Modelling Business Agreements in the Multimodal Transportation Domain Through Ontological Smart Contracts

Mario Scrocca [1], Marco Comerio, Alessio Carenini, and Irene Celino

Cefriel – Politecnico di Milano
Viale Sarca 226, 20126 Milano, Italy
e-mail: name.surname@cefriel.com

Abstract. The blockchain technology provides integrity and reliability of the information, thus offering a suitable solution to guarantee trustability in a multi-stakeholder scenario that involves actors defining business agreements. The Ride2Rail project investigated the use of the blockchain to record as smart contracts the agreements between different stakeholders defined in a multimodal transportation domain. Modelling an ontology to represent the smart contracts enables the possibility of having a machine-readable and interoperable representation of the agreements. On one hand, the underlying blockchain ensures trust in the execution of the contracts, on the other hand, their ontological representation facilitates the retrieval of information within the ecosystem. The paper describes the development of the Ride2Rail Ontology for Agreements to showcase how the concept of an ontological smart contract, defined in the OASIS ontology, can be applied to a specific domain. The usage of the designed ontology is discussed by describing the modelling as ontological smart contracts of business agreements defined in a ride-sharing scenario.

Keywords. business agreement, ontological smart contracts, multimodal transportation

1. Introduction

In the context of multimodal transportation, a wide set of stakeholders should cooperate to provide passengers with a seamless travel experience. The Shift2Rail Innovation Programme 4[2] (IP4) investigated the design and implementation of an ecosystem of transportation stakeholders relying on an *Interoperability Framework* to support the communication among them and the definition of new services for the users. In this context, the Ride2Rail project[3], focused its attention on the integration of ride-sharing alternatives.

The introduction of new transportation modes, such as ride-sharing, poses additional challenges related to a business environment blending companies and private actors. For this reason, Ride2Rail investigated the usage of blockchain technology to offer suitable

---

[1] Corresponding Author. E-mail: mario.scrocca@cefriel.com

[2] Shift2Rail IP4, https://rail-research.europa.eu/research-development/ip4/

[3] Ride2Rail, https://ride2rail.eu/

guarantees for trust in a multi-stakeholder scenario involving the definition of business agreements. Ride2Rail developed a specific software module for the definition of business agreements as smart contracts, i.e, executable software applications that implement self-executing logic in a blockchain. A smart contract could be implemented as a tool to automate the execution of an agreement that will run when certain conditions are met.

To foster the semantic interoperability of the developed module within the IP4 ecosystem, we designed and adopted an ontology to model the defined agreements through a shared semantic and decoupling the specification of the agreements from its technological implementation. This paper describes the development of such ontology and its application within the Ride2Rail project to describe business agreements in the considered ride-sharing scenario. The main contributions of the paper are: (i) analysis of the literature surveying approaches for an ontological representation of smart contracts, (ii) design and development of an ontology for the definition of business agreements in a multimodal transportation scenario, and (iii) validation of the ontology and exemplification of its usage.

The developed *Ride2Rail Ontology for Agreements* leverages the concept of *Ontological Smart Contract* defined in the OASIS ontology [1] and investigates how it can be extended to model the semantic of business agreements. Although we focused our work on the transportation domain, the proposed approach can be generalised to model ontological smart contracts in different domains.

The remainder of the paper is structured as follows: Section 2 discusses preliminaries regarding the context, terminology and methodology followed; Section 3 frames the work considering the state-of-the-art; Section 4 describes the ontology engineering process and the implemented ontology; Section 5 exemplifies the usage of the ontology reporting how it is used to describe the agreements defined within the Ride2Rail project; Section 6 draws the conclusions, discusses how the proposed approach can be generalised and future work.

2. Preliminaries

This section presents a preliminary introduction to the context and terminology analysed in the paper and the methodology adopted.

2.1. Context

The Ride2Rail project investigated the specific requirements of ride-sharing for its integration in the multimodal transport ecosystem defined in IP4. The idea behind Ride2Rail is to consider each driver (i.e., a user offering a shared ride with her/his car) as a private transport service provider (TSP) offering transportation services on a specific route. The challenges of such an integration are multiple, for example, the need for a user application that could allow passengers to become a driver offering a ride with their own vehicle, or the need for a dynamic update of shared rides published by drivers for the multimodal journey planning. Moreover, a major challenge is related to how to guarantee trust in an environment where private actors can offer, along with companies, a paid transportation service to passengers. For this reason, the *Agreement Ledger Module* was designed and developed in Ride2Rail exploiting the blockchain technology to guarantee trust in the

definition of agreements between parties (TSPs, drivers and passengers). The *Agreement Ledger Module* is a software module exposing through an API a set of functionalities relying on smart contracts deployed on a distributed ledger for the digital representation and execution of business agreements. The overall design and implementation of the *Agreement Ledger Module* is documented in the project deliverable [2].

The objective of the ontology engineering activity described in this paper is not to transpose ontologically the content of the ledger, but to identify and provide a *semantic description* of the contracts implemented so that they can facilitate interoperability within IP4 and their comprehension from IP4 stakeholders.

For this purpose, the *Ride2Rail Ontology for Agreements* aims at identifying a set of classes and properties to represent the agreements between parties that could be implemented through the *Agreement Ledger Module*. In this paper, we demonstrate how the following agreements could be described using the ontology:

- The *Ridesharing Booking*, as an agreement between a driver and a passenger.
- The *Incentive*, as an agreement between different parties to grant, according to a set of conditions, a reward that could promote more sustainable transportation alternatives.

2.2. Terminology

In the IP4 scenario, a user with a mobility need can interrogate an application for passengers that is able to process a mobility request and return a set of multimodal offers to cover the itinerary between the required origin and destination. Each *offer* is associated with a *trip* and a set of *offer items* that the passenger can book to be entitled to travel according to the proposed trip. In the general case, a trip is composed of multiple *travel episode*s, i.e., multimodal trip legs, offered by different *travel expert* systems, provided by different operators and combined to generate a trip.

In this context, the following terminology is introduced for the considered scenario:

- a *lyft* is defined as a ride-sharing leg in the (multi-modal) trip of a passenger, therefore, it is a travel episode for ridesharing;
- a *ride* is the transportation service offered by a *driver* that enables a travel episode for one or more passengers;
- a *ridesharing booking* represents the booking made by a passenger of an offer item associated with a ride offered by a driver;
- a *Crowd-based Travel Service Provider* is a travel expert system handling offer items for the rides offered by a set of drivers;
- an *incentive* is an agreement between parties that is offered by an *incentive provider* defining the *incentive mechanism* and *incentive conditions* for granting it.

A more detailed discussion of the presented terminology can be found in the Ride2Rail deliverables [3,4,5].

2.3. Methodology

The methodology adopted for the definition of the ontology is based on *Linked Open Terms* (LOT) [6], a consolidated industrial method to develop ontologies and glossaries.

The LOT methodology is divided into four steps: ontology requirements specification, ontology implementation, ontology publication and ontology maintenance. In the following paragraphs, we briefly discuss the first three steps of the methodology performed to design and implement the *Ride2Rail Ontology for Agreements*.

The ontology requirements specification consists of the definition of the ontology requirements considering the purpose and scope of the ontology, domain analysis and an investigation of the existing data flows. The activity starts with the identification of a set of *use cases* and *user stories* for the ontology. A *use case* should answer the following questions: *Who will be the actors interested in querying the ontological data? What are the expected usages of data modelled through the ontology?*

Considering the *use cases* and *user stories* defined and the domain analysis performed, then the ontological requirements are specified in the form of *competency questions* and *facts*. The set of *competency questions* defines, in the form of hypothetical questions associated with a *user story*, the information that should be possible to retrieve from data modelled using the ontology. The set of *facts* describes the semantics and the requirements associated with the domain-specific terminology (e.g., attributes describing a specific term, etc.). In this phase, domain experts and stakeholders, are involved to ensure a comprehensive set of ontological requirements is specified.

The second step is the implementation of the ontology. Considering the requirements, a first conceptual model is produced with the required set of classes and properties. Then, in line with the best practices of ontology engineering, the relevant and already existing vocabularies are analysed to assess the possibility of reusing them.

Finally, the actual ontology is coded in the OWL language and it is validated with the support of automatic diagnosis tools and by manually assessing it with respect to the ontological requirements specified.

The third step is the documentation and publication of the ontological model.

3. Related work

Blockchain-based solutions and Semantic Web technologies are complementary and may benefit each other. Many researchers are investigating how to combine these technologies and also the European Commission is supporting this research area. As an example, we cite the H2020 ONTOCHAIN[4] project financed under the Next Generation Internet initiative.

Different approaches combining these two areas are reported in the literature. J. Cano-Benito et al. discuss in [7] six different scenarios: (a) blockchain with semantic meta-data, (b) blockchain with RDF content, (c) blockchain and virtual RDF service to publish its content, (d) blockchain with external pointers to RDF data, (e) blockchain referencing another blockchain through RDF, (f) semantic blockchain implemented relying on Semantic Web.

The implementation of the mentioned scenarios relies on the definition of ontologies to model the content of the blockchain. In Ride2Rail, we were interested in investigating how smart contracts implemented on the blockchain could be described using an ontology. This approach addresses two interoperability needs: (i) the description in

---

[4]https://ontochain.ngi.eu/

an implementation-independent way of the smart contracts defined according to a specific blockchain-based solution, and (ii) the adoption of proper and shared terminology to describe domain entities and their relationships. Moreover, even if out of scope for the Ride2Rail project, it can foster the implementation of virtual RDF services to query the content of the blockchain using the ontology (scenario *c.*).

In the following paragraphs, we analyse existing vocabularies that were considered to evaluate their re-use in the implementation of the ontology.

3.1. Smart Contracts and Ontologies

In the analysis of the literature, we found relevant related work regarding the definition of ontologies for smart contracts.

The paper *Ontologies for Commitment-Based Smart Contracts* [8] defines a platform-independent conceptualization of smart contracts, however, as mentioned also by the authors in their conclusions, it represents an initial model that should be refined and evaluated before being finally implemented and published as an ontology.

Kruijff and Weigand in *Understanding the Blockchain Using Enterprise Ontology* [9] adopt an ontology-based approach to formalise the terminology related to the blockchain, including smart contracts, but the proposed modelling considers a higher level of abstraction and doesn't allow for the detailed description of a specific smart contract. Moreover, the authors mention that the presented ontology is still an initial model to be validated and finalised.

Similarly, the *Ontology for Smart Contracts* proposed by McAdams [10] identifies the basic terminology for a conceptualisation of smart contracts that could aid in implementing formal reasoning over their behaviour. The proposed contribution is, however, not implemented as an ontology yet.

Finally, the paper *Ontological Smart Contracts in OASIS* [1] defines the concept of Ontological Smart Contract extending the OASIS ontology for agents, systems, and integration of services [11]. The proposed ontology defines the concept of a smart contract as an entity to define agreements between agents and specify their terms, independently from the specific blockchain implementation. Differently from the other work in the literature, the OASIS ontology is fully implemented in OWL, published online and its usage is documented and exemplified by the authors. Moreover, we contacted the authors that provided us with additional documentation to re-use the vocabulary and confirmed that a plan for the maintenance of the ontology is in place. For these reasons, we decided to adopt this vocabulary as a basis for our work.

In this paper, we discuss how the OASIS ontology can support the modelling of ontological smart contracts considering different blockchain technologies, and how it can be extended and leveraged to model business agreements in different domains. Indeed, the usage of the OASIS ontology is exemplified in [1] considering a trading agent selling stocks and the Ethereum platform, while, in Section 5, we consider agreements in the transportation domain implemented through the Hyperledger Fabric API[5].

---

[5]https://github.com/hyperledger/fabric-contract-api-go

3.2. IP4 Ontologies

To support the definition of domain-specific classes and properties, and to support interoperability within IP4, we analysed the current status of the ontologies for the multimodal transportation domain defined in the context of IP4.

The IP4 ontology is currently undergoing an in-depth process of modularization and extension [12] considering already standardized formats (e.g., Transmodel[6], OSDM[7], TRIAS[8], GTFS-RT[9] etc.). A preliminary release of the new modules of the IP4 ontology, currently under finalisation is available on Github[10]. The two already available modules of the IP4 ontologies address the *Transmodel* concepts (Core, Commons, Fares, Facilities and Journeys submodules) [13] and the *Open Sales and Distribution Model* (OSDM) specification to model the booking process.

4. Ontology for Agreements

The objective of the *Ride2Rail Ontology for Agreements* is to provide a conceptualization of the basic terms for the description of the business agreements defined in the multimodal transportation context discussed in Section 2.1. The ontology takes into account the terminology (Section 2.2) and the ontologies already defined within IP4 (Section 3.2) to support interoperability through shared semantics.

The following sections describe the application of the presented methodology (Section 2.3) for the design, implementation and publication of the *Ride2Rail Ontology for Agreements*.

4.1. Ontological Requirements Specification

This section describes the ontological requirements identified for the *Ride2Rail Ontology for Agreements*. The collection of requirements leveraged the analysis of the overall requirements defined for the Ride2Rail project and the specific ones identified for the implementation of the Agreement Ledger Module. Furthermore, additional stakeholders from the transportation domain were involved to take into account additional considerations from the project consortium and other IP4 actors.

Two use cases were identified to support the definition of the ontology.

UC1 – Dispute Resolution about Ridesharing
Description: in case of a dispute between a driver and a passenger regarding a booked ride, the responsible authority wants to access trusted data to resolve it.
Stakeholders: Driver, Passenger, Authority
Workflow: The responsible authority analyses the details of the booking agreement between the driver and the passenger obtaining trusted information that can help in solving the dispute.

---

[6]https://www.transmodel-cen.eu/
[7]https://unioninternationalcheminsdefer.github.io/OSDM/
[8]https://github.com/VDVde/TRIAS
[9]https://developers.google.com/transit/gtfs-realtime
[10]Working repository https://github.com/oeg-upm/mobility

UC2 – Incentives to promote Ridesharing

Description: travellers (both drivers and passengers) are given incentives to involve ride-sharing in their multimodal rides.

Stakeholders: Passenger, Driver, Travel Service Provider

Workflow: Incentives are represented as ontological smart contracts and can be queried to get information about the conditions and mechanisms of available incentives within IP4.

For each use case, different user stories considering the stakeholders involved were identified. Finally, a set of facts and competency questions was defined considering: (i) the use cases and user stories defined, (ii) the analysis of the business agreements modelled on the blockchain, and (iii) the relevant terminology in the considered domain.

Table 1 contains the competency questions identified for the two use cases, terms starting with a capital letter are concepts described in facts. The complete list of user stories, facts and competency questions is reported in the ontology repository[11].

Competency Question	Expected Result	Use case
What is the origin/destination of the Ridesharing Leg offered by the Driver and booked by the Passenger?	Origin/Destination of the Ridesharing Leg involved in the Ridesharing Booking	UC1
What is the price agreed upon between the Driver and the Passenger?	Price agreed for the Ridesharing Leg involved in the Ridesharing Booking	UC1
What is the number of seats declared by a Driver offering a Ride?	Number of seats associated with a Ride	UC1
What are the incentive agreements involving a TSP as an incentive provider?	Incentives involving a TSP in the agreement	UC2
What are the conditions defined for a given Incentive?	Conditions defined for the applicability of the Incentive	UC2
Is there a tangible good or benefit associated with a given Incentive?	Benefit associated with the Incentive	UC2

Table 1.: Competency Questions identified for the Ontological Requirements Specification

4.2. Ontology Implementation

To support the ontology implementation phase, we adopted the *Chowlk*[12] notation and converter [14] that allows building the conceptual model graphically and then to directly obtain a first serialization of the ontology in OWL.

The design of the conceptual model, starting from a glossary of terms extracted from the ontological requirements, went through several iterations considering also the outcomes of the review of already available ontological and non-ontological data formats.

To facilitate the description of the final conceptual model, we first discuss the reused vocabularies to model the ontological smart contracts and the domain terminology. Then,

---

[11]https://github.com/Ride2Rail/agreement-ledger-ontology/tree/main/requirements
[12]https://chowlk.linkeddata.es/

we present the final version of the ontology through the *Chowlk* notation. Using the diagram, we motivate our design decisions by describing the introduced classes and properties and the alignment with the re-used vocabularies.

4.2.1. Ontological Smart Contracts

The Ontology for Agents, Systems, and Integration of Services (OASIS) is published online[13] with the namespace `http://www.dmi.unict.it/oasis.owl#` (`oasis:` prefix).

An ontological smart contract in OASIS (*oasis:SmartContract*) is modelled defining the set of entries involved in the agreement (*oasis:SmartContractEntry*) and the set of conditionals (*oasis:ConditionalSet*) specifying the terms of the agreement. Agreement instances are modelled through the class *oasis:SmartContractInstance* and they are associated with a specific *oasis:SmartContract*.

A *oasis:SmartContractEntry* can be of class *oasis:SmartContractEntryParticipant*, describing a participant involved in the agreement, or class *oasis:SmartContractEntry Value*, describing values involved. Each *oasis:SmartContractEntry* can be described using the property *oasis:refersExactlyTo*, if it refers to a specific individual for each instance of the described agreement, or using the property *oasis:refersAsNewTo*, if it describes an individual through an *oasis:EntryTemplate*. An *oasis:EntryTemplate* allows to specify the features that an *oasis:SmartContractEntry* should have in an *oasis:SmartContractInstance* of the modelled *oasis:SmartContract*.

The terms of the agreements are modelled using *oasis:Conditional*, which represent an implication between an antecedent (*oasis:ConditionalBody*) and consequent (*oasis:ConditionalHead*). Whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold. Both an *oasis:ConditionalBody* and an *oasis:ConditionalHead* can specify multiple conditions modelling different *oasis:ConditionalAtom*. All the *oasis:ConditionalAtom* should be satisfied to satisfy the antecedent/consequent. A *oasis:ConditionalAtom* can be described through:

- *oasis:ConditionalSubject*: representing the subject;
- *oasis:ConditionalObject*: representing the object;
- *oasis:ConditionalOperator:* representing actions (*oasis:Action*) from subject(s) to object(s);
- *oasis:ConditionalParameter*: representing a parameter of the action described by the operator (the two subclasses *oasis:ConditionalInputParameter* and *oasis:ConditionalOutputParameter* representing an input and an output parameter, respectively);
- *oasis:ConditionalOperatorArgument*: representing operator arguments for a subordinate characteristic of the operator

Also in the modelling of conditionals, an *oasis:EntryTemplate* can be leveraged to specify the features that the entities involved in the conditional.

4.2.2. Domain Terminology

For the definition ontology, considering the elicited requirements, we focused mainly on the *OSDM* module of the IP4 ontologies. The namespace for the considered ontology is `https://w3id.org/mobility/`

---

[13]OASIS ontology `https://www.dmi.unict.it/santamaria/projects/oasis/oasis.php`

osdm/core# (osdm: module), but the publication process is not finalised yet. In particular, we reused classes and properties related to the concepts of:

- *osdm:Offer*: defined as "a response to a customer mobility request as a result of the travel shopping process, it is composed of offer item(s) for service(s) designed to cover each proposed journey, and, optionally, ancillary services".
- *osdm:Booking:* defined as "an operational process as part of the sales process to commit to a sales transaction binding the customer and supplier to the offer".

To complement the set of classes and properties already available in the IP4 ontologies, the MaaSive Glossary and the conceptualisation effort made in Ride2Rail WP2.1 (Deliverable D2.1 [3] and D2.4 [4]) and WP3.1 (Deliverable 3.1 [5]) were taken into account as non-ontological resources to improve the semantic interoperability of entities modelled in our ontology.

4.2.3. Ride2Rail Ontology for Agreements

The implementation of the *Ride2Rail Ontology for Agreements* is based on the decision to reuse the *oasis:SmartContract* as the class to model a business agreement. Our claim is that concept of *oasis:SmarContract* is modelled in the OASIS ontology using a generic approach that can be extended and applied to different domains once identified a suitable vocabulary for the representation of the domain terminology. An *oasis:SmarContract* not only enables the representation of the agreement and the entities involved a *oasis:SmartContractEntry*, but also a detailed model of the terms of the agreements as *oasis:Conditionals*. The main objective of the ontology is to complement the current IP4 ontologies and extend the OASIS ontology providing the needed terminology to model ontological smart contracts in the multimodal transportation domain.

Figure 1 adopts the Chowlk notation to describe the classes and properties modelled and their relations with reused vocabularies. The namespace selected for publication is https://w3id.org/ride2rail/terms# (r2r: prefix).
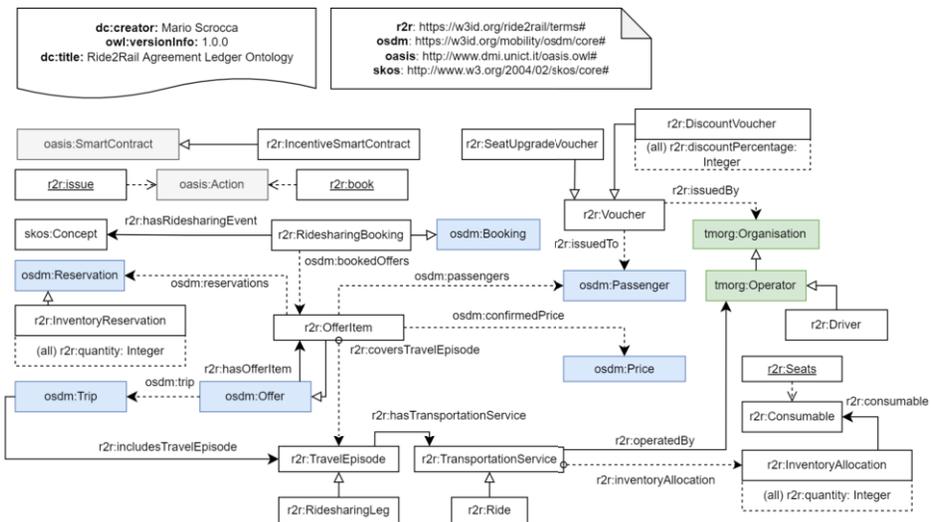

Figure 1. Chowlk diagram for the Ride2Rail Ontology for Agreements

To support the modelling of the *oasis:SmartContract* for a ridesharing booking, the class *r2r:RidesharingBooking* is defined as a subclass of *osdm:Booking.* An *osdm:Booking* is associated with the booked *osdm:Offer* and the *osdm:Price* paid for the offer. *An osdm:Offer* represents the pair between a computed *osdm:Trip* and the set of products (*osdm:OfferPart*) offered to an *osdm:Passenger* and required to perform the trip.

The ontology extends the concept of *osdm:Offer* defining an *r2r:OfferItem* to model the corresponding term in the IP4 glossary (an offer item is part of an offer, which is provided by a single TSP). As a result, *r2r:RidesharingBooking* is associated with an *r2r:OfferItem* provided by a ridesharing TSP, i.e. the Crowd-based TSP in the ridesharing scenario.

The ontology also defines two classes, *r2r:TravelEpisode* and *r2r:TransportationService* to model, respectively, the IP4 concepts of *travel episode* ("part of a trip operated with the same vehicle") and *transportation service* ("service that provides transportation on a travel episode"). An *osdm:Trip* may include multiple *r2r:TravelEpisode* (possibly aligned with the concept of *osdm:Segment*), an *r2r:TravelEpisode* may be operated by an *r2r:TransportationService*.

An *r2r:OfferItem* for an *r2r:RidesharingBooking*, is associated with an *r2r:RidesharingLeg* (subclass of *r2r:TravelEpisode*) that is made possible through an *r2r:Ride* (subclass of *r2r:TransportationService*) operated by an r2r:Driver. In this sense, *r2r:Driver* extends the concept of *tmorg:Operator*.

As specified in the requirements, an *r2r:Ride* may specify an *r2r:InventoryAllocation* that indicates a given quantity of available consumables (*r2r:Consumable*). The ontology defines *r2r:Seat* as an individual of the class *r2r:Consumable*. An *r2r:InventoryReservation* (subclass of *osdm:Reservation*) can be associated with an *r2r:OfferItem* defining the number of consumables reserved by the corresponding booking.

To support the modelling of incentives, the class *r2r:IncentiveSmartContract* is defined as subclass of *oasis:SmarContract*. The ontology also models the concept of *r2r:Voucher* to define a redeemable good issued by a *tmorg:Organisation*. Two subclasses are defined for *r2r:Voucher* to model the mechanisms of the incentives implemented in Ride2Rail: a *r2r:DiscountVoucher* allowing to model a certain percentage of discount granted to the beneficiary of the voucher, and a *r2r:SeatUpgradeVoucher* granting an upgrade of seat class for the beneficiary.

To express the incentive conditions defined in the requirements, the ontology also defines two *oasis:Action* individuals, *r2r:issue* and *r2r:book*, that can be used as *oasis:ConditionalOperator* in the modelling of conditionals for an *oasis:SmartContract*.

Finally, to model the events that can be associated with a ridesharing booking we decided to implement a SKOS[14] Concept Scheme.

As defined in the requirements, five events are identified in the first level of the taxonomy: *RidesharingStarted* for the start of the ride associated with the ridesharing booking, *RidesharingCompleted* for the completion of the ride associated with the ridesharing booking, *RidesharingCancelled* for the cancellation of the ridesharing booking by the passenger or by the driver, *RidesharingDelayed* for a delay in the ride due to the passenger or to the driver, *RidesharingNoShow* for a passenger or a driver not showing as expected for the booked ridesharing.

---

[14]https://www.w3.org/2004/02/skos/

The modelled ontology was validated against the ontological requirements and using OOPS! [15] as the state-of-the-art tool for automatic diagnosis of anomalies in the ontology[15].

4.3. Ontology Publication

The *Ride2Rail Ontology for Agreements* is published online following the best practices for ontology publication at `https://w3id.org/ride2rail/terms#` (`r2r:` prefix). We adopted the *w3id* service for permanent identifiers[16], and we implemented content negotiation to serve the ontology in different human-readable and machine-readable formats[17]. The Widoco [16] tool was used to generate the ontology documentation, then complemented through diagrams and the description of the main design decisions. The SKOS taxonomy for ridesharing booking events is published at `https://w3id.org/ride2rail/rb-events#` (`rbe:` prefix) using a similar approach.

The license adopted is the Creative Commons with Attribution right (CC-BY), which allows licensees to copy and distribute the work and make derivative works, giving the authors proper credits.

All the material related to the ontology and the artifacts produced during the ontology engineering process are hosted on Github under the Ride2Rail organisation in the repository `https://github.com/Ride2Rail/agreement-ledger-ontology`.

5. Modelling Business Agreements

In this section, we exemplify the usage of the defined *Ride2Rail Ontology for Agreements* discussing the RDF representation of the business agreements implemented through the Agreement Ledger Module in the Ride2Rail project, i.e., the ridesharing booking and the incentives. We decided to distinguish between the ontology and the agreements since the definition of different business agreements (e.g. different conditionals and incentives) is made possible by relying on the implemented ontology.

The RDF dataset is published at `https://w3id.org/ride2rail/agreements#` (`ag:` prefix) and hosted on Github[18]. A Chowlk diagram is provided in the ontology documentation for each *oasis:SmartContract*[19]. In the following, we discuss the defined agreements and the expected usage of the terms in the considered scenario.

The first agreement is the *ag:RidesharingBookingSmartContract* that defines the entities associated with a ridesharing booking and their relations. Figure 2 represents the Chowlk diagram describing the modelled *oasis:SmartContract*. An *ag:RidesharingBookingSmartContract* involves an *r2r:Driver* and an *osdm:Passenger* as participants in an *r2r:RidesharingBooking*. The agreement specifies that the *r2r:RidesharingBooking* is associated with an *r2r:OfferItem* for the passenger, that has a specific *osdm:Price*, is associated with an *r2r:RidesharingLeg*, and includes an *r2r:InventoryReservation*. The

---

[15]A report of the validation is available in the ontology repository `https://github.com/Ride2Rail/agreement-ledger-ontology`

[16]`https://w3id.org/`

[17]Recipe 3 from `https://www.w3.org/TR/swbp-vocab-pub`

[18]`https://github.com/Ride2Rail/agreement-ledger-ontology/tree/main/agreements`

[19]The Chowlk diagrams for the defined *r2r:IncentiveSmartContract*s are available also in the repository `https://github.com/Ride2Rail/agreement-ledger-ontology/tree/main/docs/diagrams`
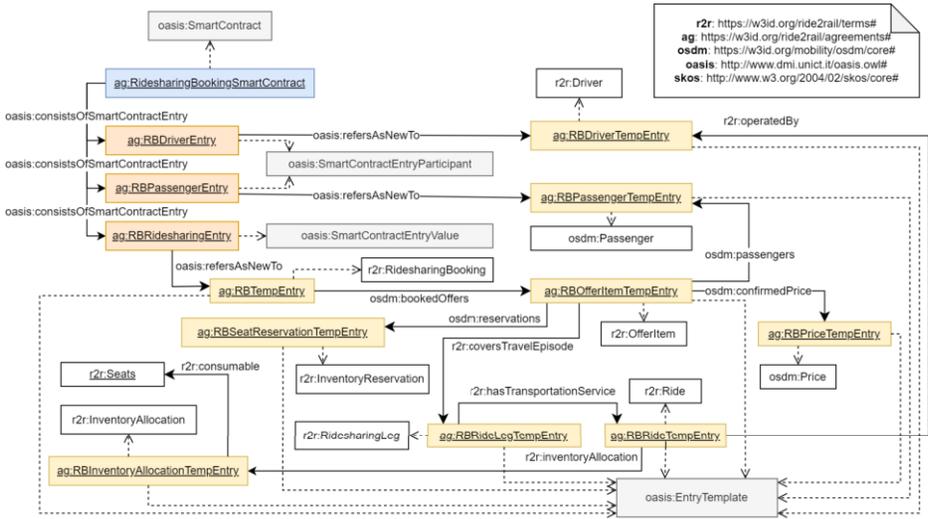
Figure 2. Diagram describing the *RidesharingBookingSmartContract* agreement.

r2r:RidesharingLeg has as transportation service an *r2r:Ride* offered by the driver with a specific *r2r:InventoryAllocation*.

The *ag:RidesharingBookingSmartContract* does not define any *oasis:Conditional* since the terms of the agreement are not directly modelled through the Agreement Ledger Module. Table 2 defines examples of *oasis:Conditional* that can be defined exploiting the ontology, in particular, leveraging the ride-sharing booking events taxonomy. Example 1 describes the fact that the money from the passenger should be transferred to the driver if the ride is correctly started and ended, examples 2 and 3 describe potential refund policies if the driver doesn't show up or cancels the ride.

Conditional		*example-1*	*example-2*	*example-3*
Entities	Participant	Passenger Driver	Passenger Driver	Passenger Driver
	Value	Ridesharing Booking	Ridesharing Booking	Ridesharing Booking
Body	Subject	Ridesharing Booking	Ridesharing Booking	Ridesharing Booking
	Operator	is associated with	is associated with	is associated with
	Object	Start Event, End Event	Cancelled Driver	No Show Driver
Head	Subject	Passenger	-	-
	Input Parameter	Price	-	-
	Operator	pay	refund	refund
	Output Parameter	-	Price	Price
	Object	Driver	Passenger	Passenger

Table 2.: Example of *oasis:Conditional* for a ride-sharing booking smart contract.

The Agreement Ledger Module implemented in Ride2Rail defines three agreements related to incentives, i.e., three *r2r:IncentiveSmartContract*: Ride with other passengers (*ag:RideWithOtherPassengersIncentive*), Multimodal discount (*ag: MultimodalDiscountIncentive*), Multimodal repetition discount (ag:*MultimodalDiscountIncentive3*). Table 3 summarises the defined agreements reporting the *oasis:SmartContractEntry* involved and the modelled *oasis:Conditional*. All the agreements are between an *osdm:Passenger* and a *tmorg:Operator,* then:

- *ag:RideWithOtherPassengersIncentive* defines a conditional specifying that if the passenger books a ride also booked by another passenger, then the TSP issues an *r2r:SeatUpgradeVoucher* to the passenger;
- *ag:MultimodalDiscountIncentive* defines a conditional specifying that if the passenger books a multimodal ride involving an *r2r:RidesharingLeg* and at least another leg, then the TSP issues an *r2r:DiscountVoucher* of 10% to the passenger;
- *ag:MultimodalDiscountIncentive3* defines a conditional specifying that if the passenger books a multimodal ride involving an r2r:RidesharingLeg and at least another leg for three times, then the TSP issues an r2r:DiscountVoucher of 20% to the passenger.

Incentive Smart Contract		*Ride With Other Passengers Incentive*	*Multimodal Discount Incentive*	*Multimodal Repetition Discount Incentive*
Entities	Participant	Passenger P1 Travel Service Provider TSP	Passenger P1 Travel Service Provider TSP	Passenger P1 Travel Service Provider TSP
Body	Subject	Passenger P1, Passenger P2	Passenger	Passenger
	Operator	makes a booking	makes a booking	makes a booking
	Operator Argument	-	-	3 times
	Object	Ride R1	Offer for a Trip involving a Ridesharing Leg and another Leg	Offer for a Trip involving a Ridesharing Leg and another Leg
Head	Subject	TSP	TSP	TSP
	Operator	issue	issue	issue
	Object	Seat Upgrade Voucher for Passenger P1	10% Discount Voucher for Passenger	20% Discount Voucher for Passenger

Table 3.: *IncentiveSmartContract*s implemented in Ride2Rail

The described *oasis:SmartContract*s not only enable the sharing of the modelled business agreements between the IP4 stakeholders, but also the representation of smart contract instances stored on the blockchain to support the use cases defined.

6. Conclusions

In an ecosystem comprehending various stakeholders, the implementation of business agreements through a distributed ledger provides several benefits regarding information trust and the automatic execution of the agreed terms modelled as smart contracts. This approach, however, doesn't provide any guarantee about the interoperability of the defined agreements from a technological and semantic perspective. On one hand, the domain terminology shared among the involved stakeholders should be referenced by the modelled entities, on the other hand, other software systems can benefit from a machine-readable representation of the agreements.

The concept of ontological smart contract, defined in the OASIS ontology, supports the representation of business agreements independently from their implementation and relying on standardised vocabularies. The defined *Ride2Rail Ontology for Agreements* enables the application of this concept to support the interoperability of business agreements in the multimodal transportation scenario considered by the Ride2Rail project. Two use cases were considered: the representation of the ride-sharing booking as an agreement between the driver and the passengers for dispute resolution, and the definition of incentives as agreements between different stakeholders to promote the usage of multimodal transportation. In the paper, we validated and exemplified the usage of the ontology by modelling the specific business agreements implemented in the project on the ledger.

The discussed approach can be generalised to support an ontological representation of smart contracts in different domains. The following steps summarise the discussed activities: (i) investigation of the business agreements to be modelled in the considered scenario (use cases and user stories), (ii) analysis of the domain terminology covered by the business agreements (facts and competency questions), (iii) identification of existing vocabularies covering the relevant domain entities and relationships, and/or implementation of an ontology supporting their representation, (iv) modelling of each business agreement as an ontological smart contract identifying the involved entities and the terms of the agreement, and, optionally, (v) representation of specific entities of the business agreement stored on the ledger using the ontology. In this way, different stakeholders are able to access through uniform terminology a description of the smart contracts and, possibly, their instances.

In future work, we will investigate the materialisation/virtualisation of smart contracts and/or related instances from the blockchain to enable querying according to the defined ontology. In particular, we will explore the configuration of semantic conversion pipelines [17] and the exploitation of the obtained knowledge graph. Moreover, we would like to extend the scope of the defined ontology to enable the representation of heterogeneous agreements in the multimodal transportation domain, for example, considering requirements for the sharing and electric mobility [18]. Finally, the evolution of the suite of IP4 ontologies will be taken into account to extend and update the defined ontology.

Acknowledgements

Programme. We would like to thank Daniele Santamaria for his support regarding the OASIS ontology.

References

[1] Cantone D, Longo CF, Nicolosi Asmundo M, Santamaria DF, Santoro C. Ontological Smart Contracts in OASIS: Ontology for Agents, Systems, and Integration of Services. In: Intelligent Distributed Computing XIV. Studies in Computational Intelligence. Cham: Springer International Publishing; 2022. p. 237-47.

[2] Ride2Rail Project. D3.5 - Ride-sharing Agreements Ledger Module; 2022. Project Deliverable (pending approval by EC). Available from: https://ride2rail.eu/resources-library/.

[3] Ride2Rail Project. D2.1 - First conceptualization of choice criteria and incentives; 2020. Project Deliverable. Available from: https://doi.org/10.5281/zenodo.4537031.

[4] Ride2Rail Project. D2.4 - Final conceptualization of choice criteria and incentives; 2021. Project Deliverable. Available from: https://doi.org/10.5281/zenodo.4607203.

[5] Ride2Rail Project. D3.1 - Quantitative Estimate of Service Quality Factors; 2021. Project Deliverable. Available from: https://doi.org/10.5281/zenodo.6803935.

[6] Poveda-Villalón M, Fernández-Izquierdo A, García-Castro R. Linked Open Terms (LOT) Methodology; 2019. Publisher: Zenodo. Available from: https://zenodo.org/record/2539305.

[7] Cano-Benito J, Cimmino A, García-Castro R. Towards Blockchain and Semantic Web. In: Business Information Systems Workshops. Lecture Notes in Business Information Processing. Cham: Springer International Publishing; 2019. p. 220-31.

[8] De Kruijff J, Weigand H. Ontologies for Commitment-Based Smart Contracts. In: On the Move to Meaningful Internet Systems. OTM 2017 Conferences. Lecture Notes in Computer Science. Cham: Springer International Publishing; 2017. p. 383-98.

[9] De Kruijff J, Weigand H. Understanding the Blockchain Using Enterprise Ontology. In: Advanced Information Systems Engineering. Lecture Notes in Computer Science. Cham: Springer International Publishing; 2017. p. 29-43.

[10] McAdams D. An Ontology for Smart Contracts. IOHK paper; 2017. p. 3. Accessed online at https://cryptopapers.info/assets/pdf/ontology_for_smart_contracts.pdf.

[11] Cantone D, Longo CF, Nicolosi-Asmundo M, Santamaria DF, Santoro C. Towards an Ontology-Based Framework for a Behavior-Oriented Integration of the IoT. In: Proceedings of the 20th Workshop "From Objects to Agents". vol. 2404 of CEUR Workshop Proceedings. Parma, Italy: CEUR; 2019. p. 119-26. ISSN: 1613-0073. Available from: http://ceur-ws.org/Vol-2404/#paper18.

[12] CONNECTIVE Project. D1.2 – Architectural Principles and Design A-REL; 2020. Project Deliverable. Available from: https://projects.shift2rail.org/s2r_ip4_n.aspx?p=MaaSive.

[13] Ruckhaus E, Anton-Bravo A, Scrocca M, Corcho O. Applying the LOT Methodology to a Public Bus Transport Ontology aligned with Transmodel: Challenges and Results. Semantic Web. 2021 Jan;Pre-press:1-19. Publisher: IOS Press. Available from: https://doi.org/10.3233/SW-210451.

[14] Feria SC, García-Castro R, Poveda-Villalón M. Converting UML-based ontology conceptualizations to OWL with Chowlk; 2021. Available from: https://openreview.net/forum?id=u1Vp2y_QE1.

[15] Poveda-Villalón M, Suárez-Figueroa MC, Gómez-Pérez A. Validating Ontologies with OOPS! In: Knowledge Engineering and Knowledge Management. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer; 2012. p. 267-81.

[16] Garijo D. WIDOCO: A Wizard for Documenting Ontologies. In: The Semantic Web – ISWC 2017. vol. 10588. Springer International Publishing; 2017. p. 94-102. Series Title: Lecture Notes in Computer Science. Available from: http://link.springer.com/10.1007/978-3-319-68204-4_9.

[17] Scrocca M, Comerio M, Carenini A, Celino I. Turning Transport Data to Comply with EU Standards While Enabling a Multimodal Transport Knowledge Graph. In: Proceedings of the 19th International Semantic Web Conference. vol. 12507. Springer; 2020. p. 411-29. Available from: https://doi.org/10.1007/978-3-030-62466-8_26.

[18] Scrocca M, Baroni I, Celino I. Urban IoT ontologies for sharing and electric mobility. Semantic Web. 2021 Jan;Pre-press:1-22. Publisher: IOS Press. Available from: https://doi.org/10.3233/SW-210445.