB. Figure 5, shows the CPDB of PTP, we explains the dependencies on SCPGs, found above.

B. Identifying the type of Logistics Problems

After finding the general type of logistics problem, we require logistic expert to specify the attributes of the general problem, in order to further identify its exact type. This specification can lead us to find the exact problem type on the basis of logistic ontologies. In our case the general type problem is PTP, it includes many different types such that RSP, CSP, and PTP. These problem types are different. As explained in section V, we know that:

- RSP has two objective functions that are minimum travel cost and maximum number of passengers,
- CSP has one objective function that is minimum number of staff,
- PTP has one objective function that is minimum wait time for passenger,
- The objective function of LTPP is the minimum travel cost.

The logistic expert associates objective functions for his/her problem; hence, it executes a SPARQL query to identify the particular type of logistics problem.

For example, to extract the RSP problem, the logistic expert interrogates the reasoning system, with the selection of objective functions of the problem, such as min travel cost, max number of passengers, min wait time of passenger, min number of staff, etc. As shown in the Figure 6, the SPARQL query to identify the RSP problem on the selection criteria of logistic expert.

```

SPARQL query:
PREFIX r: <http://www.semanticweb.org/hendi/ontologies/2014/4/untitled-ontology-58#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?problem ?process ?activity
WHERE(
  ?process r:hasTime ?activity.
  ?activity r:ManageTrain ?vehicle;
  r:ManagePeople ?warehouse.
  ?process r:produces ?problem;
  r:hasFunction r:b1;
  r:hasFunction r:b4.
)

```

problem	process	activity
RSP	PTL	TT

Fig 6. SPARQL query to finding Type of PTP problem

C. Finding a List of optimization methods to solve the problem

Our Reasoning System maps the type of logistics problem to the corresponding types of optimization problems. It further assists to find the list of optimization methods that solve the logistics problem. It can be extracted by means of a SPARQL query, as shown Fig 7.

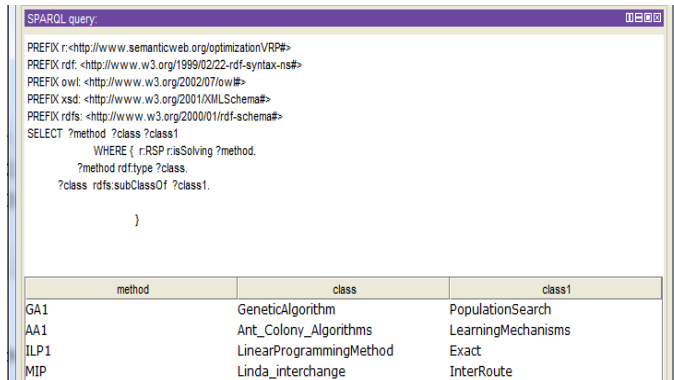


Fig 7. The SPARQL query that found list methods.

D. Find a list of articles that use the selected optimization method

When the logistic expert selects an optimization method, the Logistic Reasoning System proposes a list of articles comprising the name of article, author, and other details. The Fig 8 shows an excerpt of the Article database. The proposed articles deal with a particular logistics problem and selected methods to solve the problem. The ADB also contains URLs of web pages of the articles.

Name of article	author	URL	Name of problem	Type of method
Ant colony optimization approach to solving Rolling Stock Planning for passenger trains	Tsuij	URL1	RSP	ant colony
Fast and efficient adaptive large neighborhood search heuristic for Passenger Train Timetabling Problem	Eva	URL2	PTTP	Neighborhood heuristics

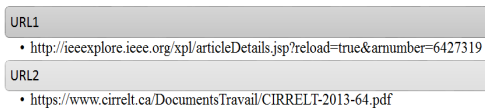


Fig 8. Example of Article database (ADB)

IX. CONCLUSION

We develop a logistic reasoning system to assist logistic experts to identify the logistics problem and also to provide associated optimization methods. The reasoning is based on the use of the conceptual modeling of logistics and optimization domains. This is achieved by the use of ontologies, querying languages, and tools to better specify the logistic semantics. We especially deal with the Passenger Transportation Problems. So, we describe and formalize the general concepts concerning this particular problem type and its variants. We also show, through a kind of scenario the use of Resource Concepts and Resource Concepts Paths used to identify a logistics problem, which allows the identification of related optimization methods and related bibliography

This work has been extended to propose web services constituting the resolution of the initial problem. We have been further defining some quality criteria in order to achieve the most convenient and useful web service orchestration for each logistics expert request.

ACKNOWLEDGMENT

The research for this work has been supported, in part, by the CAMPUS France, Ministry of High Education and Scientific Research, Iraq, and Le Laboratoire d'Informatique Signal et Image de la Côte d'Opale (LISIC), Calais, France

REFERENCES

- [1] A. Rushton, P. Croucher and P. Baker, The handbook of logistics and distribution management .
- [2] D. Huisman, L. Kroon, R. Lentink, M. Vromans, D. Huisman, L. Kroon, R. Lentink and M. Vromans, "Operations Research in passenger railway transportation", Statistica Neerland, vol. 59, no. 4, pp. 467-497, 2005.
- [3] E. Barrena, A fast and efficient adaptive large neighborhood search heuristic for the passenger train timetabling problem and dynamic demand. Montréal: CIRRELT, 2013.
- [4] Y. Tsuji, M. Kuroda, Y. Kitagawa and Y. Imoto, "Ant Colony Optimization approach for solving rolling stock planning for passenger trains", System Integration (SI), 2012 IEEE/SICE International Symposium on, 2012.
- [5] R. Pandey and D. Dwivedi, "Ontology Description using OWL to Support Semantic Web Applications", International Journal of Computer Applications, vol. 14, no. 4, pp. 30-33, 2011.
- [6] E. Sirin and B. Parsia, "SPARQL-DL: SPARQL Query for OWL-DL.", OWLED, 2007.
- [7] J. Miller, G. Baramidze and A. Sheth, "Investigating ontologies for simulation modeling", Simulation Symposium, 2004. Proceedings. 37th Annual, 2004.
- [8] J. Leukel and S. Kirn, "A supply chain management approach to logistics ontologies in information systems", Business Information Systems, 2008.
- [9] Y. Kayikci and H. Zsifkovits, "Successful ICT Integration in Transport Collaboration", Pioneering Solutions in Supply Chain Performance Management: Concepts, Technologies and Applications, p. 237, 2013.
- [10] J. Hoxha, A. Scheuermann and S. Bloehdorn, "An approach to formal and semantic representation of logistics services", Proceedings of the Workshop on Artificial Intelligence and Logistics (AILog), 19th European Conference on Artificial Intelligence (ECAI 2010), Lisbon, Portugal, 2010.
- [11] J. Han, J. Miller and G. Silver, "SoPT: ontology for simulation optimization for scientific experiments", Proceedings of the Winter Simulation Conference, 2011.
- [12] P. Witherell, S. Krishnamurty and I. Grosse, "Ontologies for Supporting Engineering Design Optimization", Journal of Computing and Information Science in Engineering, vol. 7, no. 2, p. 141, 2007.
- [13] V. Moussas, J. Tsahalis and H. Tsahalis, "Design of an ontology for Simulation Workflow Optimization", 5th International Conference on "Experiments / Process / System Modelling / Simulation / Optimization" (5th IC-EpsMsO), At Athens, Greece, 2013.
- [14] A. Caprara, L. Kroon and P. Toth, "Optimization problems in passenger railway systems", Wiley Encyclopedia of Operations Research and Management Science, 2011.
- [15] Z. Bahramian and M. Bagheri, "A simulation-based optimization approach for passenger train timetabling with periodic track maintenance and stops for praying", Journal of Modern Transportation, vol. 23, no. 2, pp. 148-157, 2015.
- [16] R. Borndörfer, M. Reuther, T. Schlechte, K. Waas and S. Weider, "Integrated Optimization of Rolling Stock Rotations for Intercity Railways", Transportation Science, 2015.
- [17] J. Haahr, J. Wagenaar, L. Veelenturf and L. Kroon, "A Comparison of Two Exact Methods for Passenger Railway Rolling Stock (Re)Scheduling", SSRN Electronic Journal.
- [18] L. Cavique, C. Rego and I. Themido, "Subgraph ejection chains and tabu search for the crew scheduling problem", J Oper Res Soc, vol. 50, no. 6, pp. 608-616, 1999.
- [19] Econometric Institute, Erasmus University Rotterdam, "Solving large scale crew scheduling problems by using iterative partitioning", 2008.
- [20] H. Hendi, A. Ahemd, M. Bouneffa and C. Fonlupt, "Logistics optimization using ontologies", The 4th international Conference on Complex Systems and Applications, Normandy University - Le Havre, France June 23 – 26, 2013.
- [21] F. Baader, I. Horrocks and U. Sattler, "Description logics as ontology languages for the semantic web", in Mechanizing Mathematical Reasoning, 1st ed., Springer, 2016, pp. 228-248.