Ontological Representation for Railway Domain

CS4900 : UGRC Report

April 21, 2023

Sasubilli Yuvan
Faculty Mentor : Prof. Narayanaswamy N S
Indian Institute of Technology Madras

Abstract
This part of the Railway Simulation project is to formulate a Network Database for static, and reasoning queries the user can ask The results are interested in helping the railway delay prediction simulation. The goal would be to develop a standardized terminology for the domain using Ontologies and other knowledge representation techniques. Once the terminology is standardized, the constraints for simulations and predictions can be considered as queries using keywords from the standardized terminology - in essence; it can be domain-specific knowledge representation language with support for a wide variety of logical queries. The user can give a query and obtain many attributes regarding the train and train tracks which are very useful for the simulation. The results are used for the prediction of delays. Still, to take the input query, there should be a Database containing the data and an ontology to restrict the query formulation.

Contents

1 Introduction

We aimed to design an ontology to standardize the terminology in the domain and relationships among the various entities. The static data is quite extensive and has different kinds. Some of them have been outlined above. The rest are as follows: Signal positions on different tracks, signaling protocols, engine haulage information, domain knowledge representation, and simulation data for automatic decision-making. By logical reasoning, the given Knowledge Representation, we should be able to an-

swer queries the user gives. We will use MIT-based software to write the logical reasoning on the simulation outputs obtained from the discrete event simulator running in parallel with this project. We read many papers and schemas which are already existing to come up with standardized terminology. We had gone through the internet to figure out the types of tracks and types of platforms in a railway station and represented all the constant attributes in the diagram. RailML framework is one of the most used frameworks for representing train-related data We took some of the concepts and subconcepts for our domain as required.

1.1 Why This Research?

This part of the Railway Simulation project is to form a Network Database for static, reasoning, and dynamic queries. The goal would be to develop a standardized terminology for the domain using Ontologies and other knowledge representation techniques. The user can give a query and obtain many attributes regarding the train, train tracks, and signaling segments We are giving the user access to the knowledge of the railway domain by creating this knowledge base.

2 Papers Review and Related Works

OSI Paper and Pascal Hitzler Paper

2.1 Ontology for complex railway systems

European Train Control System: The signaling-system component that includes control movement authorities, automatic train protection, and interface with the interlocking.
Roughly speaking, in the past, the challenge was to define a method to derive machine code from documentation (this documentation coming from the informal and sometimes implicit" knowledge" of the system to be developed). To answer this challenge, (countless) modeling methods were defined and are now available.
we will Experiment using ontological technologies (conceptualization, formalization, reasoning) to tackle a real and complex system.
The **ERTMS (European Rail Traffic Management)** ontology aims at modeling and formalizing the System Requirements Specification documents of the ERTMS. These documents are written in natural language. The aim of this ontology is the formalization of these specifications to obtain a data structure that can be reusable in the framework of other research in the ERTMS field.
The knowledge of a domain is formalized using several notations to regroup and create a formal structure of the concepts of this domain into a web of knowledge.
The so-called "rules" are created for the purpose of modeling requirements and certain "behaviors." of the system. In the railway domain, documents describing the System Requirements Specifications were issued to explain and clarify the usage of a part of the terms/concepts used in this domain and of the system itself.
This is an ontology created as a semantic model and module extracted from the below-mentioned documents. The extraction is based on the study, the comprehension of these documents, and the transposition of the information conceptualized in the same documents.
The railway domain is an environment with numerous heterogeneous information sources. Ontologies provide several useful features for intelligent systems, as well as for knowledge representation generally. The ERTMS ontology that we propose also aims at offering a solution for information exchange, and this is for a better railway transportation world.

2.2 Pascal Hitzler Paper

Knowledge-based systems have a computational model of some domain of interest in which symbols are surrogates for real-world domain artifacts, such as physical objects, events, relationships, etc. The domain of interest can cover any part of the real world or any hypothetical system about which one desires to represent knowledge for computational purposes.
Forms of Knowledge Representation in computer systems comprise semantic networks, rules, and logic. While semantic networks use the graph metaphor to visualize conceptual structures, rules exhibit some if-then-reading to express statements about a domain. Logic is used to implement precise formal semantics for semantic networks and rules.
Semantic networks are especially suitable for capturing concept hierarchies' taxonomic structure and expressing general statements about the domain of interest. Another natural form of expressing knowledge in some domain of interest is rules that reflect the notion of consequence.
Both forms, semantic networks, and rules have been formalized using logic to give them precise semantics. They are vague, ambiguous, and thus problematic for computational purposes without such precise formalization. The most prominent and fundamental logical formalism classically used for knowledge representation is the "first-order predicate calculus," or first-order logic for short, and we choose this formalism to present logic as a form of knowledge representation here.
A knowledge base entails all the statements that have been added via the tell-operation plus those that are their logical consequences. As an example, consider the following knowledge base with sentences in first-order logic.
The importance of a knowledge-based system is neatly described in the paper and hence we choose one for representing the railway domain knowledge in an ontology

3 Standardizing Terminology for Railway Domain

Terms taken from RailML framework

Currently in the railML framework **Railway Markup Language**, two subschemas are available for productive use: Infrastructure (IS) and Interlocking (IL). The Timetable and Rostering (TT) and Rollingstock (RS) subschemas are in active development and can be obtained for development and test purposes. Elements that do not fit into this structure are subsumed in the class Common (CO).

3.1 Interlocking (IL)

The interlocking subschema will focus on information that infrastructure managers and the signal manufacturing industry typically maintain in signal plans and route locking tables:

Data transfer: a standard data exchange format will allow the automation of data transfer, which is the process of adapting a railway interlocking and signaling system to a specific yard. Simulation programs: the railML® IL schema allows modelers to quickly absorb information about the interlocking systems such as timing behavior and routes and analyze the impact on railway capacity.

3.2 Infrastructure (IS)

The railML® Infrastructure subschema is focused on the description of the railway network infrastructure including all its various facets that are needed by the data exchange applications. In particular, the railML® infrastructure schema contains the following information:
Topology: The track network is described as a topological node edge model, partly explained

by the Simple example.
Coordinates: All railway infrastructure elements can be located in an arbitrary 2- or 3-dimensional coordinate system, e.g., the WGS84 that is widely used by today's navigation software.
Geometry: The track geometry can be described in radius and gradient.
Railway infrastructure elements: They enclose a variety of railway-relevant assets that can be found on, under, over, or next to the railway track, e.g., balises, platform edges, and level crossings.
Further located elements encompass elements that are closely linked with the railway infrastructure but that "cannot be touched", e.g. speed profiles and track conditions.

3.3 Rollingstock (RS)

The rollingstock subschema is part of the complete railML® schema providing a data structure in XML language to exchange railway-specific data. The rollingstock schema provides a container for all data about any rail vehicle, including locomotives, multiple units, and passenger and freight wagons. The second part of the schema enables the combination of single vehicles to formations as a fixed composition within a train or an entire train. It is intended to use this data schema for vehicle management as well as for detailed run-time calculations. Some of the concepts for the creation of Railway ontology are directly taken from the RailML rollingstock.

4 Knowledge Captured by the Ontology TBox

The statements present in the Railway Domain

4.1 Plain Description of Domain

- A train has exactly one train Number or train ID.
- Every train has a train Name.
- Every train has a Type and at least one Engine.
- Only a train has a train Schedule.
- Every train has an Average Time travel.
- A train may or may not have a current location (because it is dynamic).
- A train has one origin station and one destination station.
- A train has a dynamic delay.
- A train has at least one train driver.
- A train traveling may or may not have the nearest junction, which is a station.
- Every station has a unique station code or ID.
- Each station has one or more platforms.
- Every station has a data property named the number of platforms.
- Each station belongs to a particular railway zone.
- Every platform has a status.
- Status can be closed or open
- Every station has an address(optional).
- Every track has a unique track ID.
- Every track has a narrow, broad, standard, or meter type.
- Every track has a data property of maximum speed limit.
- Every track may or may not have an incident.

- Every track may or may not have track geometry.
- Every track is divided into one or more signaling segments.
- A unique segment ID identifies each signaling segment.
- Each segment has a start and end Milepost.
- Each segment has a status that is in condition or repair (optional).
- Each segment has a signaling protocol.
- Each segment has a signal status.
- Signal status can be red, amber, or green.
- Each segment has a current location: latitude and longitude (optional).
- Each segment has a segment speed limit, a data property.
- Each segment has a segment length, a data property.
- Each segment refers to a track id to which it belongs.

4.2 DL Ontology TBox

$Train \equiv (= 1\ hasName.TrainName)$ $T.1$
$\exists\ hasSchedule.\top \sqsubseteq Train$ $T.2$
$\exists\ hasDriver.Person \sqsubseteq Train$ $T.3$
$Train \equiv (= 1\ hasType.TrainType)$ $T.4$
$Train \sqsubseteq \forall hasAverageTime.TravelTime$ $T.5$
$Train \sqsubseteq \forall hasNearestJunction.Station$ $T.6$
$Train \sqsubseteq \exists hasLocation.Location$ $T.7$
$\forall\ hasRunningTrack.Track \sqsubseteq \exists Train$ $T.8$
$Train \equiv (= 1\ hasOrigin.Station)$ $T.9$
$Train \equiv (= 1\ hasDestination.Station)$ $T.10$
$Train \equiv (= 1\ hasDelay.Delay)$ $T.11$

$\exists\ hasName.StationName \sqsubseteq Station$ $T.12$
$Station \sqsubseteq \exists hasAddress.Address$ $T.13$
$Station \sqsubseteq \exists hasPlatforms.Platform$ $T.14$
$Platform \equiv \exists hasNumber.Number$ $T.15$
$Platform \sqsubseteq \exists hasStatus.Status$ $T.16$
$(Open \sqcup Closed) \sqsubseteq Status$ $T.17$

$Track \equiv (= 1\ hasType.TrackType)$ $T.18$
$(Narrow \sqcup Broad \sqcup Broad \sqcup Broad) \sqsubseteq TrackType$ $T.19$
$Track \sqsubseteq \exists\ hasStatus.Status$ $T.20$
$Track \sqsubseteq \exists\ hasSegment.Segment \sqcap \forall hasSegment.Segment$ $T.21$
$Track \sqsubseteq \exists hasIncident.Incident$ $T.22$
$Track \sqsubseteq \exists hasGemoetry.Geometry$ $T.23$

$Segment \equiv (= 1\ hasProtocol.Protocol)$ $T.24$
$Segment \sqsubseteq \exists hasStatus.Status$ $T.25$
$Segment \sqsubseteq \exists hasSignalStatus.SignalStatus$ $T.26$
$(Red \sqcup Amber \sqcup Green) \sqsubseteq SignalStatus$ $T.27$
$Segment \sqsubseteq \exists hasTrack.Track$ $T.28$
$Segment \sqsubseteq \exists hasLocation.Location$ $T.29$
$hasSegment \equiv hasTrack^{-1}$ $T.30$

5 Design Choices of Railway Ontology

The concepts, roles and their explanations

5.1 Concepts and Roles

The concepts used in the ontology are :

Train	TrainType	TrainName
Station	TrackType	StationName
Location	Address	Platform
Open	Closed	Narrow
Standard	Segment	Incident
TrainTime	Protocol	Delay
Broad	Meter	Geometry
Red	Amber	Green
Person	Track	Number
Status	SignalStatus	

The roles used in the ontology are:

hasName	hasSchedule
hasAverageTime	hasType
hasTrack	hasProtocol
hasOrigin	hasPlatforms
hasDelay	hasDestination
hasStatus	hasSegment
hasRunningTrack	hasGeometry
hasLocation	hasDriver
hasNumber	hasNearestJunction
hasAddress	hasIncident

Of the above, the primitive concepts are Address, Delay, Name, TrainType, Location, TrainName, StationName, Number, Open, Closed, Narrow, Broad, Meter, Standard, Incident, Geometry, Red, Amber, and Green.

5.2 Explanation for design choices

The train and station concepts will have their respective ids or codes as elements because their ids can uniquely identify them, but their names need not be unique. The train contains train type and train engines. Every train contains one or more drivers, so we defined driver relations accordingly. TrainDriver concept contains the name of the employee. The train engine has the specification of the engine used. The train has object properties, the origin and destination stations, with the station codes as the value. Only train elements have a train schedule indicated by GCI (T.2). The Delay is a primitive concept that represents the delay of the train running in hours.
The station concept has a data property station code, consisting of an object property with a Name and the Station name. The station concept has an address indicated by the address property Each station has one or more platforms; every platform is indicated by the platform number represented by the GCI . Every platform has a status that says it is open or closed. Every station has an address, and a station having a name is optional since we may address the station just with the address sometimes. Hence we do not keep the station name as mandatory for the station. Every station concept has a (number of platforms) as data properties along with the station code.
The track does not have a name. Every Track is uniquely identified by the ID, which is the data property of the Track concept. Every Track has one Track type; we assumed for main types are Narrow, Broad, Standard, and Meter out of all the possible types and defined all four types as individual primitive concepts. Every track is made up of only segments, and it consists of at least one segment. Every track has a maximum speed limit, which can be defined as a data property, but it is not a key property in the database, so we could have null values, which is why we defined it as an object property. The incident concept decides the status concept. The Track may or may not have a Geometry concept.
The segment is Signaling Segment, in short. It has the id as the data property and no name. It belongs to a track. Hence hasTrack is inverse relation of hasSegment. Every Segment has a signaling protocol which is represented as Protocol in short, and Protocol is a primitive class that has the different kinds of protocols followed by the Indian railways as instances. Every segment has an open or closed status and a signaling status that plays a crucial role in the protocol. The signal status class contains three primitive classes Red, Amber, and Green. Every segment has a maximum speed limit indicated as the data property of the segment, along with the segment id.

6 Theoretical Railway Queries

This section contains both static and dynamic queries

6.1 Database queries

- Provide all the trains which have originStation = A and destStaion = B and start in the morning before t(time, e.g., t = 10 am)
- Please provide all the trains which stop at exactly n(number) of stops between their originStation and the nearestJunction(other than the originStation) in its path.
- Provide the number of loops present in the track on which train A travels from the originStation to the destinationStation
- Is there any incident reported on the track that train A running on that track B might delay
- What trains cross the signallingSegment S in 15 minutes (both + and -)? Train A travels it.
- Provide the number of signaling segments between station A to station B in the path of train C.
- Provide the train numbers of the trains which reach station A within 15 minutes after the departure of train B from platform C.
- Provide all the trains with a trainType A running in the morning (before noon) passing through Junction B.
- Provide me station A's platform(s) where train B stops / can stop(which means open platforms) when it delays more than 10 minutes.
- What trains are delayed by more than 30 min and running at a speed whose difference with the segmentSpeedLimit is more than 30kmph
- Provide the status of track ID TID if there is an incident on track ID TID?
- Provide the number of platforms of station X.
- Provide the status of signallingSegment of segment ID SID ?

6.2 Reasoning Queries

- Which signallingSegments have the most frequent train movements?
- What is the maximum number of trains on a specific track simultaneously?
- Which stations are directly connected by a specific track?
- What is the expected travel time for a specific train between two stations, considering the speed limits and length of tracks, and the current occupancy status of the train?
- Which stations can be reached by a specific train within a given time limit, considering the speed limits of tracks and signaling Segments on the train's route?
- Which signalingSegments and tracks should be closed for maintenance to minimize the impact on train schedules?
- What is the expected delay time for a specific train passing through a signalingSegment that is currently experiencing congestion, considering the train's current speed and the signaling segment's congestion level?
- Which signaling segments and tracks should be prioritized for repair or upgrade to improve train speed and reduce delays?
- Which trains will be affected if a signalingSegment's signal changes from green to yellow?
- What is the expected delay time for a specific train passing through a signaling

segment showing a red signal, considering the train's current speed and the signalingSegment's congestion level?

- What is the expected arrival time of a specific train at a station if it is passing through a signaling segment showing a yellow signal?
- Which signaling segments have shown the highest proportion of red signals in the past week?
- What is the expected travel time for a specific train between two stations, considering the speed limits and length of tracks and the signal colors of the signalingSegments on the train's route?
- Which trains pass through signalingSegments showing a green signal, and which stations will they reach next?
- Which signalingSegments are showing a yellow signal, and what is the expected time until they change to green or red?
- Which trains will be delayed if a signaling segment's signal changes from yellow to red?
- What is the expected occupancy rate of a specific train passing through a signalingSegment that is currently showing a red signal?
- Which signalingSegments are currently experiencing congestion, and what are their signals' status (can be upgraded for dynamic data)?
- What is the expected arrival time of train number N at destination station X if it departs from origin station Y at time Z?
- What is the expected delay for train number N if there is an incident on track ID TID?
- What is the expected travel time for train number N between stations A and B?
- What is the expected condition of track ID TID?
- What is the expected incident status of track ID TID?
- What is the expected signal status of signallingSegment ID SID?

7 Conclusions

The conclusion from the project is that we tried experimenting with the field of Ontologies and Knowledge Representation for Our Railway Simulation project. We read and reviewed a few research papers on the protocols followed by the Indian railways; Some papers were based on Real Life Ontology in Railway Domain but not precisely in the area in which we are interested in working but somewhat similar to that (The HAL Open Science Paper on European Rail Traffic Management). We also read the documentation of RailML (Railway Markup Language Framework), derived and standardized the terminology, and added terms required for the query formulation, which helps the delay prediction. We made an Entity Relationship diagram for a brief idea about the main concepts, the subconcepts, and data properties (attributes). Then we wrote the description of the design choices of ontology and the same thing in description logic. We wrote several queries which the user might be interested in and which, after forming the database, will help the prediction simulation. The data population in an ontology is complex, and the ontology reasoner runtime differs from the polynomial runtime. So this ontology is best used only for small-scale data rather than for our project. We could have done the same thing in Prolog also, so one of the conclusions from the project is that this theoretical analysis helped us to standardize the terminology, write queries in which the user might be interested, and create a small ontology in Protege

but this is not of much practical use since ontology cannot handle big data such as Indian railway data.

---

8 Acknowledgements

I am grateful to my guide, Prof. N.S. Narayanaswamy, for all his time and advice and for always being accessible to assist me. I would also like to express my gratitude to the Department of Computer Science and Engineering at IIT Madras for enabling undergrads like me to participate in research through the UGRC program. Many thanks to my research companions, Aditya Viraj and Hari Charan, for constantly supporting me and always being there when I need help. Thanks to Prof.P Sreenivasa Kumar and Ph.D. scholar Sahil for helping me learn Ontologies and Protege.

---

9 References

---

1. Ontology for complex railway systems application to ERTMS/ETCS system
   https://hal.science/hal-00912733/document
2. http://protegeproject.github.io/protege/getting-started/
3. ER diagram
   https://l1nk.dev/K1Kb6
4. Capturing simulation intent in an ontology: CAD and CAE integration application
   https://encr.pw/0Ikz2
5. Pascal Hitzler Webpage
   https://people.cs.ksu.edu/~hitzler/