# The Implementation of Hohfeldian Legal Concepts with Semantic Web Technologies

**Pieter Slootweg** and **Lloyd Rutledge** and **Lex Wedemeijer** and **Stef Joosten**[1]

**Abstract.** This research explores how and to what extent Semantic Web techniques can implement Hohfeldian legal concepts. Laws and regulations are forms of rules in natural language. Because laws are objective and formal, they are suitable for specification with formal logic. Hohfeldian legal concepts are an important tool for the explicit creation of normative legal relationships between the parties. The results of this study show that it is possible for legal requirements based on Hohfeldian legal concepts to be expressed with Semantic Web techniques. For the different Hohfeldian legal concepts, we work out a generic solution within a case study. This work shows that global qualification regarding whether or not a particular action is allowed should not only be determined on the basis of the relevant Hohfeldian legal concepts, but also by taking conditional statements into account.

## 1 INTRODUCTION

The idea of applying logic to laws and regulations is not new. For some time, scientists have explored the possibilities of deriving legal decisions from legal sources, just as with logical deduction, a conclusion is derived from a set of axioms. However, creating requirements compliant with laws and regulations is difficult. This complexity arises because articles of law are sometimes complementary, overlapping and contradictory.

One method for finding a solution for the difficult task of specifying requirements in legal texts is to focus on the legal norms in the text. Deontic logic is an important way of formally describing these legal norms. Hohfeldian legal concepts constitute a further refinement of the concepts of deontic logic [14]. The primary purpose of Hohfeld’s work is to make the normative legal relationships between parties explicit. Hofheldian legal concepts are used in different studies for extracting requirements that are compliant with legal texts. Important examples are Production Rule Modeling (PRM) [16] and the Nomos Framework [20].

A relatively new domain for the implementation of legislation is the Semantic Web. The aim of this study is to investigate how and to what extent Semantic Web techniques can be used to model legal texts with Hohfeldian legal concepts. This work also focuses on the modeling of pre- and post-conditions and exceptions within legal text. The case study we use is the Health Insurance Portability and Accountability Act (HIPAA) [22], partly because HIPAA is also used in several other relevant studies.

This research builds on previous research at the Open University in the Netherlands regarding processing legal texts with formal logic. Bos has done research on the implementation of rules with Semantic Web technologies [7]. Lalmohamed implemented Hohfeldian legal concepts with relation algebra [15]. This relation algebra implementation is a reference for modeling the rules in our study. In comparing the Semantic Web with relation algebra, the main concerns are the open and closed world assumptions and negation as failure.

Francesconi investigated the use of Hofheldian legal concepts based on Semantic Web technologies for the semantic annotation of legal texts [10]. The focus of this research is the application of Hohfeldian legal concepts to the normative qualification of several legal cases within the context of a particular law. The empirical research in our study explores how to use Semantic Web techniques to draw normative conclusions with Hohfeldian legal concepts.

We reuse an existing ontology for modeling Hohfeldian legal concepts - the Provision Model - by extending where necessary for our purposes with our new ontology: HohfeldSW. This is to complete missing Hohfeldian legal concepts and to implement normative qualification. In addition, a domain-specific ontology is elaborated: the HIPAA ontology. Requirements were extracted with normative phrase analysis based on the PRM Method. Our implementation also applied some ontology design patterns such as n-ary relations [18] and AgentRole [19]. These ontology design patterns are of added value to the transparency of the implementation.

The results of our empirical study show that it is possible to express legal requirements based on Hohfeldian legal concepts with Semantic Web techniques. The implementation makes the relationship between actors clear, along with the actions they perform, what the legal consequences are, and if they may or may not perform these actions. With our implementation, it is possible to implement generic rules for validating the various legal concepts.

## 2 RELATED WORK

### 2.1 Hohfeldian Legal Concepts

Deontic logic is used to analyze the normative structures and normative reasoning that occur in laws [27]. Deontic logic is formal logic used to reason with ideal and actual behavior: what should be the case, or what should be done by those involved [25]. Deontic logic is developed as modal predicate logic with operators for obligation (O), permission (P) and prohibition (F).

The Hohfeldian legal concepts constitute a further refinement of the concepts of deontic logic [14]. With the aid of the Hohfeldian legal concepts, it is possible to derive the most important legal norms from a text. Hohfeld has developed an analytical scheme in which he distinguishes four categories of legal relations between the parties. He also elaborates on legal differences between the different legal positions [5]. In his view there are eight such entities. On one hand, there are Right, Privilege, Power and

[1] Faculty of Science, Management and Technology, Open University in the Netherlands, Heerlen, The Netherlands, email: Lloyd.Rutledge@ou.nl

Immunity. In addition, there are the correlated entities: Duty, No-right, Liability and Disability.

## 2.2 Implementations of Hohfeld

Hohfeld's study was widely applied and marked the beginning of a systematic approach. However, this was not enough for a formal theory and a base for the implementation of information systems. Allen and Saxon developed the Hohfeldian legal concepts further into a model in which deontic norm structures could be represented: the A-HOHFELD legal concepts [1]. Allen and Saxon showed in their work how the framework of Hohfeldian legal concepts could be used to define a formal language, which makes it possible to precisely analyze a legal act, thus removing ambiguity.

There are several studies where the Hohfeldian legal concepts are used as a tool to specify legal requirements. Well-known examples include the Nomos framework [20] and the PRM (Production Rule Methodology) [16]. Siena and other researchers developed the Nomos framework to support the requirements analyst in drafting requirements that are compliant with legislation [20]. North Carolina State University focused on the use of formal methods to model legislation. Their focus was on modeling legislation and methods to systematically analyze legal texts. This resulted in the PRM. From the perspective of the PRM, relevant legal concepts are inferred from the words that are used in the normative phrases. Each legal concept also has an implied concept. For example, when a person has a right to a notification made by a hospital, it implies a duty for that hospital to send a notification. The added value of Hohfeld's theory is that implicit assumptions and consequences are made explicit.

Francesconi's model is developed for legislative provisions with axioms from RDFS and OWL [10]. His research makes design patterns with OWL-DL techniques to implement the Hohfeldian legal relationships. The outcome of his research is primarily intended to make a useful contribution to the refinement of semantic annotations to legal texts. The focus of our research is the application of Hohfeldian legal concepts to the normative qualification of various legal cases. We explore the feasibility of this within the context of a specific law: HIPAA [22].

## 2.3 Law and the Semantic Web

There is much research on the implementation of legislation with Semantic Web technologies. In particular, research on legal ontologies combined with the extraction of semantic standards based on Natural Language Processing (NLP) has given a strong impetus to the modeling of legal concepts [9]. Benjamins has developed a wide variety of ontologies with a wide variety of applications [3]. One demonstration of the importance of legal ontologies is the missing link between AI & Law and Legal Theory [23]. Ontologies for the legal domain are useful in applications such as organizing and structuring information, semantic indexing, integration, reasoning and problem solving.

This research focuses on the application of rules on legal texts, or reasoning and problem solving. Ontologies can thereby be used as a terminology part of a knowledge database in order to derive assertions from the problem to be solved. The role of an ontology in this situation is the representation of domain knowledge so that an automatic logic-reasoning mechanism can represent problems and possibly generate solutions to these problems. Design choices when constructing an ontology are strongly influenced by the ontology's purpose. How knowledge is structured and formalized in the ontology depends on how it is used by the reasoning logic to draw the desired conclusion. The reasoning context limits its reusability in the ontology. This phenomenon is known as inference bias [24]. Inference bias is unavoidable because no wording is completely neutral.

We now present some concrete examples of research on legal ontologies. Wyner developed an ontology in OWL called Legal Case-Based Reasoning (LCBR) [28]. The Leibniz Institute of Law has done extensive research into the development of ontologies for the legal domain. An important ontology in this case is FOLaw (Functional Ontology for Law) [6]. FOLaw specifies functional dependencies between different types of knowledge that are important for legal reasoning. Although FOLaw is an important source for a number of ontologies and legal reasoning systems in various research projects, it is more an epistemological framework than a core ontology. Another important ontology is LKIF, which consists of a core legal ontology and a legal rule language, which makes it possible to represent legal knowledge in detail and reason about it [12]. Other relevant ontologies include Fundamental Legal Conceptions, A-Hohfeld, Language for Legal Discourse, Frame-Based Ontology of Law, LRI-Core [6] and the Core Legal Ontology [11].

Another important development in this context is LegalRuleML. The Technical Committee of OASIS (Advancing Open Standards for the Information Society) developed a rule interchange language for the legal domain. This makes it possible to structure the content of a legal text into machine-readable format, which can be used as a source for further steps such as control and data exchange, comparison, evaluation and reasoning. An important goal in the development of Legal Rule modeling is to bridge the gap between descriptions of natural language and semantic modeling standards [2]. Another important object is to provide an expressive XML standard for modeling of normative rules which comply with requirements from the legal domain. This makes it possible to introduce a legal reasoning layer on top of the ontology.

There are important similarities between LegalRuleML and SBVR (Semantics of Business Vocabularies and Business Rules). We mention SBVR because this is also an important language for specifying rules with Semantic Web technologies. With SBVR concepts, definitions, rules and facts can be expressed in natural language, similar to LegalRuleML. SBVR involves business rules that may or may not have legal significance. LegalRuleML refers to expressions that have legal significance, in particular legal concepts and processes. Distinctive for LegalRuleML are the possibility of defeasibility and the various possibilities for expressing deontic concepts.

## 2.4 Semantic Web Ontologies for Law

Our study selected the Provision Model [10]. While this ontology is still in development - only some of the Hohfeldian legal concepts are implemented - it is a good basis for our study. This is substantiated by a number of relevant criteria. The Provision Model is implemented transparently. The Provision Model is not only available as an OWL ontology, but is also explained in the aforementioned publication. One of the objectives of the Provision Model is supporting reasoning by making use of normative rules based on Hofheldian legal concepts. The focus is on the derivation of implicit knowledge from explicitly recorded knowledge. The Provision Model is not focused on a specific legal domain, making the risk of misapplication outside the original context limited. The Provision Model meets the criteria for reusability and extensibility because the ontology is specific enough to be reused and, on the other hand, is not too specific so that reuse is impossible. We choose the Provision Model over LKIF-Core [12] because of the extents of the ontologies and because Hohfeldian legal concepts are not supported directly by LKIF. However, this ontology is a source of inspiration for qualifying legal standards.

## 3 CONCEPTUAL MODEL

The implementation of Semantic Web technologies is based on three ontologies. The Provision Model [10], based on Hohfeldian legal concepts, is used as a basis. As an extension of this, we designed our own ontology: HohfeldSW. We also developed a domain-specific ontology in OWL, based on the HIPAA Privacy Rule. In the implementation also a number of ontology design patterns are used: AgentRole and n-ary Relations.

### 3.1 Provision Model

According to Biagioli, legislation can be viewed as a set of ‘provisions’ (rules) based on speech acts, or more specifically, sentences to which meaning is assigned [4]. A legal text can be viewed from two perspectives on this basis:

1. *Structural or formal perspective*. This is consistent with the traditional classification of a legal text into chapters, articles, and paragraphs.
2. *Semantic perspective*. This is a specific representation of the legal text on the basis of the essential meaning of this text. A possible description can be given in terms of legislative provisions.

From these points of view, components of the legal text are, on one hand, sentences, paragraphs or articles, and on the other hand, provisions, focusing on the semantics. The focus in this study is on the latter. The Provision Model created a division between provision types and related attributes. Examples of types of provision are familiar terms as Duty, Right, Power and Liability. Examples of attributes are Bearer and Counterpart.

In the Provision Model, provision types are divided into two main categories: Rules, and Rules on Rules [10]. The rules of the underlying legal concepts are divided into constitutive and regulatory rules. Rules on rules involve different types of amendments to rules.

The Provision Model extends the standard Hohfeldian legal concepts by making a distinction between implicit and explicit provisions. This comes from the observation that sometimes legal texts mention legal concepts explicitly, but not related correlative legal concepts. For example, a text may explicitly mention a Duty but not a Right. In fact, in a different view of the duty itself, the rollers Bearer and Counterpart can be swapped. An OWL disjoint prevents a concept like Right from being both implicit and explicit.

### 3.2 HohfeldSW Ontology

The HohfeldSW ontology is our extension on the Provision Model. It introduces a few Hohfeldian legal concepts that are missing in the Provision Model: Privilege-NoRight and Immunity-Disability. Also, SWRL rules have been added for the validation of combinations of pairs Hohfeldian legal concept. We also introduced the concept of qualification. One of the main tasks within the legal domain is applying a particular law in a particular case. It must be established whether or not a particular case is allowed based on the relevant legal norms implemented in the system.

For the cases in which one of the concepts from a specific legal concept Hohfeldian pair is missing, we will have to evaluate whether a particular action is compliant with HIPAA. We implement these cases with SPARQL. We have also integrated the AgentRole [19] pattern in the HohfeldSW ontology.

The AgentRole pattern lets us make claims about the roles of agents without affecting the agents that fulfill these roles. In the HohfeldSW ontology, a stakeholder (agent) plays the role of both Actor and Counterpart. These roles can be coupled via the hasRole object property to a specific individual. The AgentRole pattern is applied to the roles that occur within the HIPAA ontology, such as Covered Entity, Government and Person.

### 3.3 HIPAA Ontology

The concepts in the HIPAA ontology are filled based on a normative phrase analysis for part of the HIPAA Privacy Rule, based on PRM. Each generic HIPAA Action is elaborated in the form of a conjunction of conditions, which together provide a description of the situation that is associated with that specific HIPAA Action.

In this study, each phrase has a normative Actor (Bearer), a Counterpart, an Action and an Object. Any Action from the HIPAA is linked to Hohfeld legal concept of the Provision Model of HohfeldSW. This is possible because for each legal concept of the Provision Model / HohfeldSW a related “hasBearer ‘and’ hasCounterpart” object property is available.

In line with research at the Leibniz Center for Law, a norm can be defined as a set of conditions in conjunctive normal form [26]. The norm that a Covered Entity has a privilege to use private health information (PHI) can be defined as follows:

N ≡ Use_private_health_information_privilege ∧
∃hasExplicitPrivilegeBearer ∧
∃hasExplicitPrivilegeCounterpart ∧ ∃hasPrivilegeObject

This condition is met in the following situation:

{ *Individual_perform_use_PHI* :
**Use_private_health_information_privilege**,
*Freds' Hospital* : **CoveredEntity**, *Fred* : **Person**,
*Individual_PHI_for_Use* : **Private_health_information_for_using**,
*Individual_perform_use_PHI* **hasExplicitPrivilegeBearer**
*FredsHospital*, *Individual_perform_use_PHI*
**hasExplicitPrivilegeCounterpart** *Fred,*
*Individual_perform_use_PHI*
**hasPrivilegeObject** *Individual_PHI_for_Use* }

A normative phrase is identified in HIPAA ontology with a unique legal source identifier based on the related article of the HIPAA Privacy Rule. The legal source is coupled by a hasAction / hasActivity object property to the corresponding Action.

### 3.4 N-Ary Relations Pattern

In Semantic Web languages like RDF and OWL, a property is a binary relation: it is used to link two individuals together or to link an individual to a value. In some situations, however, it is more obvious to use relationships for certain concepts involving an individual to more than one individual or value is linked these are n-ary relations [18]. In the implementation of this study, relationships in which an individual is associated with multiple other individuals occur at different places. An individual from the class Action_Individual is the relevant concept Hohfeld linked to Bearer, a Counterpart and an Object.

As a generic solution, capturing an n-ary relation involves the creation of a new class represented by new properties [18]. Translated to the HIPAA ontology for any HIPAA Action class defines a relationship with a Bearer Counterpart and an object.

# 4 IMPLEMENTATION OF HOHFELDIAN LEGAL CONCEPTS

Validation of the implementation will take place at the level of individual stakeholders that interact with each other by performing HIPAA-actions, in which one stakeholder has the role of Actor and the other has the Counterpart role (and vice versa). These interactions may result in conflicting situations and non-compliancy. Each establishment of a legal concept Hohfeld pair gives, when relevant, an indication of the level of its implementation.

## 4.1 Privilege NoRight legal concepts

The Privilege NoRight legal concept is elaborated in the HohfeldSW pattern. SWRL and SPARQL are used for the validation. With OWL, it is possible to infer implicit knowledge from explicit knowledge which is present in the model. This is consistent with the derivation of an implicit legal concept from the correlated explicit legal concept. Table 1 shows an overview of relevant OWL- DL axioms.

The rdfs:subPropertyOf axiom is used to implement a logical implication: if there is a ExplicitNoRightCounterpart then a NoRightCounterpart is implied. An object property can be linked to a certain domain: in this case, hasNoRightCounterpart is linked to the NoRight class. In this way, a Bearer can be coupled to the relevant legal concept class. The classes and object properties for the other legal concept pairs are implemented in a similar way.

Actor Fred's Hospital has the freedom (Privilege) to use Fred's private health information (PHI). Actor Fred has no right to do something about it (No-Right). When Fred tries nonetheless to prohibit the use of PHI, then an infringement occurs. Validation is effected by means of two scenarios. Scenario 1 assumes both a 'Privilege' as a 'No-Right'. Scenario 2 is only the 'Privilege' action 'use PHI'.

**Table 1.** OWF axioms for NoRight

| RDFS/OWL | Example |
|---|---|
| owl:subClassOf | ExplicitNoRight ⊆ NoRight<br>ImplictNoRight ⊆ NoRight |
| owl:EquivalentClass | ExplicitNoRight ≡ ImplicitPrivilege<br>ImplicitNoRight ≡ ExplicitPrivilege |
| rdfs:subPropertyOf | hasExplicitNoRightCounterpart ⊆ hasNoRightCounterpart<br>hasImplicitNoRightCounterpart ⊆ hasNoRightCounterpart |
| owl:equivalentProperty | hasExplicitNoRightCounterpart ≡ hasImplicitPrivilegeCounterpart<br>hasImplicitNoRightCounterpart ≡ hasExplicitPrivilegeCounterpart |

### 4.1.1 *Scenario 1: SWRL*

Step 1: Fred prohibits the use of PHI by FredsHospital (NoRight)
Step 2: Fred's Hospital uses PHI Fred (Privilege)

This is documented in the following triples:

```
Fred performProhibitUsePHI Individual_perform_prohibit_use_PHI .
Fred interactWith FredsHospital .
FredsHospital performUsePHI Individual_perform_use_PHI .
```

A generic SWRL rule validates that there is both a privilege and a NoRight same actors (with opposing roles):

```
NoRight(?x), hasNoRightBearer(?x, ?a), hasNoRightCounterpart(?x,
?b), hasNoRightObject(?x, ?o), Privilege(?y), hasPrivilegeBearer(?y,
?b), hasPrivilegeCounterpart(?y, ?a), hasPrivilegeObject(?y, ?o) ->
PrivilegeNoRightDisallowed(?y)
```

Comparison with the validation rules of the relation algebra implementation by Lalmohamed [15] helps to identify the related concepts in the Semantic Web implementation. A SWRL rule is needed for testing if both correlative legal concepts are present for an individual (intersection).

The implementation of Scenario 1 gives an individual within PrivilegeNoRightDisallowed. If in addition to a 'Privilege', a 'No-Right' action occurs. This constitutes a breach of privilege. Figure 1 shows this breach being displayed by the Semantic Web tool Protégé. In general, it is possible to check the presence of the two correlative Hohfeldian legal concepts by means of a rule-SWRL.

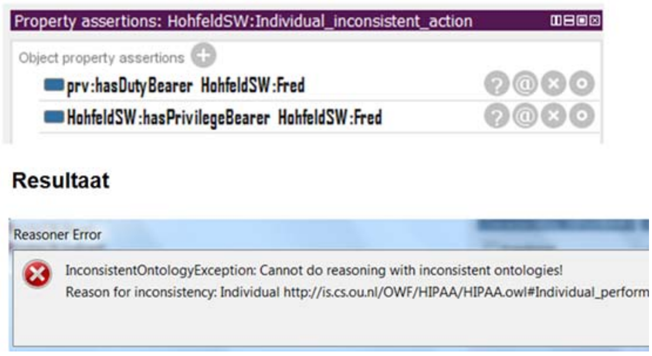


**Figure 1.** An action cannot be both a Duty and Privilege

### 4.1.2 *Scenario 2: SPARQL vs. closed world-assumption*

Step 1: Actor Fred's Hospital uses PHI Fred

This is documented in the following triples:

```
FredsHospital performUsePHI Individual_perform_use_PHI .
FredsHospital interactWith Fred .
```

RDFS, OWL and SWRL cannot establish whether a particular situation does not occur because of the open world assumption. In order to establish that there is an explicit Privilege action, but no corresponding Right-action, the following SPARQL code can be used:

```
INSERT {?ActiePrivilegeAllowed a
  HohfeldSW:PrivilegeNoRighAllowed}
WHERE
{?ActiePrivilegeAllowed a HohfeldSW:ExplicitPrivilege .
 ?ActiePrivilegeAllowed HohfeldSW:hasExplicitPrivilegeBearer
   ?CoveredEntity .
 ?ActiePrivilegeAllowed HohfeldSW:hasExplicitPrivilegeCounterpart
   ?Person .
 ?ActiePrivilegeAllowed HohfeldSW:hasPrivilegeObject ?Object
NOT EXISTS {?NoRightActie a HohfeldSW:ImplicitNoRight .
 ?NoRightActie HohfeldSW:hasImplicitNoRightBearer ?Person .
 ?NoRightActie HohfeldSW:hasImplicitNoRightCounterpart
   ?CoveredEntity .
 ?NoRightActie HohfeldSW:hasNoRightObject ?Object }}
```

This scenario provides an individual Individual_perform_use_PHI in the class PrivilegeNoRightAllowed. The SPARQL code is divided into two conjunctive elements: the conditions in respectively the WHERE and the NOT EXISTS clause. The conditions in the WHERE clause determine whether or not there is a valid Privilege action. The NOT EXISTS clause assesses that there is no NoRight action with a Bearer, Counterpart and object related to the Privilege action. The conjunctive part of the WHERE clause is in line with research into HARNESS [8] and formalized in the following way:

GC_1_Where_Cl ≡ Action(?a) ∧ ExplicitPrivilege(?e) ∧
CoveredEntity (?c) ∧ Person (?p) ∧ ObjectOfAction(?o) ∧ a(?a,?e) ∧
hasExplicitPrivilegeBearer(?a, ?c) ∧
hasExplicitPrivilegeBearer(?a,?p) ∧ hasPrivilegeObject (?a,?o)

For the sake of completeness, implicit concepts such as Person and Covered Entity are named explicitly. Note that the conditions are, to a large extent, similar to the body (condition) of SWRL rules. The ability to apply the variables in SPARQL makes it easier and more transparent to specify the conditions in a query.

This SPARQL solution is applicable in a similar way for the other pairs of correlative Hohfeldian legal concept. In our implementation, we use SPARQL on one hand to establish that a particular action does not occur (negation), and one the other hand to draw a conclusion about the classification of the action (inferencing).

The only other way to make a distinction between a potential Prohibit Use PHI action and the fact that a Prohibit Use PHI really is not applicable is to indicate explicitly that this action really does not take place, for example, in the following way:

FredsHospital performNoProhibitUsePHI
Individual_perform_no_prohibit_use_PHI .

In HohfeldSW ontology, a separate class NoRuleAvailable with relevant subclasses (like NoRightNotAvailable) can be created, which can then be used in a SWRL rule for validation in the form of:

NoRightNotAvailable(?x), hasRelatedAction(?x, ?y), Privilege(?y),
hasPrivilegeBearer(?y, ?b), hasPrivilegeCounterpart(?y, ?a),
hasPrivilegeObject(?y, ?o) -> PrivilegeNoRightAllowed(?y)

With the object property hasRelatedAction, the action which does not occur, Individual_perform_no_prohibit_use_PHI, can be linked to the action Individual_perform_Use_PHI. When Individual_perform_no_prohibit_use_PHI is made member of NoRightNotAvailable class, then application of the SWRL rules shows indeed that Use PHI is permitted.

## 4.2 Right-duty legal concepts

The Right-duty legal concept is part of the Provision Model. Compliancy is determined through the Qualification concept. SWRL and SPARQL are used for the validation. The actor Fred has the right for a notification if his private health information is used by counterpart Fred's Hospital. In addition, actor Fred's Hospital has the duty to send a notification. Two scenarios are used for validation. In Scenario 1, both a Right and Duty action are used. Scenario 2 assumes a Right action and a pre-condition (Use PHI)

### 4.2.1 *Scenario 1: SWRL rule, conditional statement, sequence actions*

Step 1: Actor Fred asks for a notification to Fred's Hospital (Right)
Step 2: Actor Fred's Hospital will send a notification Fred (Duty)

This is documented in the following triples:

Fred performRequestNotification
Individual_perform_request_notification .
Fred interactWith FredsHospital .
FredsHospital performSendNotification
Individual_perform_send_notification .

A SWRL rule can determine that there is both a Right- and Duty action for the same stakeholders (opposite roles):

Right(?x), hasRightBearer(?x, ?a), hasRightCounterpart(?x, ?b),
hasRightObject(?x, ?o), Duty(?y), hasDutyBearer(?y, ?b),
hasDutyCounterpart(?y, ?a), hasDutyObject(?y, ?o) ->
RightDutyAllowed(?y)

Implementation of the SWRL rule provides an individual in the class RightDutyAllowed. This is correct from the perspective of reasoning with Hofheldian-legal concepts. Yet this does not provide a satisfactory qualification. Fred's Hospital has only the duty to send a notification when Fred's private health information is actually used. The duty to send a notification is conditionally dependent on the use of private health information. In this study, conditional dependence has been implemented by means of a pre-condition. The pre-condition Use PHI is not fulfilled in this case, resulting in an individual in class PreConditionNotFulfilled.

It is interesting to determine what would be a logical 'total' qualification of both the Right-Duty Hohfeldian legal concept couple as the conditional dependence. The pre-condition we use here is only a pre-condition for the Duty action. As expected, only when the pre-condition is not fulfilled will this have an impact on the final qualification, in which the final classification is different from the classification on the basis of Hohfeldian legal concepts. Table 2 shows this for all combinations of the Right-Duty Hohfeldian legal concepts where the Duty pre-condition is not fulfilled.

**Table 2.** Qualification Right-Duty with precondition for Duty action

| Right | Duty | Right-Duty Qualification | Resulting Qualification |
|---|---|---|---|
| None | None | None | None |
| Request notification | | Disallowed | Allowed |
| None | Send notification | | Disallowed |
| Request notification | | Allowed | Disallowed |

It is notable that for both our Semantic Web implementation and Lalmohamed's relation algebra implementation, there is a challenge with respect to the modeling of the sequence of actions. Although the user interface of the relation algebra implementation can specify a sequence of actions, this is inferred entirely from pre-specified Hohfeldian legal action pairs, without taking into account the sequence of related actions. With Semantic Web technologies it is possible to use a Data Property "action time" in combination with numeric comparison in SWRL, to determine the order of the different actions.

### 4.2.2 *Scenario 2: SPARQL query pre-condition*

Step 1: Actor Fred asks for a notification to Fred's Hospital (Right)
Step 2: Actor Fred's Hospital uses PHI Fred (Privilege)

This is defined by the following triples:

```
Fred performRequestNotification
Individual_perform_request_notification .
Fred interactWith FredsHospital .
FredsHospital performUsePHI Individual_perform_use_PHI .
```

Also, for the validation of situations in which the Duty-action is missing, it is relevant to take into account the pre-condition. In case the pre-condition of the Duty action is not fulfilled (Fred's PHI is not used), there is also no need for the Duty action. While in this case, in which the PHI of Fred is used, a Duty action is mandatory. The following SPARQL query is developed for this situation:

```
INSERT {HIPAA:Individual_perform_request_notification a
  HohfeldSW:RightDutyDisallowed}
WHERE {  ?RightAction a HIPAA:Request_notification.
  ?Person HIPAA:performAction  ?RightAction .
  ?Person HohfeldSW:interactsWith ?Hospital .
  ?PrivilegeAction a
    HIPAA:Use_private_health_information_privilege .
  ?Hospital HIPAA:performAction ?PrivilegeAction .
  ?Hospital HohfeldSW:interactsWith ?Person
NOT EXISTS { ?DutyAction a HIPAA:Send_notification.
  ?Hospital HIPAA:performAction ?DutyAction.
  ?Hospital HohfeldSW:interactsWith ?Person }}
```

This results in a qualification RightDutyDisallowed for the same stakeholders (albeit in other role). This is applicable to both a Request notification and a Use PHI, but not a Send notification.

## 4.3 Power-Liability legal concepts

The Power Liability legal concept is part of the Provision Model. Compliancy is determined through the Qualification concept. SWRL and SPARQL are used for the validation. In the example here, the actor Fred's Hospital has the power to stop the restriction of private health information. Fred is liable to agree to end the restriction. Agreement with the restriction is in contradiction with the Power of Fred's Hospital. Two scenarios can be distinguished. In scenario 1, there is both a Power and Liability. In scenario 2, there is only Power action.

### 4.3.1 *Scenario 1: SWRL*

Step 1: Fred agrees to the restriction of PHI by Fred's Hospital
Step 2: Fred's Hospital eliminates the restriction of Fred's PHI.

This is defined by the following triples:

```
Fred performAgreeToRestrict_Liability
Individual_perform_agree_to_restrict_PHI .
FredsHospital interactWith Fred .
FredsHospital performTerminateRestriction
Individual_perform_terminate_restriction .
```

The implementation is assumed that if there is a Liability action, then it undermines the Power action. The following SWRL rule validates this:

```
Power(?x), hasPowerBearer(?x, ?a), hasPowerCounterpart(?x, ?b),
hasPowerObject(?x, ?o), Liability(?y), hasLiabilityBearer(?y, ?b),
hasLiabilityCounterpart(?y, ?a), hasLiabilityObject(?y, ?o) ->
PowerLiabilityDisallowed(?y)
```

This results in an individual in the class PowerLiabilityDisallowed.

### 4.3.2 *Scenario 2: SPARQL*

Step 1: Actor Fred's Hospital eliminates the PHI Fred restriction.

This is defined by the following triples:

```
FredsHospital performTerminateRestriction .
Individual_perform_terminate_restriction .
FredsHospital interactWith Fred .
```

The validation of the missing liability action with the absence of negation of failure in the context of Semantic Web can only be resolved by means of SPARQL. The following generic SPARQL query validates this scenario:

```
INSERT {?PowerAction a HohfeldSW:PowerLiabilityAllowed }
WHERE {?PowerAction prv:hasPowerBearer ?PowerBearer.
    ?PowerAction prv:hasPowerCounterpart ?PowerCounterpart.
    ?PowerAction prv:hasPowerObject ?PowerLiabilityObject
NOT EXISTS {
    ?LiabilityAction prv:hasLiabilityBearer ?PowerCounterpart .
    ?LiabilityAction prv:hasLiabilityCounterpart ?PowerBearer .
    ?LiabilityAction prv:hasLiabilityObject ?PowerLiabilityObject }}
```

Validation provides an individual in the class PowerLiabilityAllowed.

## 4.4 Immunity-Disability legal concepts

The Immunity-Disability legal concept is developed in the HohfeldSW ontology. The Immunity-Disability legal concept does not exist in the HIPAA Privacy Rule [15]. For demonstration purposes therefore a fictional normative phrase is developed.

A government 'Governement1' has a disability related to Fred's Hospital to prohibit the use of private health information. Fred's Hospital is immune for actions from the government to ban the use of private health information. The foregoing is validated with two scenarios. In scenario 1, there is both an Immunity action as a Disability action. In scenario 2, there is only an Immunity action. Validation of both scenarios occurs in a similar manner as in the Power Liability legal concept.

## 4.5 Opposing legal concepts

We apply OWL for validation of opposing legal concepts. A legal concept is opposed if the existence of one action rules out the existence of the other action. If action "*use private health information*" is a privilege then it cannot simultaneously be a duty because a privilege is part of the PrivilegeNoRight relationship, resulting in a different legal relationship between Actor and Counterpart. The classes in the HohfeldSW ontology are explicitly disjoint. This triggers an inconsistency message stating that the rules are contradictory. Table 3 shows the implemented disjoints.

**Table 3.** Opposing legal concepts

| Legal concept | Disjoint With |
|---|---|
| Right | NoRight |
| Duty | Privilege |
| Power | Disability |
| Liability | Immunity |

# 5 CONCLUSION

The results of this empirical study show that it is indeed possible to express legal requirements based on Hohfeldian legal concepts with Semantic Web technologies. The implementation clarifies the relationship between actors, what actions they perform and what the legal consequences are, and whether they may or may not perform these actions.

To answer the main question, with the focus on 'how and to what extend' we used a hybrid approach. On one hand for certain parts a formal logic approach was used by applying a set of conditions as a conjunctive norm. On the other hand, design principles for ontologies where used, for instance the good practice of reusability. Furthermore design patterns and normative phrase analysis played an important role in the implementation.

This study used an existing ontology as a foundation: the Provision Model. The Provision Model was no ready-made solution, but a good starting point for the implementation of Hohfeldian legal concepts. The Provision Model misses some Hohfeldian legal concepts. In this study a new ontology is developed: HohfeldSW which extends the Provision Model. In addition to legal concepts not available in the Provision Model, HohfeldSW also adds validation rules and classes to qualify legal acts. This implementation also uses ontology design patterns: n-ary relations and AgentRole.

The development of Hohfeldian legal concepts alone is insufficient to model legislation in a realistic way. In practice, laws and regulations have all kinds of dependencies between rules. In order to be able to proceed, it is necessary to model conditional statements. This is done in the form of pre- and post-conditions and exceptions.

The comparison of the Semantic Web implementation with the relation algebra implementation provides a basis for the level of implementation. The Provision Model itself was able to be implemented at the level of RDFS and OWL. Semantic Web technologies validate correlative legal concept pairs in two ways. Validation of the correlative legal concepts takes place with SWRL if something prevents both legal concepts in the relevant correlative pair, and in other cases with SPARQL because of the open world assumption. However, it is possible to provide a generic solution in all cases. The validation of opposing legal concepts is implemented with a disjoint. In addition, the treatment of pre- and post-conditions and exceptions are implemented with SWRL as SPARQL as well.

In this study, it became clear that the overall qualification about whether a particular action is or is not allowed cannot be determined on the basis of the relevant Hohfeldian legal concepts alone. Conditional statements must be factored in. Finally, it should be noted that although it has been possible to work out generic solutions for drawing conclusions normative and for cross-references, this did not happen entirely at the level of RDFS and OWL.

The source code made for this research is available online[2].

[2] http://is.cs.ou.nl/OWF/index.php5/Hohfeld_with_Semantic_Web

# 6 ACKNOWLEDGEMENTS

This work was executed as part of the Master's Thesis of Pieter Slootweg. Funding for this work comes from the Faculty of Science, Management and Technology at the Open University in the Netherlands [21].

# REFERENCES

[1] Allen, L., & Saxon, C. (1995). Better language, better thought, better communication: the A-Hohfeld language for legal analysis. 219-228. doi:10.1145/222092.222245.

[2] Athan, T., Boley, H., Governatori, G., Palmirani, M., Paschke, A., & Wyner, A. (2013). OASIS LegalRuleML. Paper presented at the Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law, Rome, Italy.

[3] Benjamins, V. R., Casanovas, P., Breuker, J., & Gangemi, A. (2005). Law and the semantic web, an introduction (Vol. 3369, pp. 1-17). Springer-Verlag, Berlin.

[4] C. Biagioli, "Law making environment: model based system for the formulation, research and diagnosis of legislation", *Artificial Intelligence and Law*. 1996

[5] A.W.F. Boer, "Legal Theory, Sources of Law and the Semantic Web", In: *Frontiers in Artificial Intelligence and Applications*, **195**, 2009.

[6] Breuker, J., & Hoekstra, R. (2004). Epistemology and ontology in core ontologies: FOLaw and LRI-Core, two core ontologies for law.

[7] P. Bos, *Bedrijfsregels in verschillende vormen-Een vergelijking op toepasbaarheid tussen SWRL en Relatie algebra bij wetteksten*, Masters Thesis, Open University in the Netherlands, Heerlen, Netherlands, 2013.

[8] Förhécz, A., and Strausz, G. (2009). Legal Assessment Using Conjunctive Queries. IDT, 1.

[9] Francesconi, E., Montemagni, S., Peters, W., & Tiscornia, D. (2010). Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language (Vol. 6036): Springer.

[10] E. Francesconi, 'Semantic model for legal resources: Annotation and reasoning over normative provisions', *Semantic Web Journal*, **7**, 255-265, 2016.

[11] Gangemi, A., & Presutti, V. (2009). Ontology design patterns Handbook on Ontologies (pp. 221-243): Springer.

[12] Hoekstra, R., Breuker, J., Di Bello, M., & Boer, A. (2007). The LKIF Core Ontology of Basic Legal Concepts. LOAIT, 321, 43-63.

[13] R.J. Hoekstra, *Ontology Representation: design patterns and ontologies that make sense*. PhD thesis, University of Amsterdam, Amsterdam, 2009.

[14] W.N. Hohfeld, "Fundamental Legal Conceptions as Applied in Judicial Reasoning", *Yale Law Journal*, 23(26(8)), p. 710-770. 1917.

[15] A. Lalmohamed, *Expressing Hohfeldian legal concepts, traceability and ambiguity with a relation algebra-based information system*, Master's thesis, Open University in the Netherlands, Heerlen, Netherlands, 2014.

[16] Maxwell, J. C., & Anton, A. I. (2010). A refined production rule model for aiding in regulatory compliance.

[17] Maxwell, J. C. (2011). A legal cross-references taxonomy for identifying conflicting software requirements.

[18] Noy, N., & Rector, A. *Defining n-ary relations on the Semantic Web*. http://www.w3.org/TR/swbp-naryRelations/. 2006

[19] V. Presutti. *AgentRole ontology pattern*. http://ontologydesignpatterns.org/wiki/index.php?title=Submissions:AgentRole. (2008)

[20] Siena, A., Mylopoulos, J., Perini, A., & Susi, A. (2009). Designing law-compliant software requirements. In e. a. A.F. Laender (Ed.), Conceptual Modeling-ER 2009 (pp. 472-486): Springer.

[21] P. Slootweg, *De implementatie van Hohfeldian legal concepts, ambiguïteit en traceerbaarheid met Semantic Web-technologieën*,

Masters thesis, Open University in the Netherlands, Heerlen, Netherlands, 2016.

[22] U.S. Department of Health & Human Services, *The HIPAA Privacy Rule*, http://www.hhs.gov/hipaa/for-professionals/privacy/index.html.

[23] Valente, A., & Breuker, J. (1994). Ontologies: The missing link between legal theory and AI & law. In H. Prakken, A. J. Muntjewerff, & A. Soeteman (Eds.), Legal Knowledge Based Systems Jurix 1994. Lelystad: Vermande.

[24] Valente, A. (2005). Types and Roles of Legal Ontologies. In V. R. Benjamins, P. Casanovas, J. Breuker, & A. Gangemi (Eds.), Law and the Semantic Web: Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications (pp. 65-76). Berlin, Heidelberg: Springer Berlin Heidelberg.

[25] T. Van Engers, A Boer, J. Breuker, A. Valente, and R. Winkels, "Ontologies in the legal domain", *Digital Government* (pp. 233-261), Springer. 2008.

[26] Ven, S. v. d., Breuker, J., Hoekstra, R., & Wortel, L. (2008). Automated Legal Assessment in OWL 2. Paper presented at the Proceedings of the 2008 conference on Legal Knowledge and Information Systems: JURIX 2008: The Twenty-First Annual Conference.

[27] R.J. Wieringa and J.-J.C. Meyer, "Applications of deontic logic in computer science: a concise overview", *Deontic logic in computer science,* M. John-Jules Ch & J. W. Roel (Eds), pp. 17-40, John Wiley and Sons Ltd, 1993.

[28] Wyner, A. (2008). An ontology in OWL for legal case-based reasoning. Artificial Intelligence and Law, 16(4), 361-387.