MASTER'S THESIS

Semantic Web Ontology Design Patterns for the Hohfeld Semantic-Conceptual Model

Zomerdijk, A.

Award date:
2018

Semantic Web Ontology Design Patterns for the Hohfeld Semantic-Conceptual Model

Degree programme:	Open University of the Netherlands, Faculty of Management, Science & Technology Business Process Management & IT master's programme
Course:	IM9806 Business Process Management and IT Graduation Assignment
Student:	A. Zomerdijk
Date:	02-09-2018
Thesis supervisor:	dr. L.W. Rutledge
Second reader:	prof. dr. ir. S.M.M. Joosten
Version number:	1.0
Status:	final version

Abstract

This paper proposes Ontology Design Patterns (ODPs) for the Hohfeld semantic-conceptual model. These ODPs facilitate modelling solutions for ontology engineering and reasoning with legal information. The Hohfeld semantic-conceptual model, as introduced by Van Engers & Nijssen, extends the Hohfeld legal relations with temporal aspects, events, and legal facts. We propose the *Temporal Information ODP* for temporal reasoning and determining the sequences of (legal) facts, actions, and events. We differentiate between valid times for temporal aspects of real world concepts and transaction times for the moment a fact is stored in a database. The *Legal Fact ODP* infers which actions and events count as certain as legal facts. This enables a qualification process, giving legal meaning to facts, and infers the legal consequences of events and actions. The *Temporal Legal Relations ODP* facilitates the expression of temporal Hohfeld legal relations. We implement this work with the Semantic Web technologies RDFS, OWL, and SWRL. This paper presents each ODP with a description of its intent, competency questions, and schema diagram. We evaluate the ODPs with a case study. For this, we use the car sale scenario of Van Engers & Nijssen. We provide an OWL file for each ODP for reusability.

Key terms

Ontology Design Patterns, Hohfeld Semantic-Conceptual Model, Temporal Reasoning, Legal Reasoning, Hohfeld Legal Relations, Semantic Web.

Contents

1. Introduction

Increasingly complex and ever faster changing laws and regulations confront organizations with the need for frequent assessment or even redesign of their processes (Van Engers & Nijssen, 2014b). Information systems can promote compliant behavior by monitoring the legitimacy of activities and business processes (Hulstijn, 2012). In addition, designing processes and IT systems based on legal requirements can enforce compliant behavior. However, Van Engers and Nijssen claim most companies have difficulties building IT systems based on legislation because of a lack of a consistent method for interpreting and analyzing the sources of the law (Van Engers & Nijssen, 2014b). Offering knowledge about laws and regulations in an accessible, manageable, and reusable form is essential when deriving requirements from the law (Dulfer, 2014). Ontologies can represent extracted knowledge from various legal texts in a structural form (Casanovas, González-Conejero, & Koker, 2017). We propose Ontology Design Patterns (ODPs) for representing temporal and legal information.

ODPs are domain specific solutions intended for ontology development with Semantic Web technologies (Hammar, 2017). The Semantic Web is designed for sharing information and ontologies (Berners-Lee, 2001). Knowledge representation with Semantic Web ontologies has the benefits that the information is explicit, reusable, and interpretable for both humans and machines (Pandey & Dwivedi, 2011). Within the legal domain, ODPs are useful for combining the knowledge of legal experts and system engineers (Casanovas et al., 2017).

For the interpretation of legal texts, Hohfeld proposes a conceptual framework of legal relations (Hohfeld, 1917). According to Schlag (2015), Hohfeld's work is still useful today for analyzing and understanding information from legal sources. Allen and Saxon introduce the A-Hohfeld language that extends Hohfeld's work with events and third legal parties that start or end legal relations (L. E. Allen & Saxon, 1995).

Van Engers and Nijssen introduce the Hohfeld sematic-conceptual model based on Hohfeld's legal relations. They propose a method for systematically analyzing legal texts and designing compliant information systems (Van Engers & Nijssen, 2014b). Like Allen & Saxon (1995), Van Engers and Nijssen expand Hohfeld's work with events. Van Engers and Nijssen propose a qualification process for inferring legal facts from events and actions. They distinguish between facts in real world, facts in an institutional reality, and legal facts. Understanding behavior in the real world requires a method for sequencing events and actions. Facts are stored in databases in an institutional reality. Legal facts have legal consequences and start or end legal relations. Van Engers and Nijssen extend Hohfeld's work with temporal aspects to represent temporal legal relations that are effective during a predetermined period. The objective of this research is to verify how, and to what extent, Semantic Web ODPs can implement and express the Hohfeld semantic-conceptual model of Van Engers & Nijssen (2014).

We propose ODPs for implementing temporal and legal information with Semantic Web technologies. The *Temporal Information ODP* enables representing time aspects of facts. We make a distinction between the valid time aspects of facts and the moment facts are stored in a database. We propose to use this ODP to sequence events. The *Legal Fact ODP* infers which events or actions count as legal facts. We propose to use this ODP for the qualification process of Van Engers and Nijssen with Semantic Web technologies (Van Engers & Nijssen, 2014b; Voorwinden, 2018). The *Temporal Legal Relations ODP* infers which legal relations are valid at a given time. We propose to use this ODP to sequence legal relations.

This work builds on previous research on Semantic Web technologies and relation algebra within the legal domain at the Open University in the Netherlands. Bos describes the implementation of rules in Semantic Web ontologies (Bos, 2013). Lalmohamed uses relation algebra for implementing Hohfeld's legal relations for defining legal compliant requirements (Lalmohamed, 2014). Slootweg implements the Hohfeld legal relations with Semantic Web technologies (Slootweg, Rutledge, Wedemeijer, & Joosten, 2016). Voorwinden applies relation algebra for qualifying facts as legal facts according to the qualification process of Van Engers and Nijssen (Van Engers & Nijssen, 2014b; Voorwinden, 2018). De Klerk demonstrates ODPs as a solution for implementing the A-Hohfeld language with Semantic Web technologies (De Klerk, 2018).

The rest of this paper is structured as follows.
Chapter 2 provides the theoretical framework.
Chapter 3 clarifies the research method and examines the validity, reliability, and ethical aspects.
Chapter 4 shows the results of the empirical study. We provide a description and evaluation of each ODP. *Chapter 5* draws the conclusions and recommendations for practical use and future research.

2. Theoretical framework

Laws and regulations are subject to constant change. According to Van Engers and Nijssen, organizations must regularly review their processes for ensuring the legitimacy of their actions. Van Engers and Nijssen propose a method for systematically analyzing legal texts and for the design of compliant information systems (Van Engers & Nijssen, 2014b). Hulstijn explains information systems can contribute to the demonstrable compliance of organizations. First, automated systems collect evidence about compliant or incompliant behavior. Second, the design of fully compliant information systems can make unlawful behavior impossible (Hulstijn, 2012). In both cases, it is necessary to represent relevant legal information in a machine-interpretable way. Sharing and reusing knowledge about legislation is time-saving and efficient (Hammar, 2017). Ontologies store information in a reusable and machine-readable format (Pandey & Dwivedi, 2011).

2.1. Semantic Web Ontologies

The Semantic Web offers a framework for sharing data including the semantics, the meaning, and the information derived from the data (Shadbolt, Berners-Lee, & Hall, 2006). Ontologies support sharing, integrating, and managing information sources (O'Connor & Das, 2011). The Web Ontology Language (OWL) is designed for sharing information in the Semantic Web. OWL is suitable for making information interpretable for machines and present the information to people in an understandable form (Pandey & Dwivedi, 2011). Ontologies in OWL offer support in decision procedures and various types of inferencing (Shadbolt et al., 2006). The Semantic Web Rule Language (SWRL) is introduced by W3C as an expressive OWL-based rule language that extends OWL ontologies with increasing possibilities for knowledge representation (Horrocks et al., 2004). SWRL facilitates applying rules with Horn clauses for inferencing with OWL instances (Tao et al., 2010).

An important aspect of the Semantic Web is the Open World Assumption (OWA) (De Klerk, 2018; Slootweg et al., 2016). The opposite of this is the Closed World Assumption (CWA). OWA presumes an open world where more information is available and missing information is asserted as 'unknown' (Antoniou, 2008). CWA presumes all necessary knowledge is complete and may evaluate a statement as 'false', when the system is unable in proving the statement is true. OWA is associated with monotonicity. We provide ODPs built with RDFS, OWL, and SWRL. RDFS, OWL, and SWRL support monotonic inferencing. This means existing knowledge within the ontology cannot be removed or modified. Therefore, it is common practice that OWL ontologies cannot make conclusions that could change if more information became available.

2.2. Ontology Design Patterns

Creating ontologies is expensive and time-consuming, since it requires expertise and domain knowledge (Hammar, 2017). For efficiency, reusing existing domain knowledge stored in Semantic Web ontologies is good practice (Trokanas, 2015). Patterns promote the reuse of ideas in a different context (Fowler, 1997). ODPs are reusable patterns for the Semantic Web. As building blocks, ODPs increase the efficiency of building ontologies and sharing knowledge within a certain domain (Karima, Hammar, & Hitzler, 2016). The use of ODPs as design solutions can reduce the costs of creating ontologies.

Although Semantic Web ontologies and ODPs are both presented with reusable OWL-files, some differences apply. The main difference is that ontologies are aimed at modelling one shared conceptualization, while ODPs are aimed at applicability within multiple contexts (Hammar, 2017). ODPs provide reusable building blocks for reoccurring modeling problems. In addition, clear documentation is an important part of ODPs for promoting reusability (Karima et al., 2016).

We propose ODPs for the Hohfeld semantic-conceptual model (Van Engers & Nijssen, 2014b). for promoting reuse, Karima et al. (2016) claim paying attention to the quality of the documentation of ODPs is essential. They describe best practices of OPDs and conclude a required part of ODPs is a visual presentation of schema diagrams. Therefore, this paper presents schema diagrams of each ODP. According to Karima et al. (2016), other key factors of the documentation of ODPs are use examples and competency questions. Competency questions are general questions that the ODP can answer. These questions help with understanding the intent of an ODP (Karima et al., 2016). Accordingly, we provide each ODP with a general description that addresses the competency questions. We evaluate our ODPs with example scenarios from Van Engers and Nijssen's car sale scenario (Van Engers & Nijssen, 2014a).

2.3. Hohfeld

Hohfeld offers a framework of legal relations for constructively analyzing laws (Hohfeld, 1917). His work is promotes analyzing the essence of laws and legal relations. Hohfeld describes a main problem of legal reasoning is that the expression of legal statuses is ambiguous when it is reduced to only 'rights' and 'duties'. He suggests the use of eight concepts as a basis for unambiguous expression of legal relations (Hohfeld, 1917). Table 1 shows Hohfeld's legal relations.

Table 1 Hohfeld legal relations (Hohfeld, 1917)

Jural Correlatives	Right No-right	⟷ ⟷	Duty Privilege	Power Disability	⟷ ⟷	Liability Immunity
Jural Opposites	Right Duty	⟷ ⟷	No-right Privilege	Power Liability	⟷ ⟷	Disability Immunity

Schlag states Hohfeld's work is useful for deriving legal relations from legal texts. He explains every legal relationship has a correlation. If there is an agent with a certain kind of right, there exists also an agent with a kind of duty (Schlag, 2015). In contrast, the jural opposites define the same agent can never have both a Right and a NoRight or a Duty and Privilege at the same time. Agnoloni and Francesconi used Hohfeld's work as a basis for their Provision Model, where a set of provisions expresses a law or regulation. Within the Provision Model, each legal relation has the attributes bearer and counterpart for expressing correlations between Hohfeld's legal relations (Agnoloni &

Francesconi, 2011). Each legal relation has a bearer agent and a counterpart agent for representing the jural correlatives.

Allen and Saxon offer the A-Hohfeld language that extends Hohfeld's work with conditional legal relations. Events fulfill the conditions of these legal relations (L. E. Allen & Saxon, 1995). De Klerk implements the A-Hohfeld language with Semantic Web technologies in ODPs (De Klerk, 2018). Van Engers and Nijssen introduce the Hohfeld semantic-conceptual model and extend Hohfeld's work with actions, events, legal facts, time perspective, and temporal legal relations (Van Engers & Nijssen, 2014b). In this work, we implement the Hohfeld semantic-conceptual model in Semantic Web ODPs.

2.4. Hohfeld Semantic-Conceptual Model

Van Engers and Nijssen introduce the Hohfeld semantic-conceptual model. This model extends Hohfeld's legal relations with time perspective, events and actions, and a qualification process for determining which events and actions count as a legal facts (Van Engers & Nijssen, 2014b). Van Engers and Nijssen describe legal facts have legal consequences and create or end legal relations. Legal relations are temporal and change over time (Van Engers & Nijssen, 2014b). The temporal legal relations sequence each other in time. The end of one legal relation is often the beginning of another legal relation. We propose ODPs for the Hohfeld semantic-conceptual model.

2.4.1. Sequence of Events and Actions

Behavior in the real world is complex. Van Engers and Nijssen explain the importance of understanding the sequence of events and actions for addressing complex situations (Van Engers & Nijssen, 2014b). We propose ODPs for storing and reasoning with temporal information about events and actions, so we can determine their sequence.

Within the Semantic Web, Allen's interval algebra is widely used in ontologies for representing and reasoning with temporal information (Grüninger & Li, 2017). Allen offers a method for representing temporal relationships between time intervals (J. F. Allen, 1983). Figure 1 shows Allen's temporal operators for temporal relationships.

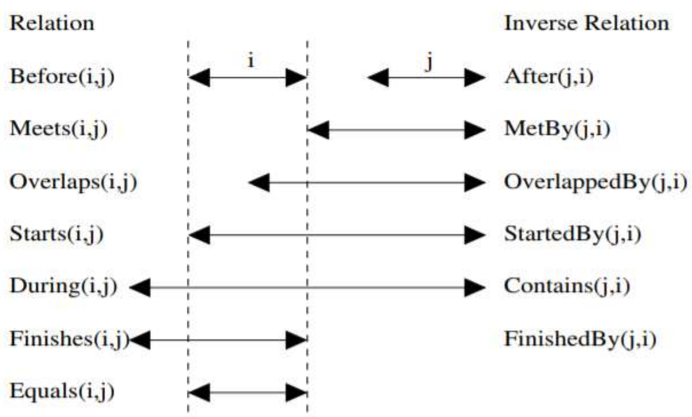

Figure 1 Allen's Temporal Relationships (J. F. Allen, 1983; Batsakis, Petrakis, Tachmazidis, & Antoniou, 2016, p. 4)

The 'Time Ontology in OWL' is perhaps the most comprehensive and fully developed Semantic Web ontology with an implementation of Allen's intervals (Hobbs & Pan, 2017). However, O'Conner & Das argue a time ontology needs to offer support in temporal reasoning, such as representing and comparing durations between time units (O'Connor & Das, 2011). They offer the SWRL Temporal

Ontology for the consistent representation of temporal information. O'Conner & Das explain their ontology connects facts to temporal information.

The SWRL Temporal Ontology applies the SWRLTemporalBuiltInLibrary with SWRL built-ins for temporal reasoning. The library provides the full set of Allen's temporal relationships, such as before, after, and meets (O'Connor & Das, 2011). This enables creating rules that specify relationships between time units. For example, a rule can specify a certain date must be before another date to satisfy the rule. In addition, the temporal built-ins facilitate rules that compare time units with each other and calculate whether the time length between two instants meets a determined duration. As such, rules can specify the difference between two dates must satisfy a certain duration. Within the Temporal Information ODP, we extend the SWRL Temporal Ontology with a class valid interval for representing the length of durations. In line with Dulfer's work, we distinguish between three categories of valid times (Dulfer, 2014):

Instant	A point on a time line;
Period	The timespan between two known instants. A period has a start date and a finish date;
Interval	A timespan, with unknown start and finish instants.

An instant is a moment in time with zero extent or duration. A period is a length of time with known start and end dates. An interval is a length or time, without knowing the start date or the end date. An interval represents a time unit triggered by an event or action. For example, seven days after signing the contract, the contracted person must transfer the money. The action of signing a contract triggers the interval of seven days.

According to Snodgrass, abilities for modelling temporal information is essential for many information systems (Snodgrass, 1992). Snodgrass recommends differentiating between valid times and transaction times. Valid times concern real world temporal information about facts. Transaction times refer to the moment information is inserted in a database (Snodgrass, 1992). When real world facts and their temporal information are stored in a database, the time the fact is stored concerns transaction time. A fact can refer to both valid times and transaction times. For example, an employment contract refers to a valid time 'the first working day' and a transaction time 'the employer stored the contract in a database'.

We propose the Temporal Information ODP for sequencing events and actions. With the Temporal Information ODP, we propose a pattern for representing temporal information with instants, periods, and intervals. We differentiate between valid times and transaction times.

2.4.2. Qualification of Legal Facts

Van Engers and Nijssen explain events and action may have legal consequences, depending on their qualification (Van Engers & Nijssen, 2014b). Collecting information about these facts helps in determining their legal effect. Van Engers and Nijssen introduce a qualification process for determining the legal consequences of facts depending on their qualification (Van Engers & Nijssen, 2014b). They distinguish between facts in the real world, institutional reality, and legal reality. Facts in the real world, referred to as brute facts by Van Engers and Nijssen, are stored as data in an institutional reality. As shown in Figure 2, a qualification process defines which real world facts count as legal facts in the institutional reality (Van Engers & Nijssen, 2014b).

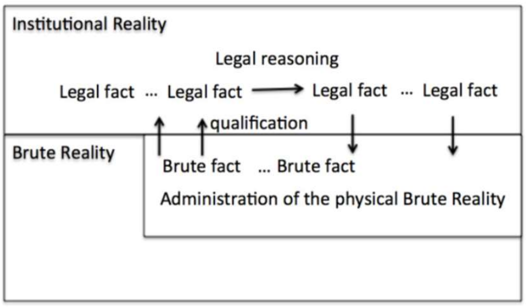

Figure 2 Qualification of legal facts (Van Engers & Nijssen, 2014b)

Voorwinden introduces patterns for the implementation of legal norms with relation algebra. In his qualification pattern, he applies the qualification process of Van Engers and Nijssen (Van Engers & Nijssen, 2014b; Voorwinden, 2018). Voorwinden considers facts as entities that are observable and measurable. Legal facts are facts with legal consequences. He defines qualification rules with relation algebra for qualifying actions as legal facts (Voorwinden, 2018).

Voorwinden proposes relationships between actions and references sources. Reference sources provide more information about actions. By forming relationships between actions and reference sources, facts are qualified as legal facts. Figure 3 shows the pattern for qualification that Voorwinden implements with relation algebra. A fact is qualified as a legal fact with a reference to an object and a reference source (Voorwinden, 2018).

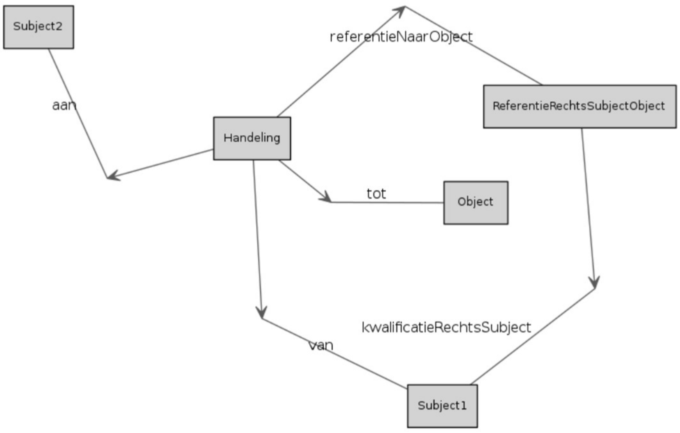

Figure 3 Pattern for qualification (Voorwinden, 2018)

We propose the Legal Fact ODP as a pattern for the qualification process with Semantic Web technologies. We qualify both actions and events as legal facts with relationships to reference sources, objects, and subjects.

2.4.3. Temporal Legal Relations

Van Engers and Nijssen extend Hohfeld's work with temporal aspects. They add start and end dates to legal relations. This enables creating, changing, and terminating legal relations over time (Van Engers & Nijssen, 2014b). In addition, temporal aspects of legal relations enable sequencing the legal relations. Van Engers and Nijssen describe temporal legal relations with fact types and integrity rules (Van Engers & Nijssen, 2014b).

Fact type legal relation: *<Party-Right-Side> in the role of <Kind-Of-Right> has a legal relation with <Party-Duty-Side> in the role of <Kind-Of-Duty> with respect to <Matter>.*

Integrity rule: *The combination of Party-Right-Side, Kind-Of-Right, Party-Duty-Side and Matter is unique.* In other words, *the combination Party-Right-Side, Kind-Of-Right, Party-Duty-Side and Matter determines the Kind-Of-Duty.*

Figure 4 shows the pattern for legal relations of the Hohfeld semantic-conceptual model. This pattern describes legal relations between two parties, a party right side with a kind of right and a party duty side with a kind of duty. Subcategories of the kind of right are Claim, Privilege, Power, and Immunity. Subcategories of the kind of duty are Duty, NoRight, Liability, and Disability. Each legal relation has a reference to a specific matter. Matter refers to domain specific facts (Van Engers & Nijssen, 2014b). In other words, a person or entity has rights or duties with respect to certain (legal) facts.

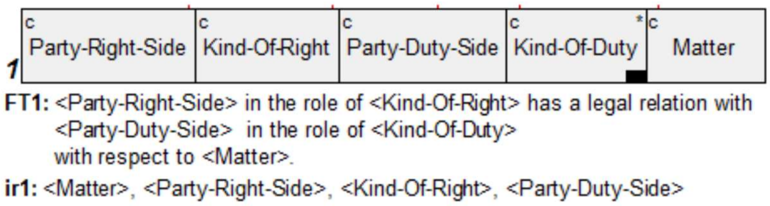

Figure 4 Pattern for legal relations (Van Engers & Nijssen, 2014b)

Van Engers and Nijssen extend Hohfeld's legal relations with time perspective by adding the following fact type: *The legal relation was established on <Date-Time-Established>, it starts to become effective on <Date-Time-Effectiveness-Start> and it has terminated respectively is supposed to end its effectiveness on <Date-Time-Effectiveness-Ended>.*

Each temporal legal relation has one kind of right and one kind of duty. According to Hohfeld's work, only certain combinations of rights and duties are allowed (Hohfeld, 1917). Van Engers and Nijssen express permitted legal relations with the pattern shown in Figure 5 and the integrity rule: *The following combinations are permitted {<Claim, Duty>, <Privilege-NoRight>, <Power, Liability>, <Immunity, Disability>}.*

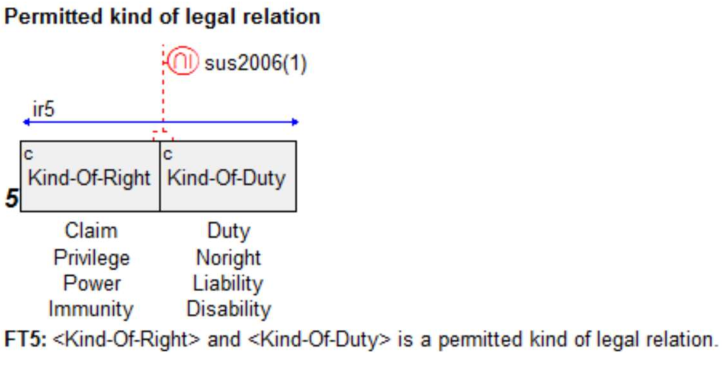

Figure 5 Permitted kind of legal relations (Van Engers & Nijssen, 2014b)

We propose the Temporal Legal Relations ODP for temporal legal relations. Temporal legal relations extend the original Hohfeld legal relations with time perspective and matter. The Temporal Legal Relations ODP enables sequencing temporal legal relations. Moreover, in the ODP we offer rules for inferencing which duties belong to which rights.

3. Research method

3.1. Conceptual Design

The aim of this work is examining how, and to what extent, the Hohfeld semantic-conceptual of Van Engers & Nijssen can be implemented with Semantic Web technologies in ODPs. This work offers reusable ODPs for reasoning within the legal domain. We especially aim at offering reasoning solutions for temporal information and legal facts. Offering knowledge about the law in a structured, accessible, reusable, and manageable way facilitates system designers and legal experts in sharing knowledge about laws and regulations.

We study existing models and ontologies for temporal reasoning or applying Hohfeld's legal relations. Where possible, we reuse existing knowledge and ontologies that are relevant for this work. Our selection of ontologies is based on reusability, relevance, and reasoning capabilities. The next section elaborates on the tools and ontologies we use in the ODPs. For validity of this research, we provide a full description of each ODP.

We validate the ODPs with a single exploratory case study (Saunders, Lewis, & Thornhill, 2015). Case study research is best suitable to answer 'how, and to what extent' research questions (Saunders, Lewis & Thornhill, 2015). Because of the exploratory nature of this research, we consider a case study the best research method.

Considering we implement the work of Van Engers and Nijssen, we use a corresponding scenario for the case study. Van Engers and Nijssen introduce the car sale scenario for explaining their Hohfeld semantic-conceptual model with a use case (Van Engers & Nijssen, 2014a). Van Engers and Nijssen divided the car sale scenario into periods and transitions with a predetermined sequence of actions and events. We use this scenario for validating our ODPs and the Semantic Web implementation of the Hohfeld semantic-conceptual model.

Internal validity

Internal validity refers to how well a research is conducted. It is important that causal conclusions based on the research are warranted. The internal validity depends on the quality and clarity of the analysis and drawn conclusion. We select an existing theory of Van Engers and Nijssen and corresponding use case of the car sale scenario (Van Engers & Nijssen, 2014a). The coherent selection of a model and corresponding use case reduces the change of selection bias.

External validity

External validity concerns the possible generalization of the research results. Generalization of the results seems impossible at first, since most case studies are not externally valid (Saunders et al., 2015). However, some generalization is possible because the case study consists of a critical study of the car sale scenario with a sequence of events, legal facts, and legal relations. We validate the ODPs are capable of implementing this scenario. Future research can confirm the ODPs can implement other scenarios with time-related aspects as well.

Reliability

Reliability deals with the repeatability of the study. We provide schema diagrams of the ODPs and describe each step of the case study evaluation to promote other researchers check the reliability. We further increase possible repeatability by using open source tools. Moreover, we provide OWL files of the ODPs at http://is.cs.ou.nl/OWF/Index.php5/Master_Thesis_Annalotte_Zomerdijk

3.2. Technical Design

We implement our ODPs using the Semantic Web technologies RDFS, OWL, and SWRL. For the repeatability of this work, we use open source software that is publicly available. This work uses Protégé (M. A. Musen, 2015). The ontology editor Protégé is open source and supports the development of Semantic Web content (Knublauch H., 2004).

For future compatibility and comparability between existing implementations of Hohfeld's work in ODPs and our ODPs, we use the same tools as De Klerk. An exception to this is De Klerk's choice for the Pellet reasoner (De Klerk, 2018). While the Pellet reasoner is a powerful tool for inspecting the consistency of ontologies and explain inferences, it cannot reason with the temporal built-ins from the SWRL Temporal Built-Ins Library. Therefore, we use the rule engine 'Drools' for reasoning with temporal information (RedHat, 2018). Table 2 is an overview of the Semantic Web tools we use in this work.

Table 2 Semantic Web tools used in this work

Tool	Version	Source of literature
Protégé	5.2.0	(Mark A. Musen & Protégé, 2015)
SWRL Tab Protégé plug-in	1.0.3 (part of Protégé 5.2.0)	(O'Connor, Shankar, Nyulas, Das, & Musen, 2008)
Drools Rule engine	2.0.5	(RedHat, 2018)

Reusing existing ontologies is efficient and good practice when building ontologies (Trokanas, 2015). This work reuses the existing ontologies and OPDs shown in Table 3.

Table 3 Overview of this works use of existing building blocks

Building block	Source of literature
SWRL Temporal Ontology	(O'Connor & Das, 2011)
SWRL Temporal Built-Ins Library	(O'Connor & Das, 2011)

This work follows the recommendations of Karima et al. (2016) by presenting each OPD with a general description of its intent, competency questions, and schema diagram. The schema diagram shows the class hierarchy and object relations in the ODP. Black lines show the class hierarchy. Blue arrows show relationships between classes and object properties. Boxes with blue background indicate original work of this paper. Figure 6 shows the meaning of the icons we use in the schema diagrams.

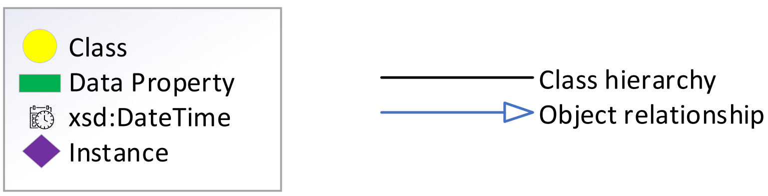

Figure 6 Explanation of the icons used in the schema diagrams

3.3. Data analysis

We evaluate our ODPs with examples from the car sale scenario of Van Engers & Nijssen. Table 4 shows the events and actions, qualification of legal facts, and temporal legal relations for each period or transition of the car sale scenario. This scenario addresses the legal process of transferring the ownership of a car (Van Engers & Nijssen, 2014a).

Table 4 Car sale scenario (Van Engers & Nijssen, 2014a)

Period or Transition	**Events and Actions**	**Qualification of Legal Facts**	**Temporal Legal Relations**
P1	Person A owns a car.		*Privilege-NoRight* Person A has the privilege and person B the NoRight to use the car. *Power-Liability* Person A has the power to offer the car for sale.
T1	Person A offers the car for sale for €20.000 on website W.	The offer counts as a legal fact if person A owns a car and offers that car at website W.	*Liability - Power* Person A creates a liability for herself. Person B has the power to accept the offer. *Termination Power – Liability* Person B can accept the offer and no longer has a liability.
P2	Person A may keep or withdraw the offer.		*Power-Liability* Person A has the power to withdraw the offer.
T2	Person B accepts the offer. Person A and Person B agree to exchange the car and €20.000 after seven days.	The acceptance of the offer counts as a legal fact if person B accepts an open offer at website W and accepts that offer in writing.	
P3	Person A cannot withdraw the offer.		*Continue Privilege-NoRight* Person A still has the privilege and person B the NoRight to use the car. *Termination of Liability-Power* Person A no longer has the power to withdraw the offer.
T3	Start of seven days after T2.		
P4	Seven days after T2.		*Claim-Duty* Person B has a claim on the car and Person A to deliver the car. *Claim-Duty* Person A has a claim on the €20.000. Person B has a duty to deliver.
T4	Person B pays 20.000. Person A exchanges the car to person B.		*Termination of Privilege-NoRight* Person A no longer has the privilege and person B no longer has the NoRight to use the car.
P5	Person B owns the car.		*Privilege-NoRight* Person B has the privilege and person A the NoRight to use the car. *Power-Liability* Person B has the power the offer the car for sale.

4. ODPs for the Hohfeld Semantic-Conceptual Model

The following sections describe the results of the empirical part of this work. We developed, implemented, and tested the ODPs with Semantic Web technologies. The main contribution of this work consists of the following ODPs for the Hohfeld semantic-conceptual model:
- Temporal Information ODP
- Legal Fact ODP
- Temporal Legal Relation ODP

4.1. Temporal Information ODP

The Temporal Information ODP facilitates the connection between facts and their associated temporal information. We implement Snodgrass's proposal of making a clear difference between valid times and transaction times (Snodgrass, 1992). For reasoning with temporal information, we extend the SRWL Temporal Ontology (O'Connor, 2017) with transaction times. Similar to Van Engers and Nijssen, we distinguish between the intuitional reality and the real world (Van Engers & Nijssen, 2014b). Valid times represent real world temporal concepts. Transaction times represent temporal information of administrative operations in the institutional reality. We distinguish between instants, intervals, and periods (Dulfer, 2014).

Intent

The Temporal Information ODP enables temporal reasoning between times, such as stating a transaction instant takes place before a valid instant. In addition, the Temporal Information ODP sequences events.

Competency Questions

For insight in the kind of questions the ODP can answer, we provide several competency questions (Karima et al., 2016):

- *What is the valid period of a contract?*
- *On what date does a duty expire?*
- *How much time is there be between the agreement and payment of the agreed amount?*

4.1.1. Schema Diagram of Temporal Information ODP

For insight in the relationships between classes and properties, Figure 7 presents a schema diagram of the Temporal Information ODP. The Temporal Information ODP reuses parts of the SWRL Temporal Ontology. Bold texts in blue boxes specify our additions to the SWRL Temporal Ontology (O'Connor & Das, 2011).

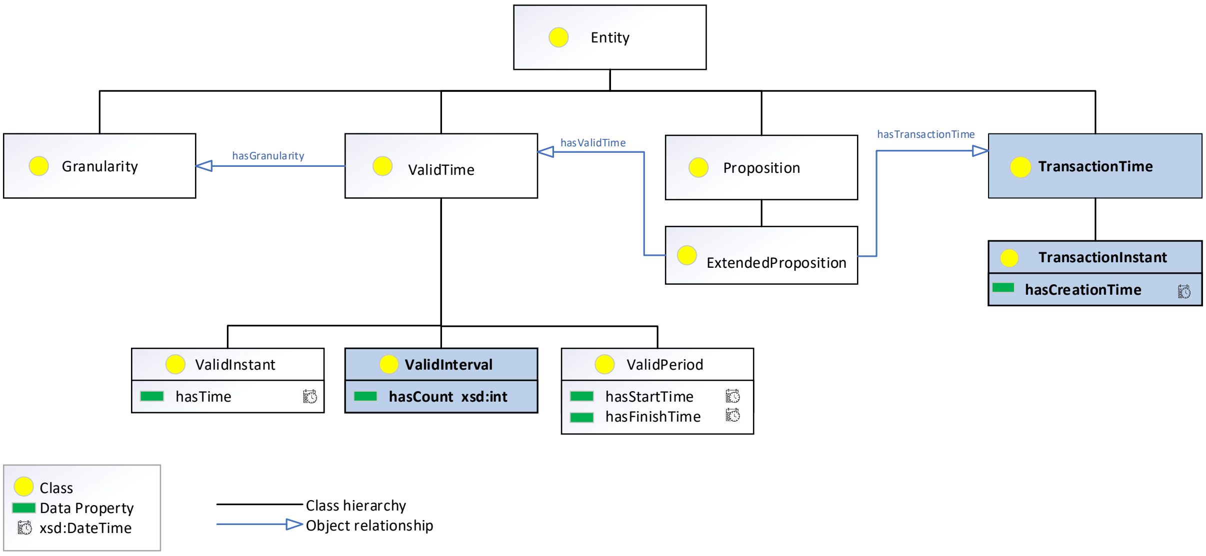

Figure 7 Temporal Information ODP, blue boxes indicate our extensions to the SWRL Temporal Ontology (O'Connor, 2017)

We extend the SWRL Temporal Ontology with transaction times. Transaction times concern the instants facts are saved in a database as stored data (O'Connor & Das, 2011; Snodgrass, 1992).

Another addition is the class valid interval. We propose using valid intervals for representing time lengths. This enables expressing durations without a predetermined start or end date. In the car sale scenario, the seller and buyer must exchange the money and car within seven days after the acceptance of an offer. In this example, a duty must be performed within the predetermined interval of seven days. The start date of this interval is unknown, until someone accepts the offer. The class valid interval enables expressing the time length of seven days, without knowing the start date. Table 5 gives an overview and explanation of our additions to the SWRL Temporal Ontology.

Table 5 Explanation of our additions to the SWRL Temporal Ontology (O'Connor, 2017)

Class	**Property**	**Explanation**
Transaction Time		A Transaction Time represents the time a fact is stored in a database.
Transaction Instant	*hasCreationTime*	The instant the fact is stored in a database.
Valid Interval	*hasCount*	A valid interval represents a length of time, without knowing the start or end time. The data property *hasCount* stores the duration of the interval in the ontology with an integer.

4.1.2. Evaluation of the Temporal Information ODP

Example from car sale scenario

We evaluate the Temporal Information ODP with an example from the car sale scenario (Van Engers & Nijssen, 2014a). The car sale scenario refers to sequences of events, valid times, transaction times, and intervals. Therefore, we consider this scenario suitable for demonstrating temporal reasoning with the Temporal Information ODP.

In the car sale scenario, person A first offers the car for sale before person B can accept the offer. We consider the moment person A offers the car for sale as a transaction instant. The car is officially for sale from the moment the offer is stored in the database of website W.

Person A and person B agree to exchange the money and car within an interval of seven days from moment person B accepts the offer (Van Engers & Nijssen, 2014a).

Scenario 1: Person A and Person B agree to exchange the car and money within seven days.

Step 1: In the first step, we determine person A first offers the car for sale before person B accepts the offer and buys the car.

The moment person A offers the car is a transaction instant. According to the car sale scenario, the offer is official as soon as it is stored in the database of website W. We validate person B accepts the offer after the transaction instant by implementing instances from the car sale scenario into the ODP. We implement this with instances from Table 6. We introduce person C for confirming it is not possible to accept an offer before the transaction date. In the example, person C tries accepting an offer before the transaction date.

Table 6 Instances for the car sale scenario with event sequence

ExtendedProposition		**TransationInstant**		
PersonA	*hasTransactionTime*	OffersCar	*hasCreationTime*	01-03-2018
ExtendedProposition		**AcceptOffer**		
PersonB	*hasValidTime*	AcceptsOffer1	*hasTime*	02-03-2018
PersonC	*hasValidTime*	AcceptsOffer2	*hasTime*	28-02-2018

SWRL offers temporal reasoning facilities and enables rules for determining the sequence of instances. We use SWRL for determining the sequence of the date of acceptance of an offer and the transaction instant of that offer. We add Code segment 1 for inferring a valid acceptance is after the transaction date of an offer.

Code segment 1 ValidInstant is after TransactionInstant

```
AcceptOffer(?vi) ^ hasTime(?vi, ?tt) ^ TransactionInstant(?oc) ^ hasCreationTime(?oc, ?ct) ^
temporal:after(?tt, ?ct) -> ValidAccept(?vi)
```

Figure 8 shows the ODP infers AcceptsOffer1 from person B as valid. The AcceptsOffer2 of person C is not valid. This a correct inference. Accepting an offer is only valid after the transaction date. We confirm the ODP enables determining the sequence of transaction times and valid times.

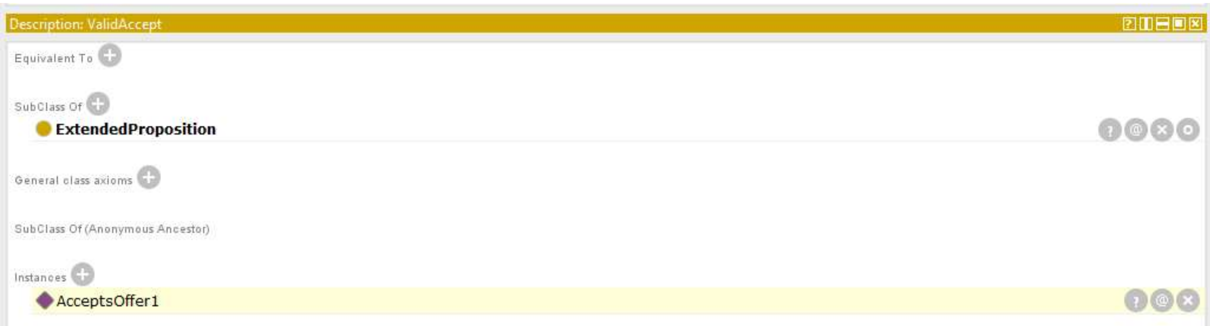

Figure 8 Result of code segment 1 indicates a valid sequence of events

Step 2: In step 2, person A and person B agree to exchange the money and car within an interval of seven day after the instant person B accepts the offer.

We validate the ODP distinguishes between an instant outside the predetermined interval and an instant within the interval. In our example, person B honors the agreement and person A does not.

Person B pays the agreed amount within seven days. Person A violates the agreement does not deliver the car within seven days. We implement this with the instances from Table 7 and Table 8.

Table 7 Instances for a valid interval of seven days

ValidInterval		**xsd:integer**		**Granularity**
within7Days	*hasCount*	7	*hasGranularity*	Days

Table 8 Instances for the dates person A and person B exchange the car and money

ExtendedProposition		**ValidInstant/Exchange**		**xsd:dateTime**
PersonA	*hasValidTime*	GivesCar	*hasTime*	01-04-2018
PersonB	*hasValidTime*	GivesMoney	*hasTime*	05-03-2018

We use SWRL for calculating the duration between the time units, since SWRL offers temporal reasoning solutions. Code segment 2 calculates the duration between the instant person B accepts the offer and the instant the exchange takes place.

Code segment 2 A timely exchange is within a duration of seven days after the ValidAccept

```
Exchange(?e) ^ hasTime(?e, ?tt) ^ ValidInterval(?vi) ^ hasCount (?vi, ?hc) ^ ValidAccept(?va) ^
hasTime(?va, ?ht) ^ temporal:duration(?du, ?tt, ?ht, temporal:Days) ^ swrlb:lessThanOrEqual(?du,
?hc) -> TimelyExchange(?e)
```

Figure 9 shows person B exchanges the money on time. However, due to the OWA, no exact conclusions can be drawn from the missing result of person A. It is possible for person A to be late with the exchange of the car or person A never exchanges the car at all. We do know person A is not on time with delivering the car. We confirm the Temporal Information ODP enables determining an instance is within a predetermined interval.

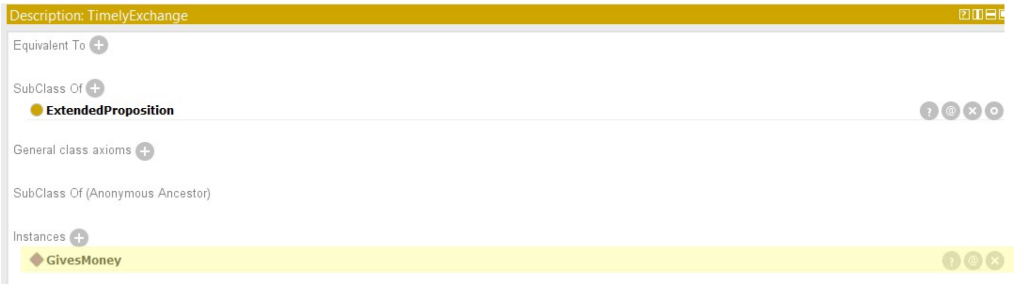

Figure 9 Result of code segment 2 demonstrates person B gives the money on time

Conclusions of the evaluation

The evaluation confirms the Temporal Information ODP enables temporal reasoning, such as sequencing events and comparing the time length between two dates to an interval. SWRL rules enable inferencing the time between two instants is shorter than a predetermined interval. The ODP is likely useful for other purposes besides time as well. Such as ensuring the amount of paid money is correct. By implementing amounts with integers instead of dates, the paid amount of money is comparable with an agreed amount. This can be interesting for future research.

4.2. Legal Fact ODP

The Legal Fact ODP enables inferring actions and events as legal facts. We implement the qualification method of Van Engers and Nijssen. The Legal Fact ODP is based on the pattern of qualification Voorwinden implements with relation algebra (Van Engers & Nijssen, 2014b; Voorwinden, 2018).

Intent

The intent of the Legal Fact ODP is inferring which events or actions count as legal facts.

Competency Questions

We provide competency questions as examples what kind of question the ODP can answer (Karima et al., 2016):

- *What kind of legal fact is an event?*
- *To which object is the action of subject 1 directed?*
- *Is an action a valid as a legal fact?*

4.2.1. Schema Diagram of Legal Fact ODP

Figure 10 presents a schema diagram of the Legal Fact ODP. This ODP is based on the qualification process of Van Engers and Nijssen that Voorwinden implements with relation algebra in Ampersand (Van Engers & Nijssen, 2014b; Voorwinden, 2018).

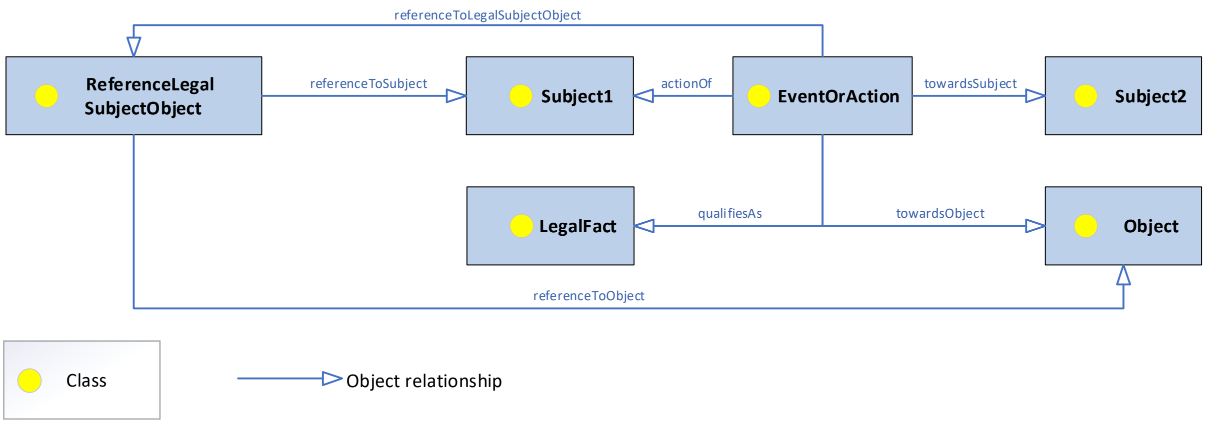

Figure 10 Schema diagram of the Legal Fact ODP based on the qualification process of Van Engers and Nijssen (Van Engers & Nijssen, 2014b; Voorwinden, 2018)

For qualifying facts as legal facts, a relationship is necessary to a source that provides more information about the fact (Voorwinden, 2018). Similar to Voorwinden, we establish a relationship between facts and sources that provide more information about facts. With this information, we infer legal facts from actions and events. In addition, this relationship serves as proof of the situation asserted by the fact (Voorwinden, 2018).

In Voorwinden's pattern for qualification, actions have a reference to a 'Reference Legal Subject Object' (Voorwinden, 2018, p. 23). We implement the class 'ReferenceLegalSubjectObject' in the ODP. This reference has a relationship to an object and a subject. We extend Voorwinden's work with events. This allows the qualification of both actions and events as legal facts. Van Engers and Nijssen indicate legal actions and events have legal consequences (Van Engers & Nijssen, 2014b).

4.2.2. Evaluation of the Legal Fact ODP

Examples from car sale scenario

We evaluate the Legal Fact ODP with examples from the car sale scenario (Van Engers & Nijssen, 2014a). The car sale scenario refers to actions and events. We us the examples of an event that person A offers a car for sale and an action of person B who buys the car. Both the event and the action count as legal facts. Therefore, we consider the car sale scenario suitable for the evaluation of inferencing legal facts with the Legal Fact ODP.

In the first scenario, person A offers her car for sale. We qualify this event as a legal fact if person A offers her own car for sale on website W. Person A cannot offer another car than her own. In this example, the offer is a legal fact when person A offers her car on website W.

In the second scenario, person B buys the car. This action counts as a legal fact if person B accepts an open offer for sale on website W in writing (Van Engers & Nijssen, 2014a). If the car is not open for sale on website W person B cannot accept the offer.

Scenario 1: Person A offers her car for sale on website W.

Step 1: In step 1, we determine person A owns car A. In the car sale scenario, during P1 person A owns a car (Van Engers, 2014 #73). We refer to this car as car A. We put this in the Legal Fact ODP with the instances from Table 9. We also introduce a car B that belongs to person B for evaluating person A only owns car A.

Table 9 Instances that person A and person B own cars

EventOrAction		Subject1		Subject2
ownsCarA	*actionOf*	PersonA	*towardsSubject*	carA
ownsCarB	*actionOf*	PersonB	*towardsSubject*	carB

For determining which events or actions count as a legal facts, we need a reference to a legal source, an object, and a subject. We implement this with the combiantion of triples '*referenceToLegalSubjectObject*', '*referenceToSubject*', and '*referenceToObject*'. SWRL enables rules with combinations of triples. We implement code segment 3 in the ODP.

Code segment 3 Events or actions count as legal facts with a reference to a legal subject object, a subject, and an object

```
EventOrAction(?eoa) ^ referenceToLegalSubjectObject(?eoa, ?so) ^ referenceToSubject(?so, ?s) ^
referenceToObject(?so, ?o) -> LegalFact(?eoa)
```

Figure 11 shows the action 'ownsCar' counts as a legal fact. However, we cannot determine whether this action belongs to person A or person B.

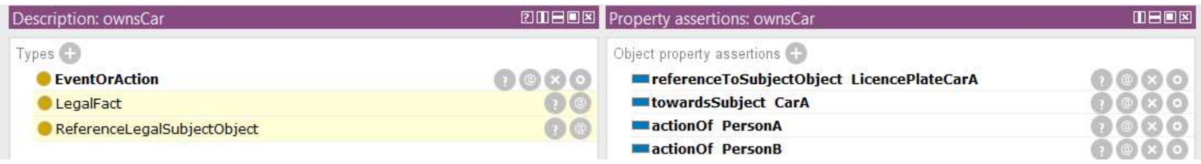

Figure 11 Inference of ownsCar as a legal fact

Events and actions with a relationship to a 'referenceToLegalSubjectObject' count as legal facts. The 'referenceToLegalSubjectObject' provides more information about events and actions with connections to 'referenceToSubject' and a 'referenceToObject'.

The events ‘ownsCarA’ and ‘ownsCarB’ count as legal facts if there is a reference to a license plate that belongs to the car and the person owning the car. The license plate needs a reference to the car and to person A or person B. We put this in the ODP with the instances from Table 10.

Table 10 Instances for the event that person A owns a car with a reference to a legal subject object

ReferenceLegal SubjectObject		**Subject1**		**Object**
LicensePlateCarA	*Reference toSubject*	PersonA	*referenceToObject*	LicensePlate11AA11
		PersonB		LicensePlate11BB11

The ‘referenceToLegalSubjectObject’ has a reference to the subject ‘personA’ and to a license plate. Figure 12 explains it is a legal fact that person A owns car A. There is no reference to a legal source for person B owning a car and this does not count as a legal fact. Therefore, it is correct there is no explanation for the event of person B owning a car.

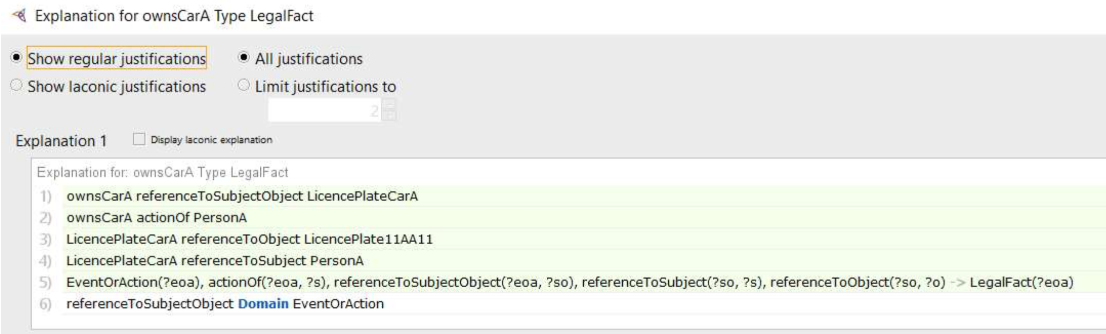

Figure 12 Explanation that person A owns car A is a legal fact

Step 2: In the car sale scenario, during T1, person A puts car A for sale (Van Engers & Nijssen, 2014a). Person A can only put her own car for sale and not the car belonging to person B. We use an example where person A offers both her own car and the car of person B for evaluating person A can only offer her own car for sale. We implement this example in the ODP with the instances from Table 11.

Table 11 Instances for the action person A offers car A and car B for sale at website W

EventOrAction		**Subject1**	**ReferenceLegal SubjectObject**		**Subject1**
offersForSaleCarA	*actionOf*	PersonA	saleOfferCarA	*ReferenceToSubject*	PersonA
offersForSaleCarB	*actionOf*	PersonA	saleOfferCarB	*ReferenceToSubject*	PersonB

For ensuring people can only put their own cars on sale, we extend code segment 3 with the property *actionOf*. Code segment 4 shows events or actions require a reference to the same subject 1 as the reference legal subject object, to count as legal facts.

Code segment 4 The event or action refers to the same subject 1 as the reference legal subject object

```
EventOrAction(?eoa) ^ actionOf(?eoa, ?s) ^ referenceToSubjectObject(?eoa, ?so) ^
referenceToSubject(?so, ?s) ^ referenceToObject(?so, ?o) -> LegalFact(?eoa)
```

The action ‘offersForSaleCarA’ and the corresponding reference to legal subject object ‘saleOfferCarA’ both refer to subject1 ‘person A’. Figure 13 and Figure 14 show person A’s action of offering car A for sale counts as a legal fact. The action ‘offersForSaleCarB’ is not a legal fact, since this is an action of person A, while the reference legal subject object refers to person B. We verify person A can only put her own car for sale.

Figure 13 The events person A owns car A and car A is offered for sale are legal facts

Figure 14 Person A offers car A for sale is a legal fact

Scenario 2: Person B buys the car.

Step 1: In step 1, we determine person A publishes the offer for sale of car A as open at the website W. Person B can only accept open offers. The open offer for sale counts as a legal fact if the offer is published at website W (Van Engers & Nijssen, 2014a).

We validate the offer only counts as a legal fact if the offer has a reference to the object ‘website W’ with two instances: 'publishes offer 1' and 'publishes offer 2', as Table 12 shows. The difference between these two instances is that person A publishes offer 1 on website W and not offer 2.

Table 12 Instances for the two actions that person A offers a car, with and without a reference to website W

EventOrAction		**Subject1**		**Subject2**		**Object**
publishesOffer1	*actionOf*	PersonA	*towardsSubject*	CarA	*towardsObject*	websiteW
publishesOffer2	*actionOf*	PersonA	*towardsSubject*	CarA		

Table 13 Instances that both offers have a reference to legal subject object that has a reference to website W

EventOrAction		**ReferenceLegal SubjectObject**		**Subject1**		**Object**
publishesOffer1	*referenceToLegal SubjectObject*	OpenOfferCarA	*referenceTo Subject*	PersonA	*referenceTo Object*	websiteW
publishesOffer2	*referenceToLegal SubjectObject*	OpenOfferCarA	*referenceTo Subject*	PersonA	*referenceTo Object*	websiteW

Table 13 shows both 'publishesOffer1' and 'publishesOffer2' have a relationship with the same reference legal subject object 'OpenOfferCarA'. The legal subject object has a reference to website W. We add code segment 5 for determining events or actions with a relationship to the same object as the ‘referenceToLegalSubjectObject’ are legal facts.

Code segment 5 Events or actions with a reference to the same object as the legal subject object count as legal facts

```
EventOrAction(?eoa) ^ towardsObject(?eoa, ?o) ^ referenceToSubjectObject(?eoa, ?so) ^
referenceToSubject(?so, ?s) ^ referenceToObject(?so, ?o) -> LegalFact(?eoa)
```

For optimization, we use as few rules as possible. We offer one rule for determining which events and action count as legal facts. Code segment 6 states events or actions count as a legal fact if:

- The action or event has a reference to a legal subject object;
- The action or event has a relation with the same subject 1 as the reference;
- And the action or event has a relation with the same object as the reference.

Code segment 6 Legal facts are events or actions with references to a legal subject objects, subjects, and objects

```
EventOrAction(?eoa) ^ referenceToSubjectObject(?eoa, ?so) ^ actionOf(?eoa, ?s) ^
referenceToSubject(?so, ?s) ^ towardsObject(?eoa, ?o) ^ referenceToObject(?so, ?o) ->
LegalFact(?eoa)
```

Figure 15 and Figure 16 show 'publishedOffer1' is a legal fact and 'publishedOffer2' is not. In this case, the offering of a car is a legal fact if the offer is placed by the owner on website W. The offer needs a reference to the subject person A and object website W. This demonstrates events or actions with references to the correct subjects and objects count as a legal facts.

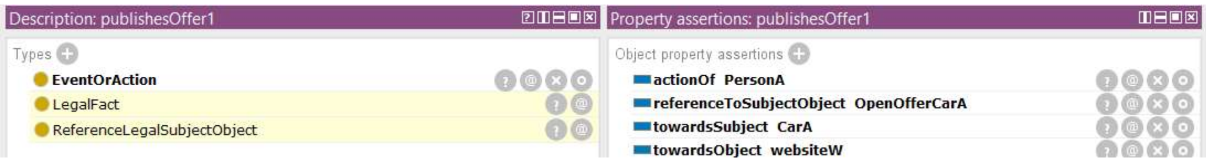

Figure 15 Person A publishesOffer1 is a legal fact because person A offers car A at website W

Figure 16 PublishesOffer2 is not a legal fact because the offer does not have a reference towards the object website W

Step 2: In this step, person B accepts the offer for sale of car A. This is part T2 of the car sale scenario. The acceptance of the offer counts as a legal fact if person B accepts an open offer for sale in writing. In addition, there must be a reference between the acceptance of person B and the 'OpenOfferCar'. The reference legal object subject 'OpenOfferCar' from step 1 is now an object. This only applies if the offer is a legal fact. We add code segment 7 for inferring a reference legal subject object is an object if it has a relationship with an event or action that counts as a legal fact.

Code segment 7 Inference of reference legal subject objects with relations to legal facts as objects

```
EventOrAction(?eoa) ^ LegalFact(?eoa) ^ referenceToSubjectObject(?eoa, ?rlso) -> Object(?rlso)
```

Figure 17 shows the OPD infers 'OpenOfferCarA' as an object. This allows us in creating a relationship between the action of person B to accept the offer and the object ‘OpenOfferCarA’. This leaves room for adding another reference to a legal source.

Figure 17 The open offer of car A in an object

We put a written acceptance in the ODP as a reference legal subject object. Table 14 shows instances of person B accepting the offer of car A. Person B accepts the offer in writing. Table 15 shows the acceptance of person B has a relation to a reference legal subject object 'WrittenAcceptence'.

Table 14 Person B accepts the offer for sale of car A

EventOrAction		**Subject1**		**Subject2**		**Object**
acceptsOffer	*actionOf*	PersonB	*towardsSubject*	CarA	*towardsObject*	OpenOfferCarA

Table 15 Person B accepts the offer in writing

ReferenceLegal SubjectObject		**Subject1**		**Object**
WrittenAcceptance	*referenceTo Subject*	PersonB	*ReferenceTo Object*	OpenOfferCarA

Figure 18 shows the action of person B to accept the offer of car A counts as a legal fact. The inference is correct, since the acceptance of the offer has a relationship with the same object and subject as the reference to legal subject object. We confirm the Legal Fact ODP infers events and actions with references to legal sources as legal facts.

Figure 18 Person B accepts car A is a legal fact

Conclusions of the evaluation

Two scenarios confirm the Legal Fact ODP infers events and actions as legal facts. We offer additional rules with the following conditions for events or actions to count as legal facts:

- The event or action has a reference to legal subject object;
- The legal subject object has a reference to an object;
- The legal subject object has a reference to a subject;
- The event or action has relationship to the same object as the reference source;
- The event or action has relationship to the same subject as the reference source.

Voorwinden uses multiple rules for determining an action is a legal fact (Voorwinden, 2018). For optimization, we limit the quantity of rules. We propose one rule with all conditions for inferring legal facts. This requires a rule with multiple triples. SWRL supports inferencing with combinations of triples. Therefore, we implement rules in the ODP with SWRL. The evaluation demonstrates the ODP and added rules support correct inferencing of legal facts.

In addition, we create a rule for inferring references to legal subject objects as objects. Van Engers and Nijssen do not provide guidelines for determining when an instance is an object, subject, or reference. We propose references to legal subjects objects with relationships to legal facts are objects in other scenarios. This gives certainty about the legality of the object. For future research can examine whether similar rules apply to subjects.

4.3. Temporal Legal Relations ODP

The Temporal Legal Relations ODP represents temporal legal relations based on Van Engers and Nijssen's Hohfeld semantic-conceptual model (Van Engers & Nijssen, 2014b).

Intent

The intent of the Temporal Legal Relations ODP is inferring the rights and corresponding duties. In addition, the ODP enables sequencing legal relations by starting a legal relation at the same time a preceding legal relation ends.

Competency Questions

The following competency questions are examples for understanding which questions the ODP can answer (Karima et al., 2016):

- *What kind of duty does party B have if party A has a Privilege regarding a certain matter?*
- *What is the creation date of this legal relation?*
- *What is the sequence of legal relations?*

4.3.1. Schema Diagram of Temporal Legal Relations ODP

We put Van Engers and Nijssen's pattern of legal relations in the Temporal Relation OPD by creating classes and object relations for the fact type: *<Party-Right-Side> in the role of <Kind-Of-Right> has a legal relation with <Party-Duty-Side> in the role of <Kind-Of-Duty> with respect to <Matter>* (Van Engers & Nijssen, 2014b). This includes the pattern for legal relations of Figure 19.

Figure 19 Legal Relation Pattern (Van Engers & Nijssen, 2014b)

For determining the sequence of temporal legal relations, Van Engers and Nijssen describe the following fact type: *The legal relation was established on <Date-Time-Established>, it starts on <Date-Time-Effectiveness-Start> and it has terminated on <Date-Time-Effectiveness-Ended>.* We implement the fact type with Semantic Web technologies. We add dates to temporal legal relations with data properties. Figure 20 shows the schema diagram of the Temporal Legal Relations ODP.

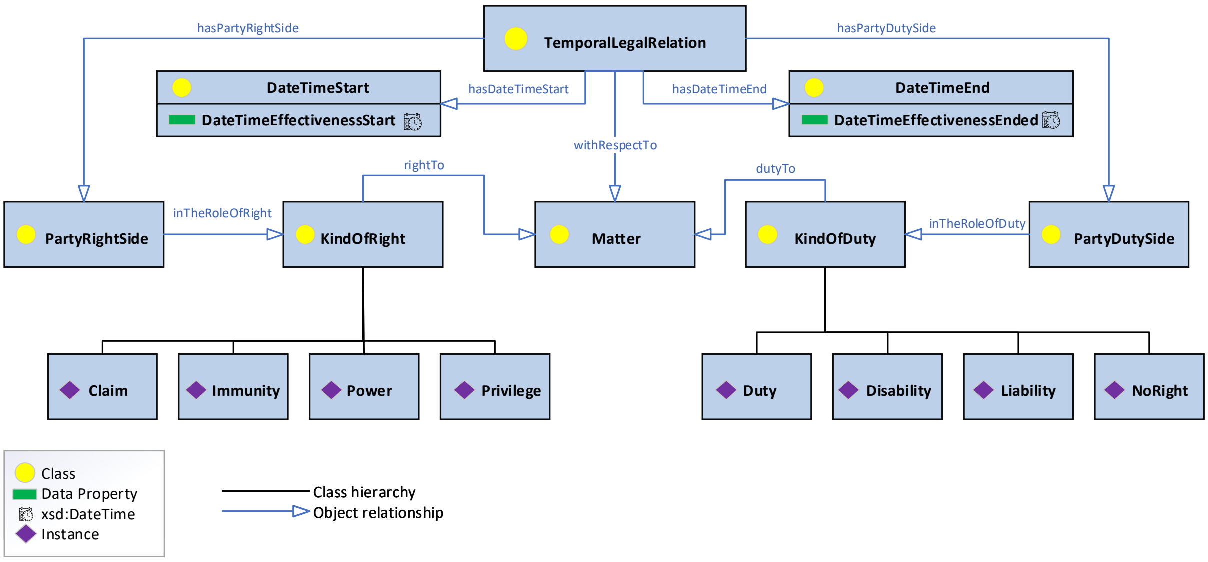

Figure 20 Temporal Legal Relations based on the fact type legal relation (Van Engers & Nijssen, 2014b)

Each temporal legal relation has a bearer and a counterpart (Agnoloni & Francesconi, 2011; De Klerk, 2018; Van Engers & Nijssen, 2014b). Similar to Van Engers and Nijssen's work, we refer to this as a party right side and a party duty side.

Van Engers and Nijssen describe permitted combinations of kind of rights and kind of duties, such as Claim and Duty. An integrity rule determines which kind of right corresponds with which kind of duti: *the combination Party-Right-Side, Kind-Of-Right, Party-Duty-Side and Matter determines the Kind-Of-Duty (Van Engers & Nijssen, 2014b).*

We apply Horn clauses for inferring the combination of Party-Right-Side, Kind-Of-Right, Party-Duty-Side and Matter leads to a specific Kind-Of-Duty. Horn clause 1 and Horn clause 2 illustrate the Horn clauses for Privilege-NoRight and Claim-Duty legal relations.

Horn clause 1 A party right side Privilege to a matter has a party duty side NoRight to the same matter

```
Party-Right-Side ^ Kind-Of-Right:Privilege ^ Party-Duty-Side ^ Matter -> Kind-Of-Duty:NoRight
```

Horn clause 2 A party right side Claim to a matter has a party duty side Duty to the same matter

```
Party-Right-Side ^ Kind-Of-Right:Claim ^ Party-Duty-Side ^ Matter -> Kind-Of-Duty:Duty
```

We put the Horn clauses' in the ODP with the Semantic Web technology SWRL. SWRL enables using Horn clauses within the Semantic Web. Each permitted combination of 'kind or right' and 'kind of duty' has an associated SWRL rule. Code segment 8 gives an example for inferring 'party right side' has a Privilege with respect to a certain matter means 'party duty side' has a NoRight regarding the same matter. Within the ODP, we provide similar SWRL rules for the other permitted legal relations and their associated rights and duties.

Code segment 8 Infer the party duty side has a NoRight when party right side has a Privilege

```
LegalRelation(?lr) ^ hasPartyRightSide(?lr, ?prs) ^ inTheRoleOfPrivilege(?prs, ?rop) ^
hasPartyDutySide(?lr, ?pds) ^ Matter(?m) ^ withRespectTo(?lr, ?m) ^ rightTo(?rop, ?m) ->
inTheRoleOfNoRight(?pds, ?m)
```

4.3.1. Evaluation of the Temporal Legal Relations ODP

Examples from car sale scenario

We use two scenario examples from the car sale scenario for evaluating the Temporal Legal Relations ODP (Van Engers & Nijssen, 2014a). In the car sale scenario, Van Engers and Nijssen include temporal legal relations. Each period or transition within the scenario has a reference to one or more legal relations. Several temporal legal relations are active during more than one period.

In the first scenario, we demonstrate a Power-Liability legal relation that starts in P1 and ends in T1. During this scenario, the roles of person A and person B exchange. First person A has a Power to offer the car for sale, then person B has a Power to accept the offer. We sequence the temporal legal relations. The Power of person B starts when the Power of person A ends.

In the second scenario, person A has a privilege to use the car and person B has a NoRight. They exchange roles when person B buys the car from person A. The Privilege of person B starts at the same time this Privilege ends for person A.

Scenario 1: Power – Liability P1 and T1

Step 1: In this step, person A has a Power to offer the car for sale. This corresponds with the period P1 of the car sale scenario. The instances of Table 16 are enough for inferring the rights and duties. Due to the added rules such as Code segment 8, knowing the kind of right of person A is sufficient for inferring the kind of duty of person B.

Table 16 Instances that determine person A has a power towards person B

Party-Right-Side	Kind-Of-Right	Party-Duty-Side	Matter
PersonA	Power	Person B	offersForSaleCar

Power – Liability is a permitted legal relation. Therefore, person B has a Liability towards person A to offer the car for sale. Figure 21 shows person B has a Liability with respect to the matter 'offersForSaleCar'.

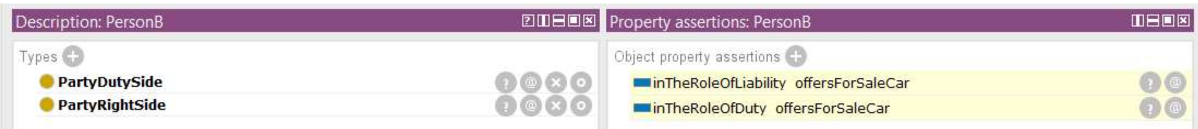

Figure 21 Person B has the kind of duty Liability toward the matter that person A offer the car for sale

Step 2: In the next step of T1, person A offers the car for sale and creates a Liability for herself. The roles are now reversed. The instances from Table 17 determine person A has a Liability to person B towards the matter 'acceptsOffer'.

Table 17 Instances for person B has a Power to accept the offer

Party-Right-Side	Kind-Of-Right	Party-Duty-Side	Kind-Of-Duty	Matter
Person B	Power	PersonA	Liability	acceptsOffer

Van Engers and Nijssen provide an integrity rule for inferring the kind of duty for a given right. In this part of the scenario, we know person A has a duty and infer the kind of right of person B. We extend Van Engers and Nijssen's work with the following integrity rule for inferring the kind of right for a given duty: *The combination Party-Duty-Side, Kind-Of-Duty, Party-Right-Side and Matter determines the Kind-Of-Right.* Code segment 9 shows the corresponding SWRL rule.

Code segment 9 Infer the party right side has a Power when party duty side has a Liability

```
withRespectTo(?lr, ?m) ^ hasPartyDutySide(?lr, ?pds) ^ hasPartyRightSide(?lr, ?prs) ^
inTheRoleOfLiability(?pds, ?rol) ^ dutyTo(?rol, ?m) -> inTheRoleOfPower(?prs, ?m)
```

Figure 22 shows person B has both a Liability to 'offerForSaleCar' and a Power to 'acceptOffer'. Person B has a duty and a right at the same time. In some scenarios, it is desirable a person has roles as 'party duty side' and 'party right side'. For example, a person has the right to receive income and a duty to pay taxes at the same time. However, in the car sale scenario, there is a sequence of legal relations. Person B loses his Liability as soon as he gains a Power. Therefore, we add dates to the legal relations and determine certain rights or duties start as soon as another legal relation ends.

Figure 22 Person B has a right and a duty at the same time

Step 3: In the third step, person B no longer has a Liability as soon as the car is for sale. Person A no longer has the Power to offer the car for sale, since she has already used this power. In the car sale scenario, the legal relations replace each other at the end of P1 and start of T1. We add fictional dates of Table 18 to the ODP for determining T1 starts when P1 ends.

Table 18 Date instances for the Power-Liability temporal legal relations in P1 and T1.

TemporalLegalRelation	DateTimeEffectivenessStart	DateTimeEffectivenessEnded
P1_Power-Liability	01-03-2017	01-04-2018
T1_Power-Liability	P1 DateTimeEffectiveness Ended	

We add code segment 10 for determining the end date of temporal legal relations P1_Power-Liability is the start date of the temporal legal relation T1_Power-Liability.

Code segment 10 The end of P1 starts T1

```
TemporalLegalRelation(P1_Power-Liability) ^ hasDateTimeEnd(P1_Power-Liability, ?dte) ^
DateTimeEffectivenessEnded(?dte, ?ee) ^ TemporalLegalRelation(T1_Power-Liability) ^
hasDateTimeStart(T1_Power-Liability, ?dts) -> DateTimeEffectivenessStart(?dts, ?ee)
```

Figure 23 shows the legal relation T1_Power-Liability starts at the date of 'DateTimeEffectivenessEnded' of P1. We confirm the ODP enables sequencing legal relations.

Figure 23 De legal relation T1_Power-Liability starts at the date the legal relation P1_Power-Liability ends

Scenario 2: Privilege – NoRight

Step 1: During this step and P1 of the car sale scenario, person A owns the car and has a Privilege to use the car. Table 19 shows the instances we put in the ODP for indicating person A has Privilege with respect to the matter ‘useTheCar’. Knowing the kind of right of person A is sufficient for inferring the kind of duty of person B.

Table 19 Instances person A has a Privilege to use the car

TemporalLegalRelation	Party-Right-Side	Kind-Of-Right	Party-Duty-Side	Matter
P1_Privilege-NoRight	Person A	Privilege	Person B	useTheCar

Privilege-NoRight is a permitted legal relation. Figure 24 displays person B has a NoRight to use the car.

Figure 24 Person B has a NoRight to use the car.

Step 2: In this step, person A gives the car to person B for the exchange of €20.000,-. This corresponds with transition T4 of the car sale scenario. Person A no longer has a Privilege to use the car and person B no longer has a NoRight. The end date from Table 20 indicates when the legal relation ‘P1_Privilege-NoRight’ ends.

Table 20 Start and end date for the legal relation P1_Privilege-NoRight

TemporalLegalRelation		DateTime Effectiveness Start		DateTime Effectiveness Ended
P1_Privilege-NoRight	*hasDateTimeStart*	StartP1: 01-03-2017	*hasDateTimeEnd*	EndP1: 01-04-2018

Step 3: In this step, person A and person B exchange the car and the money. At the same time, they exchange roles. Person B now has a Privilege to use the car. We already know this means the party duty side, now person A, has a NoRight. We put the instances from Table 21 in the ODP to give person B a Privilege.

Table 21 Instances that person B has a Privilege to use the car

TemporalLegalRelation	Party-Right-Side	Kind-Of-Right	Party-Duty-Side	Matter
T4_Privilege-NoRight	Person B	Privilege	Person A	useTheCar

The legal relation T4_Privilege-NoRight starts as soon as person B’s NoRight to use the car ends. This is at the end of the legal relation P1_Privilege-NoRight. We use code segment 11 for inferring the start date of T4 is the same as the end date of P1.

Code segment 11 The end of person A’s Privilege starts the beginning of person B’s Privilege

```
TemporalLegalRelation(P1_Privilege-NoRight) ^ hasDateTimeEnd(P1_Privilege-NoRight, ?dte) ^
DateTimeEffectivenessEnded(?dte, ?ee) ^ TemporalLegalRelation(T4_Privilege-NoRight) ->
DateTimeEffectivenessStart(T4_Privilege-NoRight, ?ee)
```

The end date of legal relation P1_Privilege-NoRight is 01-04-2018. Figure 25 shows T4 starts at the same date. We validate the legal relation T4_Privilege-NoRight starts at the same time P1_Privilege-NoRight ends. Person B has a Privilege to use the car as soon as person A loses that Privilege.

Figure 25 The legal relation in time span T4 has the same start date as the end of P1

During T4, person B no longer has a NoRight. Instead, he has a Privilege to use the car. However, exiting knowledge in Semantic Web ontologies cannot be removed or modified. As figure 26 shows, person B has a NoRight and a Privilege to use the car.

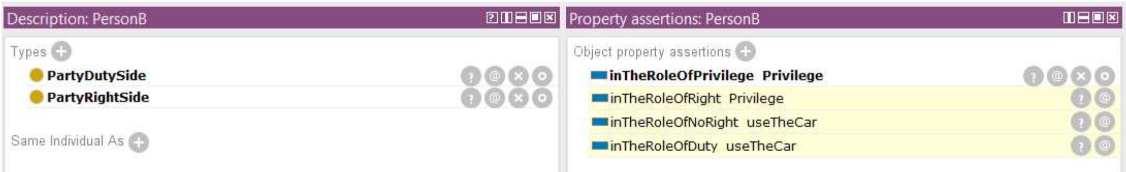

Figure 26 Person B has a Privilege and a NoRight to use the car

Step 4: In the last step of this scenario, we infer which temporal legal relations are active. A person cannot have a Privilege and a NoRight towards the same matter at the same time. Because existing knowledge cannot be removed from the ontology, we provide code segment 12 for inferring which of the two legal relations is currently active. Legal relations with an end date after today and a start date before today are active legal relations. Code segment 12 enables inferring whether person B currently has a NoRight or a Privilege to use the car.

Code segment 12 Infer active temporal legal relations

```
TemporalLegalRelation(?tlr) ^ hasDateTimeStart(?trl, ?dts) ^ DateTimeEffectivenessStart(?dts,
?des) ^ temporal:before(?des, "now") ^ hasDateTimeEnd(?tlr, ?dte) ^
DateTimeEffectivenessEnded(?dte, ?dee) ^ temporal:after(?dee, "now") ->
ActiveTemporalLegalRelation(?tlr)
```

Figure 27 shows T4_Privilege-NoRight is an active temporal legal relation. At the time of this inference, person B has a Privilege and person A has a NoRight to use the car. The temporal legal relation in which person A has a Privilege to use the car is not active. This means the legal relationship P1_Privilege-NoRight is no longer valid at this time.

This also means conclusions can change over time. The end date of the temporal legal relation T4_Privilege-NoRight is 01-04-2019. After this date, this legal relationship is no longer inferred as an active legal relation by the ODP. We confirm the Temporal Legal Relations ODP infers which temporal legal relations are active at a certain date.

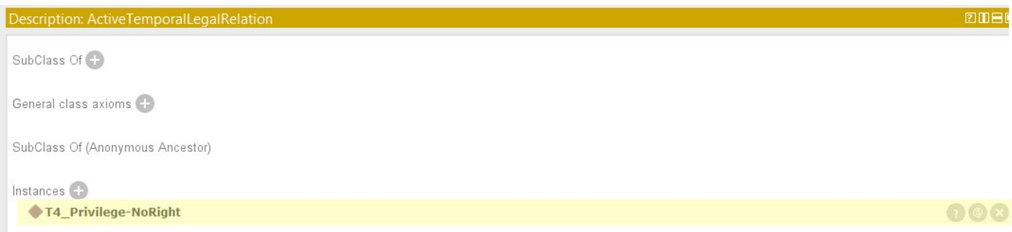

Figure 27 T4_Privilege-NoRight is an active legal relation at the time of the inference

Conclusions of the evaluation

Two scenarios demonstrate the Temporal Legal Relations ODP sequences legal relations. The end of a legal relation is the start of another legal relation. We use RDFS, OWL, and SWRL that supports monotonic reasoning. Added knowledge remains within the Semantic Web ontology. This makes ending legal relations challenging. Existing legal relations in the ontology cannot be removed or modified. Because of this, legal relations still exist in the ontology after the end date. Conflicting legal relations may occur, such as a person with both a duty to do something and the right not to do so.

We provide SWRL rules for inferring which legal relations are active at a certain moment. In the evaluation, we confirm which legal relation is active at the time of inferencing. By changing "now" in code segment 12 into a different date, the ODP infers which temporal legal relations are active at the time of that specific date.

Within the Semantic Web domain, it is the norm that OWL ontologies cannot make decisions that could change if more information becomes available. However, the Temporal Legal Relations ODP inferences active legal relations that can change over time. The ODP may infer a temporal legal relation as active today and not infer that same legal relation as active after the end date. This is unusual for people who are accustomed to OWA and important aspect for consideration while using the Temporal Legal Relations ODP.

5. Conclusion and Recommendations

5.1. Conclusion

The aim of this work is examining how, and to what extent, the Hohfeld semantic-conceptual model of Van Engers and Nijssen can be implemented in Semantic Web ODPs.

This paper demonstrates a Semantic Web implementation of the Hohfeld semantic-conceptual model in ODPs. We provide OWL files for the ODPs. We offer SWRL code segments for temporal reasoning, qualifying legal facts, sequencing temporal legal relations, and inferring rights and duties. We evaluate the ODPs by implementing examples from the car sale scenario of Van Engers and Nijssen. Other scenarios may require additional SWRL rules.

We confirm the *Temporal Information ODP* facilitates temporal reasoning. The ODP utilizes Allen's intervals and supports SWRL rules for determining certain events take place before or after other events. We demonstrate the ODPs capabilities for sequencing events. We add a predetermined interval and infer which events take place within this duration. We manually put dates in the ODP and the SWRL rules. Knowledge of SWRL is required for the use of the ODP. Within the examples, we use explicit time units. Uncertain time expressions are underexposed. This work offers no possibilities for expressing implicit time units, such as a 'little earlier' or 'soon'.

The *Legal Fact ODP* implements Van Engers and Nijssen's qualification process and infers actions and events as legal facts. For determining which events or actions count as legal facts, a combination of triples is required. Legal facts have references to reference sources, subjects, and objects. We use SWRL for combining triples. For optimization, we propose a single rule with all triples for inferring legal facts. The evaluation demonstrates the ODP and this rule supports correct inferencing of legal facts.

The *Temporal Legal Relations ODP* provides start and end dates of legal relations for determining the sequence of temporal legal relations. We confirm the end of one legal relation can be the start of another legal relation. However, we observe terminating legal relations is challenging because of the monotonic reasoning of Semantic Web ontologies. Existing knowledge cannot be modified or removed. This means once a legal relation is established in the ODP, we cannot remove or change it. Therefore, we infer which temporal legal relations are active at the time of inferencing. As a result, knowledge within the ODP can change over time. The ODP infers legal relations as active before the end date of that legal relation. After the end date of the legal relation, the ODP will no longer infer the legal relations as active. The evaluation confirms the ODP infers which temporal legal relations are active at the given date.

This work builds on earlier work at the Open University concerning Hohfeld legal relations and the Semantic Web. Slootweg and De Klerk implement the Hohfeld legal relations with Semantic Web technologies and enable determining which combinations of rights and duties are allowed. We extend their work by inferring the kind of duties belonging to a certain kind of rights. The *Temporal Legal Relations ODP* infers the kind of duty a party has towards a certain matter if the opposite party has a certain right. As a result, it is enough to know the rights of one party for inferring the duties of the other party.

5.2. Recommendation for practical use

We provide OWL files including SWRL rules for the ODPs of this work. Users can add new rules to the OPDs as desired, as long as the rules are not contradictory. For example, the Legal Fact ODP contains a rule for inferring references to legal sources as objects. A similar rule is imaginable for subjects. For optimization, we keep the quantity of SWRL rules in the ODPs as low as possible. We have not done any research on the consequences of adding more rules.

Contrary to customary practice in the Semantic Web domain, conclusions of inferencing with the Temporal Legal Relations ODP may change over time. The ODP infers a temporal legal relation as an active relation on the date of inferencing. The date of inferencing is changeable to any other data for inferring which legal relations are active on that particular date. In the current state, the ODP infers which relationships are active at the time of inferencing. This means the ODP may not draw the same conclusions when conducted on another date. It is important to take this into account during practical use.

5.3. Recommendation for future research

In this work, we offer ODPs for the Hohfeld semantic-conceptual model of Van Engers and Nijssen. The Temporal Information ODP enables temporal reasoning with numbers as dates. It is interesting to examine how, and to what extent, the ODP facilitates other types of reasoning. Future research can study the ODPs facilities for other units, such as money and financial transactions.

For optimization, we limit the amount of SWRL rules in the ODPs. For example, we propose one SWRL rule with a combination of several triples for inferring legal facts. Another possibility is adding multiple SWRL rules with fewer triples per rule. In our examples, legal facts have references to subjects, objects, and reference sources. Nonetheless, in certain scenarios a reference to a subject is sufficient. Future research can investigate the different consequences of multiple rules with few triples or fewer rules with more triples

Future research can examine which connections between the ODPs are possible and desirable. For example, importing the inferred legal facts from the Legal Fact ODP as matter in the Temporal Legal Relation ODP. We add start and end dates of legal relations manually. Future research can determine potential of importing dates from the Temporal Information ODP.

References

Agnoloni, T., & Francesconi, E. (2011). *Modelling semantic profiles in legislative documents for enhanced norm accessibility*.

Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Commun. ACM, 26*(11), 832-843. doi:10.1145/182.358434

Allen, L. E., & Saxon, C. S. (1995). *Better language, better thought, better communication: the A-Hohfeld language for legal analysis.* Paper presented at the Proceedings of the 5th international conference on Artificial intelligence and law.

Antoniou, G., & Van Harmelen, F. . (2008). *A semantic web primer*: MIT press.

Batsakis, S., Petrakis, E., Tachmazidis, I., & Antoniou, G. (2016). Temporal Representation and Reasoning in OWL 2. *Semantic Web*, 1-21.

Berners-Lee, T., Hendler, J., & Lassila, O. . (2001). The Semantic Web. *Scientific American*, 34–43.

Bos, P. (2013). *Bedrijfsregels in verschillende vormen-Een vergelijking op toepasbaarheid tussen SWRL en Relatie algebra bij wetteksten.* Open Universiteit Nederland,

Casanovas, P., González-Conejero, J., & Koker, L. D. (2017). Legal Compliance by Design (LCbD) and through Design (LCtD): Preliminary Survey.

De Klerk, S. C. M. (2018). *Semantic Web Content Ontology Design Patterns for A-Hohfeld Legal Relations.* Open University, Utrecht.

Dulfer, D. (2014). Gestructureerd Nederlands voor feiten en regels.

Fowler, M. (1997). *Analysis Patterns, reusable object models*. Massachusetts: Addison-Wesley.

Grüninger, M., & Li, Z. (2017). *The Time Ontology of Allen's Interval Algebra*. Paper presented at the 24th International Symposium on Temporal Representation and Reasoning (TIME 2017) http://drops.dagstuhl.de/opus/volltexte/2017/7927/

Hammar, K. (2017). *Content Ontology Design Patterns: Qualities, Methods, and Tools* (Vol. 1879): Linköping University Electronic Press.

Hobbs, J. R., & Pan, F. (2017). Time Ontology in OWL
W3C Recommendation 19 October 2017. Retrieved from https://www.w3.org/TR/owl-time/

Hohfeld, W. N. (1917). Fundamental Legal Conceptions as Applied in Judicial Reasoning. *Faculty Scholarship Series, 4378*.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Said Tabet, M., Grosof, B., & Dean, M. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Retrieved from https://www.w3.org/Submission/SWRL/

Hulstijn, J. (2012). Compliance by Design. Het inbouwen van regelgeving in bedrijfsprocessen en informatiesystemen. *Regelmaat*.

Karima, N., Hammar, K., & Hitzler, P. (2016). How to document ontology design patterns.

Knublauch H., F. R. W., Noy N.F., Musen M.A. . (2004). The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. *Lecture Notes in Computer Science, 3298*.

Lalmohamed, A. (2014). *Expressing Hohfeldian legal concepts, traceability and ambiguity with a relation algebra-based information system.* Open Universiteit Nederland,

Métais, E., Ghorbelb, F., Hamdia, F., Ellouzeb, N., Herradia, N., & Soukanec, A. (2018). Representing Imprecise Time Intervals in OWL 2. *International Journal of Conceptual Modeling*(Special Issue on Conceptual Modelling in Honour of Heinrich C. Mayr), 120-132.

Musen, M. A. (2015). The Protégé Project: A Look Back and a Look Forward. *AI matters, 1*(4), 4-12. doi:10.1145/2757001.2757003

Musen, M. A., & Protégé, T. (2015). The Protégé Project: A Look Back and a Look Forward. *AI matters, 1*(4), 4-12. doi:10.1145/2757001.2757003

O'Connor, M. J., Shankar, R. D., Nyulas, C., Das, A. K., & Musen, M. A. (2008). *The SWRLAPI: A Development Environment for Working with SWRL Rules*. Paper presented at the OWL: Experiences and Directions (OWLED), 4th International Workshop Washington, D.C., U.S.A

O'Connor, M. J. (2017). SWRL Temporal Ontology Release 2.0.5. Retrieved from http://swrl.stanford.edu/ontologies/built-ins/3.3/temporal.owl

O'Connor, M. J., & Das, A. K. (2011). *A Method for Representing and Querying Temporal Information in OWL*.

Pandey, R., & Dwivedi, S. (2011). Ontology Description using OWL to Support Semantic Web Applications. *International Journal of Computer Applications, 14*(4), 30-33.

RedHat. (2018). Drools rule engine.

Saunders, M., Lewis, P., & Thornhill, A. (2015). Methoden en technieken van onderzoek.

Schlag, P. (2015). How To Do Things With Hohfeld. *Law and Contemporary Problems*, 185-234.

Shadbolt, N., Berners-Lee, T., & Hall, W. (2006). The Semantic Web Revisited. *IEEE Intelligent Systems, 21*(3), 96-101. doi:10.1109/mis.2006.62

Slootweg, P., Rutledge, L., Wedemeijer, L., & Joosten, S. (2016). The Implementation of Hohfeldian Legal Concepts with Semantic Web Technologies. *AI4J – Artificial Intelligence for Justice*, 65-72.

Snodgrass, R. T. (1992). *Temporal databases*, Berlin, Heidelberg.

Tao, C., Solbrig, H. R., Sharma, D. K., Wei, W.-Q., Savova, G. K., & Chute, C. G. (2010). *Time-Oriented Question Answering from Clinical Narratives Using Semantic-Web Techniques*, Berlin, Heidelberg.

Trokanas, N. (2015). *Practical Issues of Ontology Design, Evaluation and Reuse: A Case Study in Chemical and Process Engineering.* University of Surrey.

Van Engers, T., & Nijssen, S. (2014a). *Bridging Social Reality with Rules*. Paper presented at the Internationales Rechts Informatik Symposion, Salzburg, Austria.

Van Engers, T., & Nijssen, S. (2014b). *Connecting people: semantic-conceptual modeling for laws and regulations.* Paper presented at the International Conference on Electronic Government.

Voorwinden, H. (2018). *Legal Norm from Structured Dutch to Relation Algebra.The bridge between law and business rules.* Open University, Utrecht.