University of Nebraska at Omaha

DigitalCommons@UNO

Modular norm models: practical representation and analysis of contractual rights and obligations

Sayonnha Mandal
St. Ambrose University, smandal@unomaha.edu

Robin Gandhi
University of Nebraska at Omaha, rgandhi@unomaha.edu

Harvey Siy
University of Nebraska at Omaha, hsiy@unomaha.edu

Recommended Citation

ORIGINAL ARTICLE

Modular norm models: practical representation and analysis of contractual rights and obligations

Sayonnha Mandal[1] · **Robin Gandhi**[2] · **Harvey Siy**[2]

Abstract

Compliance analysis requires legal counsel but is generally unavailable in many software projects. Analysis of legal text using logic-based models can help developers understand requirements for the development and use of software-intensive systems throughout its lifecycle. We outline a practical modeling process for norms in legally binding agreements that include contractual rights and obligations. A computational norm model analyzes available rights and required duties based on the satisfiability of situations, a state of affairs, in a given scenario. Our method enables modular norm model extraction, representation, and reasoning. For norm extraction, using the theory of frame semantics, we construct two foundational norm templates for linguistic guidance. These templates correspond to Hohfeld's concepts of claim-right and its jural correlative, duty. Each template instantiation results in a norm model, encapsulated in a modular unit which we call a super-situation that corresponds to an atomic fragment of law. For hierarchical modularity, super-situations contain a primary norm that participates in relationships with other norm models. Norm compliance values are logically derived from its related situations and propagated to the norm's containing super-situation, which in turn participates in other super-situations. This modularity allows on-demand incremental modeling and reasoning using simpler model primitives than previous approaches. While we demonstrate the usefulness of our norm models through empirical studies with contractual statements in open source software and privacy domains, its grounding in theories of law and linguistics allows wide applicability.

Keywords Frame semantics · Hohfeld rights · Norms · Laws · Compliance

1 Introduction

Demonstrating information system compliance with regulations and contractual agreements requires a thorough understanding of the legal text. But regulations are necessarily complex owing to the need to accommodate a diversity of affected parties and situations. The complexity often leads to inconsistent interpretation by stakeholders and system designers. Thus, the linguistic analysis of legal text can benefit from structured methods to extract, represent and analyze normative clauses and references. We outline an approach for consistently extracting norms and related elements using semantic frames applied to regulatory text and expressing them as *modular norm models*. These models can in turn be understood visually as well as analyzed with deontic logic to check norm applicability, satisfiability, consistency and compliance.

Legal statements are typically complex due to the presence of numerous preconditions, exceptions and cross-references that may span across several sections or documents. Such conditionals generate many alternative solutions for solving the compliance problem [28]. Through this work, the theory of frame semantics for computational linguistics [19] is integrated with Hohfeld's concepts of jural claim-rights and duties [25] in a visual norm modeling framework [28, 41]. This integration occurs using semantic frames for extraction of claim-rights and duties in legal text, which are

✉ Sayonnha Mandal
   mandalsayonnha@sau.edu

   Robin Gandhi
   rgandhi@unomaha.edu

   Harvey Siy
   hsiy@unomaha.edu

1   Computer and Information Sciences Department, St. Ambrose University, 518 Locust St., Davenport, IA 52803, USA

2   College of Information Science and Technology, The University of Nebraska at Omaha, 6001 Dodge St., Omaha, NE 68182, USA

in turn mapped to norm models. This work produces templates for repeatable linguistic processing of legal text to extract norms and their relationships.

Our work extends the natural modularity of linguistic semantic frames into the representation and analysis mechanisms of a norm modeling framework. Modularization allows norm analysis and reasoning to be localized and yet stacked into larger collections when needed. One-to-one correspondence between legal statements and modular norm models promotes traceability and transparency for stakeholders. To achieve these qualities in a norm model, we extend the Nòmos 2 framework for norm modeling. We identify the basic unit of modularity in a norm model as a super-situation, which corresponds to an atomic fragment of law. A super-situation contains a single primary norm, whose activation and satisfaction may depend on a logical combination (and, or, not) of situations that describe a state of affairs. Norm compliance results are then propagated to its containing super-situation, which in turn participates in the activation or satisfaction of norms in other super-situations. This modularity decouples norm–norm relationships to allow for on-demand incremental modeling and reasoning using simpler model primitives than previous approaches. It exploits the often-hierarchical organization of regulatory documents to encapsulate norms as part of other norms, producing stackable modular norms models. An incremental modeling approach addresses scalability issues by tackling only those parts of the regulations that are relevant with respect to a given compliance question or query.

The rest of the paper is organized as follows. Section 2 outlines the development of norm templates with background necessary to understand the presented approach. Section 3 provides the steps taken to apply the norm templates to analyze open source license text. Section 4 provides a framework for reasoning with modular norm models followed by its implementation in Sect. 5. Section 6 shows the application of norm templates for privacy regulations. Section 7 describes several studies carried out to validate the readability of modular norm models. We discuss related work in Sect. 8. We conclude our paper in Sect. 9 with a discussion of our contributions, limitations, and ongoing and future research activities.

2 Modular norm models

2.1 Norm modeling framework

Analysis of norms in legal text can benefit from a representation and analysis framework. In our work, we extend the Nòmos 2 framework [28, 41] which was developed for modeling law compliant solutions in software system design. Nòmos 2 provides a norm meta-model, which enables the exploration, and selection of alternatives in a variability space defined by laws. It has a graphical notation and provides extended tool support for compliance analysis. It is primarily targeted for use by requirements engineers, but also it provides ways to collaborate with lawyers for law interpretations and validation.

The elements outlined in red in Fig. 1 are our extensions to the Nòmos 2 meta-model to enable modular norm model construction and reasoning. After defining the meta-model elements, we discuss the rationale for this extension.
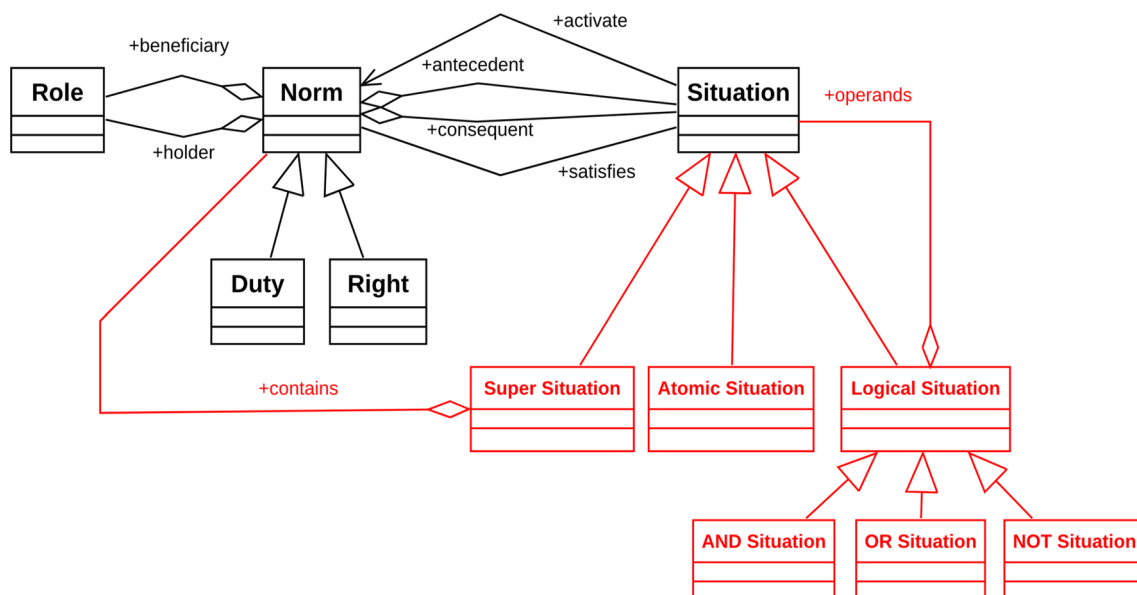

Fig. 1 Nòmos 2 meta-model extension (color figure online)

Central to the meta-model in Fig. 1 is the concept of a **Norm** [28]. It is an abstract class best described using related classes in the meta-model.

- *Duty and right* These are concrete realizations of a Norm. We only focus on claim-rights and corresponding duties.
- *Role* These denote the roles of entities related to a norm. The holder of the norm is the role having to satisfy the norm, if that norm applies. The beneficiary is the role whose interests are helped if the norm is satisfied [46].
- *Situation* A situation denotes a partial state of the world, as expressed through a proposition. Situations are antecedents or consequents of norms. *Antecedent* situations are a state of affairs that if satisfied make the norm applicable. They are related to norms using the *Activate* relationship. *Consequent* situations are a state of affairs that if satisfied then the norm is satisfied, i.e., compliant (duty) or exercised (claim-right). They are related to norms using the *Satisfy* relationship.

The term *satisfiability* is not to be confused with Boolean satisfiability as used by SAT solvers and theorem provers. Per Nòmos 2 framework it broadly refers to the satisfaction of legal clauses, e.g., Duties being carried out, Rights being exercised, or Situations determined to be true.

Beyond these core elements and relationships, we extend the original Nòmos 2 [41] meta-model as follows:

- *Super-situation* We introduce the notion of a super-situation. A super-situation *contains* a primary norm, whose activation and satisfaction may depend on a logical combination (and, or, not) of other situations. Norm compliance values are re-interpreted as satisfiability values of its containing super-situation.
- *Logical situation* Logical situations combine other situation types as operands for AND, OR and NOT situations. These situations form subclasses of the Logical Situation class in the meta-model.
- *Atomic situation* Satisfiability of atomic situations can be determined solely based on collected evidence and facts in the problem context. Unlike super-situations or logical situations, atomic situations do not require any further norm analysis.

The original Nòmos 2 meta-model relied on norm–norm relationships to express the interdependencies between atomic fragments of law. This led to a complex network of interconnected compliance values with no support for aggregation or hierarchical organization. Our extension embeds the norm within a super-situation, which corresponds to an atomic fragment of law. In turn, super-situations may participate as antecedent or consequent of norms in other super-situations. This setup allows on-demand incremental modeling and reasoning with interrelated norms based on simpler truth tables for situation–situation relationships.

The atomic situations represent assumptions that abstract or limit the scope of compliance analysis. For example, a normative reference to an entire document can be modeled as an atomic situation that represents the final compliance result of the referenced document.

In our extension, we did not include norm–situation relationships with a negative sense (e.g., Break and Block) from the original Nòmos 2 framework. This cut was made to allow reasoning consistency with both open- and closed-world assumptions. At the same time, to preserve this expressiveness of Nòmos 2, we introduce the notion of a NOT situation. With this representation, for example, a situation blocking a norm can be negated with a NOT operator, resulting in a new NOT situation. This new logical situation can then be combined with other activating situations related to the norm, to achieve an effect similar to Break and Block relationships in the Nòmos 2 framework. Finally, AND and OR situations explicitly capture the intermediate result of conjunction and disjunction relationship between two situations. This modeling feature allows a modeler to call attention to logical complexities often embedded in deceptively simple looking legal text.

2.2 Contractual rights

Hohfeld's analytical framework for fundamental legal rights [25] identifies four distinct types of rights: claim-rights, privilege, power and immunity. From this larger analytical framework, we are primarily concerned with claim-rights that impose a correlative duty on an entity in normative phrases. Such claim-rights hold not "in rem" but "in personam," i.e., they hold only for only certain people [18]. These are called contractual rights. The statement that Y has a certain duty to X, or that Y has a certain right toward X, results in X being obligated to Y to do a certain action P, which can be expressed as a relation:

Obligation (X, Y) with respect to action P [45].

Using this relation, rights and duties are always understood in the context of each other rather than being thought of as independent concepts. Such relationships abound in open source software and data licenses. Users gain rights for the use, modification or distribution of open source software/data in exchange for duties toward the copyright holders per the terms of the license. Similarly, service providers have rights to use customer information only if certain duties are carried out. In this latter situation, contractual agreements are based on laws such as HIPAA, GLB, SOX and FISMA.

2.3 Theory of frame semantics

Stakeholders who use open source software, engineer IT systems or require services from service providers are not legal experts. Thus, to avoid errors, linguistic guidance can benefit the interpretation of contractual agreements. For such guidance, we use linguistic structures based on a theory of meaning called frame semantics [19] to parameterize normative phrases in a legal document. The theory of frame semantics has resulted in the development of the FrameNet [4] database. FrameNet is an extensive collection of pre-defined linguistic structures called frames, which help understand the meaning of most words within the context of their sentences.

In FrameNet, a named frame aggregates frame elements (FEs) that describe a type of event, relation or entity and the participants involved in the frame. Words in a sentence that evoke frames are called lexical units (LUs). Normative sentences typically contain modal verbs as LUs, which evoke frames related to contractual rights and duties. Modal verbs that express obligation or logical conclusions such as "must," evoke the "*Being_obligated*" Frame. Whereas modal verbs that express ability, permission or possibility, such as "may" evoke the "*Capability*" Frame. Figure 2 shows both the "*Being_obligated*" and "*Capability*" frames.

While words other than modal verbs in a normative sentence can trigger additional frames, only the "*Being_obligated*" and "*Capability*" frames are needed to understand and model contractual rights and duties. These two frames also capture the elements of the obligation relation identified in the previous subsection.

"*Being_obligated*," as shown in Fig. 2, includes core FEs, i.e., those essential to the meaning of a frame, *Duty* and *Responsible_party*, and non-core FEs, *Condition*, *Consequence*, *Frequency*, *Place* and *Time*. This frame, like others in FrameNet, enumerates lexical units and action verbs that evoke it. For example, "responsibility," "must" and "should" are some LUs referenced by the "*Being_obligated*" frame. The presence of these lexical units in a sentence assists in evoking the appropriate frames for manual or automated annotation. Once an annotator evokes the "*Being_obligated*" frame, the definitions of its FEs guide mapping them to parts of the normative sentence to understand the meaning of the duty required. Similarly, the "*Capability*" frame, as shown in Fig. 2, has its own set of core and non-core FEs as well as LUs that evoke it. Again, the definition of each FE of the "*Capability*" frame guides manual or automated annotation of a normative sentence to understand the meaning of the claim-right provided by it.

Fig. 2 "Being_obligated" and "Capability" frame description from FrameNet (https://framenet.icsi.berkeley.edu)

Automatically mapping text to these frames continues to be a challenging problem. Research in computational linguistics presents many schemes for semantic representation and mapping of text [1]. Abstract representation collections such as AMR [5] have a large corpus of annotated text, but do not readily fit the needs of modeling claim-rights and corresponding duties. Logical forms convert sentences with similar meanings into the same structure [35] and show promise of automatic extraction [40] but do not have a mapping to frames. They also lose one-to-one correspondence with the original text. While semantic role labeling [22] tools such as SEMAFOR [17] are available for automatic sentence annotation with Frames from FrameNet, we found the output of these tools to be quite noisy for legalese and end up needing manual review. Similar issues have been reported in prior applications of natural language processing tools to identify Hohfeldian relations from text [38] as well as attempts to semiautomatically build norm models from voluminous legal text [47]. Additionally, normative sentences where modal verbs are missing or include negations require manual review. We continue to investigate computational linguistics approaches for automatic sentence annotation that can address these issues. At present, we recommend a manual, more precise annotation of normative sentences. For our application, sentence annotation is narrowed to just two frames, "*Being_obligated*" and "*Capability*," further limiting manual subjectivity and required expertise.

2.4 Norm templates

While frames in FrameNet are well defined linguistically, their correspondence to a norm model requires further development. In this subsection, we map frame FEs to norm meta-model elements. The resulting templates can be instantiated repeatedly for frame-based annotations of contractual rights and duties in legal text to produce norm models.

We perform the mapping by examining the semantic role of an FE as defined in a frame and the corresponding norm meta-model element. For example, in the "*Being_obligated*" frame, the FE "Responsible_Party" is defined in FrameNet as: *The person who must perform the Duty*. In the Norm meta-model, entities responsible for performing a duty norm are modeled as a role aggregated with the *holder* relationship. Based on this semantic equivalence, the "Responsible_Party" FE is mapped to a Role with a *holder* aggregation in the norm meta-model. Note that the Frame "*Being_obligated*" itself is mapped to the class Duty in the norm meta-model. Table 1 enumerates the complete mapping for the "*Being_obligated*" frame.

In Table 1, non-core FEs are expressed in square brackets, which FrameNet considers nonessential to the

Table 1 Being_obligated frame, frame elements and their mappings to norm meta-model elements (color table online)

Frame and Frame Element Definitions		Norm Meta-Model
Frame: *Being_obligated* **Definition:** *Under some* **Condition**, *usually left implicit, a* **Responsible_party** *is required to perform some* **Duty**.		Duty
Responsible_party	The person who must perform the Duty.	Role- Holder
[Condition]	Conditions under which the Responsible_party is obligated to perform the Duty.	Situation – antecedent (activate)
[Frequency]	Number of times an event occurs per some unit of time.	Situation – antecedent (activate)
[Place]	A location in which the Responsible_party has a specified Duty.	Situation – antecedent (activate)
[Time]	The time interval during which the Responsibly_party is obligated to fulfill the Duty.	Situation – antecedent (activate)
Duty	The action that the Responsible_party is obligated to perform.	Situation – consequent (satisfy)

meaning of the frame. However, the non-core frame elements do contribute to the schematic structure of a norm. For example, *Condition* is a non-core FE that identifies a state of affairs, i.e., a *situation*, that triggers the applicability of the duty. From a norm meta-model perspective, this *situation* would have an *activate* relationship to the *duty* Norm. Duty, a core FE for the "***Being_obligated***" frame, represents an action that the responsible party is obliged to perform. This identifies another situation. From a norm meta-model perspective, this *situation* would have a *satisfy* relationship to the *duty* Norm. Finally, the mappings in Table 1 are expressed visually in Fig. 3 as overlay of a norm-based schematic structure over the frame's linguistic structure. Frame and frame elements are in gray, while norm-based relationships are in blue. This model forms a fundamental norm template for an atomic fragment of law with duty status. While the FE definitions guide the parsing of a natural language legal statement, their mappings to norm model elements instantly transform the extracted elements into a coherent norm model.

Following a similar process, the mapping for the ***Capability*** frame is enumerated in Table 2, with the corresponding visual norm model in Fig. 4. This forms a fundamental norm template for modeling an atomic fragment of law with claim-right status.

Hohfeldian claim-rights are expressed as a relation (as explained in the previous subsection): *Obligation* (X, Y) with respect to action P [45]. Norm templates are visual manifestations of this relation. Hence, to instantiate the norm templates for a given legal text, we first need to extract Hohfeldian relations from legal text. These relations are referred to as normative sentences representing atomic fragments of law. But legalese may make it difficult to discern atomic fragments of law. Previously, Ghanavati et al. [21] have developed four rules to identify normative sentences from legal text. We reproduce the rules here for convenience.
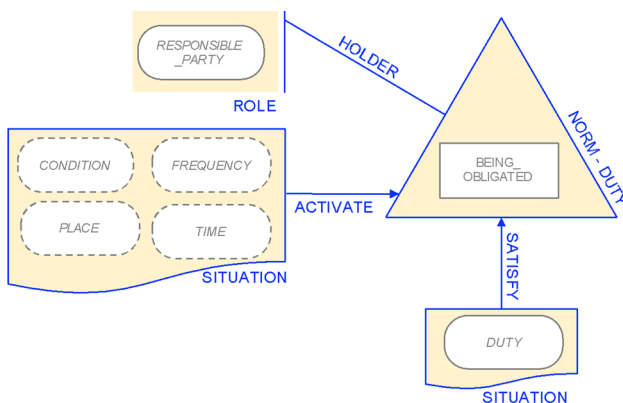
- *Rule 1* Each legal statement shall be atomic. This means that each legal statement contains one < actor > (the subject) and one < modal verb > (modality). However, the statement can also have one to many < clause > (< verb > and < actions >), 0 to many < cross-reference > , 0 to many < precondition > and 0 to many < exception > .
- *Rule 2* If a legal statement contains more than one modal verb, it must be broken down into atomic statements.
- *Rule 3* Exceptions are treated as separate statements.
- *Rule 4* If there is an internal cross-reference in a legal statement, we replace the referencing part of the statement with the referenced statement and break the statement into atomic statements. External cross-references also break into atomic statements, but they are mapped to the original legal statement via links.

These rules and the resulting sentence structure allow identifying and annotating normative sentences in legal text such that appropriate instantiations can be made using the norm templates. Table 3 shows the three-way mapping between frames, parts of speech based on the rules by Ghanavati et al. [21] and norm meta-model elements.

3 Norm template application to open source licenses

We now outline the method for applying the norm templates in the context of a strong copyleft license, Affero General Public License (AGPL) v1.0 from the Software Package Data Exchange (SPDX) License list [2]. SPDX promotes a standard annotation of license information by upstream open source software developers. This specification enables any entity in the software supply chain to effectively deal with copyrightable material for creation, alteration or use of the information in a consistent and understandable manner. Every time open source artifacts are used, copied, modified or distributed, it is prudent to analyze the legal obligations in associated licenses for compliance. We now outline the steps involved in applying norm templates to systematically analyze legal obligations based on AGPL v1.0.

Step 1: problem-driven slicing

A norm modeling framework should provide the ability to selectively elaborate a subset of norms relevant to a legal query being posed. For example, a legal query might be related to a user seeking to acquire the right to distribute a modified software package that is licensed under AGPL v1.0. We identify the action verbs "distribute" and "modify" (and their synonyms) in this legal query to scope a problem-driven slice from the entire license text. Using these action verbs, we isolate the following statements from AGPL v1.0 license. All identified statements are part of Section 2 in the license document.

Fig. 3 Duty norm template (color figure online)

Table 2 Capability frame, frame elements and their mappings to norm model elements (color table online)

Frame and Frame Element Definitions		Norm Meta-Model
Frame: *Capability* **Definition:** *An **Entity** meets the **pre-conditions** for participating in an **Event**.*		Claim-Right
Entity	The Entity whose meeting or falling short of pre-conditions is assessed.	Role- Holder
[Degree]	The extent to which the Entity exceeds or falls short of the minimum requirements for participating in the Event.	Situation – antecedent (activate)
[Relevant_ feature]	The characteristic or behavior of the Entity that is considered when judging whether the Entity might participate in the Event.	
[Role]	The particular role that the Entity plays in the Event.	
[Status]	The status of the capability.	
[Cause]	An entity, event, or state that causes the Entity to be meet or fail to meet the pre-conditions for participating in an Event.	
[Circumstances]	The Circumstances under which the Entity meets or fails to meet the pre-conditions.	
[Event]	The Event that the Entity may participate in.	Situation – consequent (satisfy)
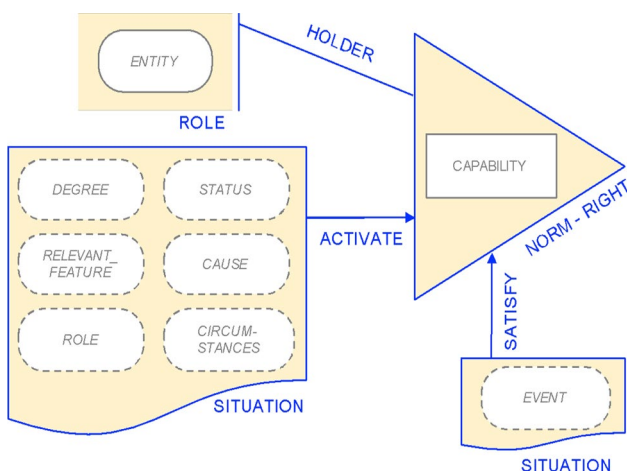
Fig. 4 Claim-right norm template (color figure online)

2. You may **modify** your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and **distribute** such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

(a) You must cause the **modified** files to carry prominent notices stating that you **changed** the files and the date of any change.

(b) You must cause any work that you **distribute** or publish, that in whole or in part contains or is **derived** from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

(c) If the **modified** program normally reads commands interactively when run, you must cause

Table 3 Mapping of elements of an atomic fragment of law to norm model elements and frames

Capability FEs	*Being_obligated FEs*	Structure of an Atomic Fragment of Law			Norm Meta-Model
Entity	Responsible_party	Subject Clause/Actor		Entity who has the right or duty to perform a action based on this norm.	Role - Holder
Frame Selection: *Capability* or *Being_obligated*		Modal verb		The modal verb (including negation), which may or may not be explicitly stated. When the modal verb is implicit, manual effort is required to determine the status of the norm as a right or duty. When explicit, a look up table is used determine the norm type.	Duty or Claim-right Norm
[Event]	Duty	Object Clause	Verb	Identifies the verb used to state the action in the Object clause.	Situation – consequent (satisfy)
			Actions	Identifies the actions or the activities required to be performed or prohibited from being performed (Duty) or can be performed by the Actor ([Event]).	
[Degree], [Relevant_ feature], [Role], [Status], [Cause], [Circumstances]	[Condition], [Frequency], [Place], [Time]	Precondition		Identifies any preconditions necessary for the Object Clause.	Situation – antecedent (activate)
Above mappings as appropriate		Exception		Exceptions are treated as separate statements.	Super Situation
		Reference/ Cross-reference		Identifies any external or internal references to different sections, different paragraphs of the same section or even different documents contained in the atomic sentence.	As needed by Situations

it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License.

(Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

(d) If the Program as you received it is intended to interact with users through a computer network and if, in the version you received, any user interacting with the Program was given the opportunity to request transmission to that user of the Program's complete source code, you must not remove that facility from your **modified** version of the Program or work based on the Program, and must offer an equivalent opportunity for all users

interacting with your Program through a computer network to request immediate transmission by HTTP of the complete source code of your modified version or other derivative work.

Step 2: Hohfeldian atomic sentence extraction and norm model transformation

We annotate the first sentence of Section 2 in the license document with elements of an atomic legal statement, as shown in Table 3, to demonstrate this step.

2. [You]$_{subject/actor}$ [may]$_{modal}$ [modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work]$_{object-clause}$ [under the terms of Section 1 above, provided that you also meet all of these conditions: (a)…(b)…(c)…(d)…]$_{preconditions}$

The modal "may" invokes the Capability frame, which results in the following equivalent mappings per Table 3.

2. [You]_{Entity} [may]_{Capability Frame} [modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work]_{Event} [under the terms of Section 1 above, provided that you also meet all of these conditions: (a)…(b)…(c)…(d)…]_{Circumstances}

The transformation of the capability frame to a norm model is then accomplished by instantiating the Capability Norm Template with mappings in Table 2.

2. [You]_{role-Holder} [may]_{claim-right} [modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work]_{situation-satisfy} [under the terms of Section 1 above, provided that you also meet all of these conditions: (a)…(b)…(c)…(d)…]_{situation-activate}

As part of activating situations, the phrase "under the terms of Section 1 above" and Sections 2(a), 2(b), 2(c) and 2(d), each point to statements with modals. The analyst/user has to decide whether to model these as either super-situations or atomic situations. The latter modeling decision terminates further expansion of the model. The former option allows detailed investigation of the norm associated with the situation. We allow this flexibility so that the analyst can decide on an appropriate stopping condition. To demonstrate both alternatives, we model the situation reflected in the phrase "under the terms of Section 1 above" as an atomic situation and Sections 2(a), 2(b), 2(c) and 2(d) as super-situations. If the referenced statement (e.g., Section 1 of the license document) is not included in the problem-driven slice, then it is best to model such references as a terminating atomic situation. This allows better scoping of the resulting model. Finally, phrases "under the terms" and "provided you meet all of the following" in this clause also suggests that these situations are in a conjunction relationship to activate the claim-right norm. This conjunction represents a AND situation, a sub-type of logical situations in the norm meta-model (Fig. 1).

Another modeling decision is related to activating and satisfying situations with compound clauses. One can decompose a compound clause into situations combined with logical situations or leave them as is. This flexibility allows for achieving the desired model abstraction and complexity. To demonstrate this, we do not further decompose the long object clause that corresponds to the satisfying situation in the first sentence of AGPL v1.0 Section 2, despite the presence of conjunctive and disjunctive conditions.

Above modeling decisions result in the norm model shown in Fig. 5. Each atomic statement of Law is contained within a unique super-situation. This super-situation and its primary norm are labeled using its legal section
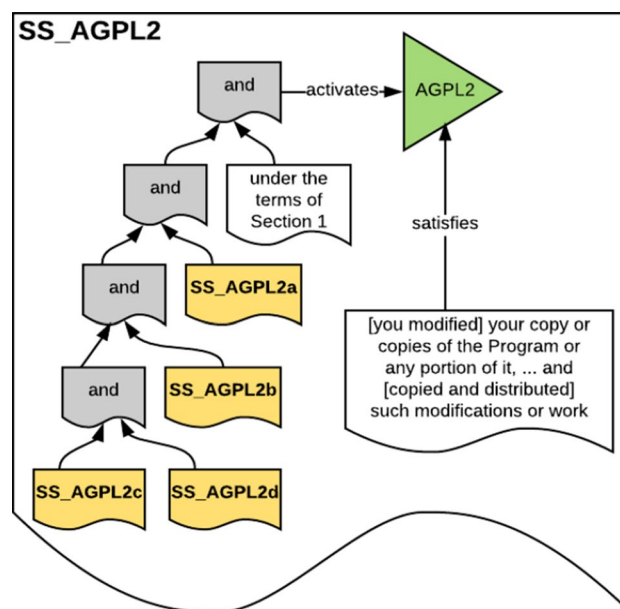

Fig. 5 Norm model for AGPL Section 2

identifier or abbreviations, for example "SS_AGPL2" with the SS prefix for super-situation and "AGPL2" for the claim-right norm in Fig. 5. Note that Sections 2(a), 2(b), 2(c) and 2(d) are modeled as super-situations (highlighted in yellow). For better readability of situations, we recommend adding contextual words. For example, the text of the satisfying situation in Fig. 5 includes "*[you modified]*" to clarify its context. We include contextual words in square brackets to distinguish them from the original license text.

Step 3: repeat process for all referenced Super-situations
A user now recursively applies Steps 1 and 2 for all super-situations that require further development. The resulting logical norm model should provide full coverage of the statements in the problem-driven slice. For "SS_AGPL2" in Fig. 5, "SS_AGPL2a," "SS_AGPL2b," "SS_AGPL2c," and "SS_AGPL2d" super-situations need further development. These super-situations correspond to Sections 2(a), 2(b), 2(c) and 2(d) from the license text, respectively. Each of these statements includes the modal "must" and thus instantiate the ***Being_obligated*** frame and corresponding norm template. Section 2(c) includes an Exception. Per rules from Ghanavati et al. [21], exceptions should be treated as separate statements. Since the exception here grants a right, we model it using the ***Capability*** frame. Figure 6 shows the super-situations resulting from all statements in the problem-driven slice. Now, it is easy to see that the AGPL2 norm becomes an exercisable 'Claim-right' of the user when the "Duties" related to its preconditions are compliant. The structure of modular norm models naturally lends itself to zoom-able interfaces that reduce information overload. But,
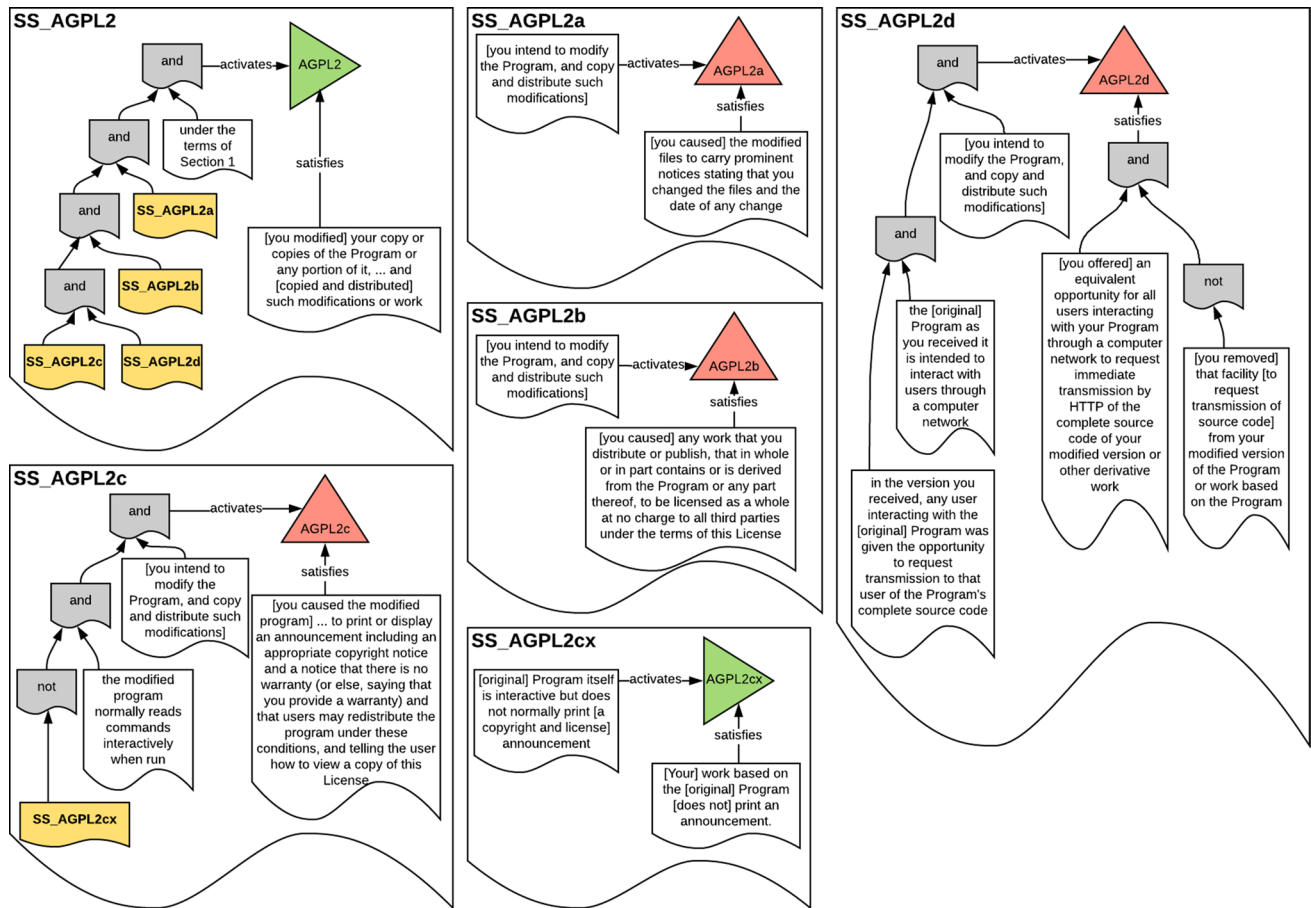
Fig. 6 Modular norm models for AGPL problem-driven slice

due to limitations in print, Fig. 6 lays them all out in the same 2D plane.

The actions in the object clause of an atomic fragment of law may include logical situations. For example, in Fig. 6 the super-situation SS_AGPL2d includes "and" and "not" logical situations with atomic situations as operands. On the other hand, logical situations with super-situations as operands compose modular norm models into an extensive network for compliance reasoning. For example, to activate the claim-right in "SS_AGPL2," all related duties in "SS_AGPL2a," "SS_AGPL2b," "SS_AGPL2c," and "SS_AGPL2d" need to be complied with. To enable automated compliance reasoning, these models are encoded in JSON, as explained in Appendix B. In later sections, we demonstrate how satisfiability and applicability values propagate in this modular norm model network.

To examine differences between norm structures of copyleft and non-copyleft licenses for distribution of modified code, we modeled Apache 2.0, a non-copyleft license. The model resulted in less stringent duties for the claim-right related to the distribution of modified code (https://github.com/robinagandhi/modularnorms).

4 Modular norm model reasoning

Norm models are appealing for manual visual analysis as well as automated reasoning about compliance. The norm models developed in the previous section are amenable to reasoning with deontic logic.

Much like the Nomos 2 framework [41], both norms and situations have satisfiability values, which can be ST (satisfied), SF (not satisfied) and SU (satisfiability undefined). Only atomic situations can be directly assigned satisfiability values based on collected evidence from the environment or by a user. By default, all atomic situations are SU unless stated explicitly. Activate and satisfy relations of a norm with situations determine the norm's applicability and satisfiability. Applicability values for a norm can be AT (Applicable), AF (Not applicable) and AU (Applicability Undefined). To facilitate rule-based reasoning using inference engines for semantic web languages, all our relationships are binary, including logical operations with situations. This is a departure from the Nòmos 2 framework, which allows multiple relationships to be combined with a single logical relationship visually as well as in textual model specification.

Table 4 Mapping from situation to norm applicability and satisfiability

Situation A --activate→	Norm	Situation A --satisfy→	Norm
ST	AT	ST	ST
SF	AF	SF	SF
SU	AU	SU	SU

Table 5 Conjunction (AND), disjunction (OR) and negation (NOT) of situations

Situation A	Situation B	Logical Situation (AND)	Situation A	Situation B	Logical Situation (OR)	Situation A	Logical Situation (NOT)
ST	ST	ST	ST	ST	ST	ST	SF
ST	SF	SF	ST	SF	ST	SF	ST
ST	SU	SU	ST	SU	ST	SU	SU
SF	SF	SF	SF	SF	SF		
SF	SU	SF	SF	SU	SU		
SU	SU	SU	SU	SU	SU		

We now list a series of truth tables to propagate satisfiability and applicability values among norm model elements. First, the truth tables for propagating applicability and satisfiability values from situations to norms based on *activate* and *satisfy* relationships are summarized in Table 4.

Similarly, truth tables for propagating applicability and satisfiability values for logical situations (AND, OR, NOT) that relates two Situations A and B as operands are summarized in Table 5.

In another departure from the Nòmos 2 framework, which uses words "comply" and "exercise" interchangeably, we distinguish between compliance with a duty or exercisability of a right. In our reasoning approach, applicability and satisfiability values of a norm and its type determine its compliance or exercisability. Duty and Right norms can be in the following states:

Duty norm

- *Compliant* (*Com*) The duty applies and is satisfied; or the duty does not apply.
- *Non-compliant* (*Vio*) The duty applies, but is not satisfied;

Right norm

- *Exercisable* (*Exr*) The right can be exercised;
- *Not Exercisable* (*Nex*) The right cannot be exercised.

Norms whose applicability cannot be determined are assigned the following state:

- *Inconclusive* (*Inc*) It is not known if the norm applies.

Table 6 lists the truth table for Duty and Right states based their applicability and satisfiability values.

Finally, the state of a norm determines the satisfiability of its containing super-situation. These values for the "contained by" relationship between a norm and its super-situation are summarized in Table 7.

With this setup, our reasoning process is much simpler compared to Nomos 2. In fact, it is quite feasible to perform it manually. Let us consider the modular super-situation SS_AGPL2 (Fig. 6). If a developer does not "*cause the modified files to carry prominent notices stating of the changed the files and date of any change,*" then norm AGPL2a is determined to be non-compliant (Vio) when the program files are modified, copied or distributed. See the propagation of truth values for SS_AGPL2a in Fig. 7 for this scenario. With super-situation SS_AGPL2a in a not satisfied (SF) status, this value is propagated for use within super-situation

Table 6 Determining norm compliance

Duty		Satisfiability		
		ST	SF	SU
Applicability	AT	Com	Vio	Vio
	AF	Com	Com	Com
	AU	Inc	Inc	Inc

Right		Satisfiability		
		ST	SF	SU
Applicability	AT	Exr	Exr	Exr
	AF	Nex	Nex	Nex
	AU	Inc	Inc	Inc

Table 7 Propagation from norms to super-situations

Duty –contained by→	Super-Situation	Right –contained by→	Super-Situation
Com	ST	Exr	ST
Vio	SF	Nex	SF
Inc	SU	Inc	SU

SS_AGPL2. This propagation, in addition to other conditions within SS_AGPL2, causes norm AGPL2's antecedent to be not-applicable (AF), which means the holder cannot exercise the right of distributing modified code. All propagations of values are per the truth tables. We have currently implemented this reasoning in OWL [7, 27] using the Semantic Web Rule Language (SWRL) [26]. As a result, a user has to simply identify the truth values of atomic situations in a problem-driven slice to obtain recommendations of compliance with duties and exercisability of rights.

Section 5 provides the OWL and SWRL implementation details.

5 Semantic web-based formalization

We make use of Semantic Web representations for the formalization of norm models [39]. We describe here the use of OWL as a way to operationalize the truth tables in Sect. 4. Axioms for computing applicability and satisfiability truth values are expressed as rules in SWRL. Rule development with SWRL first requires a mapping between norm model elements and OWL modeling elements. This mapping is as shown in Table 8.

SWRL rules are written in the form of an implication between an antecedent (body) and consequent (head), that is, if the body is true, the head is true. The OWL elements in Table 8 are used as predicates in the rules, as explained in the following tables. The rules also demonstrate how the satisfiability, applicability and compliance values propagate through inference.

Table 9 shows how compliance value COM for duty norms is computed based on its satisfiability and applicability values. Consider a duty norm, *z*. If it is applicable

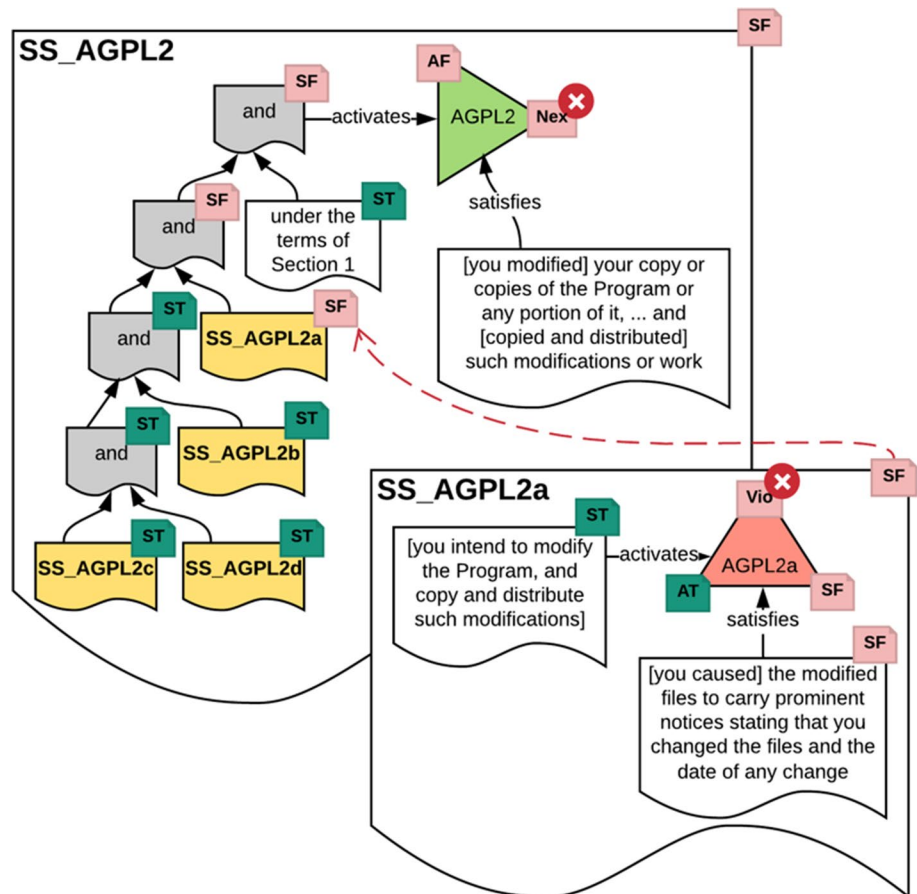**Fig. 7** Example propagation of compliance values for AGPL2

Table 8 Mapping of modular norm model elements with OWL-DL elements

Norm model elements	OWL elements
Norms (Duty, Right), Situations (Atomic, Super, Logical—AND, OR, NOT)	Class
Relations (satisfy, activate, contains, and, or, not)	Object property
Satisfiability values (ST, SF, SU), Applicability values (AT, AF, AU)	Data property (String)
Compliance values (Com, Vio, Exr, Nex, Inc)	Data property (Boolean)

Table 9 Implementing duty compliance truth table to SWRL

If Duty has values AT and ST **Then**, Duty has value Com	Let *?z* be a SWRL variable representing a norm: *Duty(?z), applicable(?z, "AT"), satisfied(?z, "ST") → Com(?z, true)*

Table 10 Computing applicability of a norm from its activating situation

If Situation *activates* Norm and Situation has value ST **Then**, Norm has value AT	Let *?z* and *?a* represent a norm and a situation, respectively: *Norm(?z), Situation(?a), activate(?a, ?z), satisfied(?a, "ST"), DifferentFrom (?a, ?z) → applicable(?z, "AT")*

Table 11 Computing satisfiability of an AND logical situation

If both Situation A and Situation B have ST values and are both related to an AND logical situation **Then**, AND logical situation has value ST	Let *?a, ?b* represent situations and *?o* represent a logical situation: *Situation(?a), Situation(?b) , AND(?o), and_(?a, ?o), and_(?b, ?o), satisfied(?a, "ST"), satisfied(?b, "ST"), DifferentFrom (?a, ?b), DifferentFrom(?a, ?o), DifferentFrom(?b, ?o) → satisfied(?o, "ST")*

Table 12 Computing the propagation of a duty norm to its super-situation

If Duty has value Com and Duty is *contained* in SuperSituation **Then**, SuperSituation has value ST	Let *?z* and *?a* represent a duty norm and a super-situation, respectively: *Duty(?z), SuperSituation(?a), contains(?a, ?z), Com(?z, true), DifferentFrom (?a, ?z) → satisfied(?a, "ST")*

(*applicable*(*?z*, "*AT*")), then the compliance value depends on whether is it satisfied (*satisfied*(*?z*, "*ST*")) or not. If it is satisfied, then *?z* is compliant (*Com*(*?z*, *true*)). A similar set of rules is used to compute other compliance values of duty norms and the exercisability of right norms. A complete listing of rules is available in Appendix A.

The applicability value of a norm is determined by the satisfiability value of its activating situation, as illustrated by the rule in Table 10. The SWRL predicate *DifferentFrom* is used to ensure that *?z* and *?a* are distinct instances; its necessity is explained later. Similar rules are used to determine additional applicability and satisfiability values of a norm.

The implementation of AND logical situations is illustrated by the conjunction rules in Table 11. In this rule, *AND*(*?o*) is a predicate that asserts *?o* as a situation representing the conjunction of *?a* and *?b*. The relation *and_*(*?a*, *?o*) specifies that *?a* is an operand of the conjunction. The same applies for *and_*(*?b*, *?o*). The underscore in the operator name distinguishes it from the built-in SWRL operator. The rules for disjunction and negation are similar and available in Appendix A.

A SWRL rule for propagating the COM compliance value of a duty norm to its super-situation as ST is illustrated in Table 12. Similar rules are used to propagate other duty-related compliance values as well as the exercisability of a right to its super-situation.

We use Protégé with Pellet reasoner plugin [32, 42] for OWL-DL-based ontological modeling and reasoning over Nòmos2 models. Pellet is a complete OWL-DL reasoner with support for reasoning with individuals and nominal support for conjunctive query [42]. Pellet provides a standard set of description inference services as follows.

- *Consistency checking* Ensuring that there are no contradictory facts present in the ontology.
- *Concept satisfiability* Checking the possibility of the presence of instances for a class.
- *Classification* Computing the subclass relations between every named class to create the complete class hierarchy.
- *Realization* Finding the most specific classes to which an individual belongs.

OWL reasoning is based on the *open-world assumption* [27], that is, information that is missing does not necessarily mean it is false. One consequence of the open world assumption is the loss of the unique name assumption [27]. With unique name assumption, two different names refer to two different individuals. However, in the open-world assumption, it is possible that the different names are in fact the same individual, information that is not yet known. In our SWRL rules, we use the *DifferentFrom* predicate to explicitly state that two individuals are not the same. Though this adds complexity to the reasoning, the open-world assumption acknowledges the fact that during compliance checking, no single assessor has complete knowledge. Moreover, the decision to resolve conflicting truth values can be deferred to application logic rather than being part of the core reasoning.

As an additional validation of the propagation rules, we re-implemented our truth tables using Datalog with Disjunction (DLV) [34]. DLV was used to implement the original norm model in Nòmos 2 [28]. Though it follows a closed-world assumption, our tests have shown that the DLV implementation of the rules produced identical results as the SWRL rules, when all atomic situations are given a satisfiability value (one of ST, SF or SU). The DLV implementation produces multiple results in cases where an atomic situation is given more than one satisfiability value.

6 Norm template application to privacy regulations

In Sect. 4, all text was contained in a single section of a relatively short license. This is often not the case with large multi-volume regulatory documents. Regulatory documents include many hierarchical sections with cross-references among its statements. Each legal statement has to be understood in the context of its containing sections, which in turn may reference or demand compliance with other sections. Modular norm models are uniquely suited to address this characteristic of regulatory documents. To highlight this aspect, we apply norm templates to text from the Health Insurance and Portability and Accountability Act (HIPAA) [24]. HIPAA stipulates the claim-rights and duties involved between individuals (patients) and service providers also known as covered entities (e.g., hospitals, insurance companies) in using the individuals' protected health information (PHI). HIPAA has been the subject of analysis in many prior research efforts on norm modeling and analysis.

Step 1: problem-driven slicing

Under the HIPAA general privacy rule, §164.502 the use or disclosure of PHI is not allowed, except as permitted. This is a "deny by default" and "allow by explicit permission" philosophy for access to protected health information. So, covered entities often have legal queries regarding when the use or disclosure of PHI is prohibited or permitted. For a running scenario, consider an individual, who is brought to a hospital for medical care. The individual subsequently dies at the hospital. The individual's son pays for the individual's medical care at the hospital. The son is now demanding that the hospital provide him with information about his father's medical condition and how it led to his death. To carve a problem-driven slice through the HIPPA regulations for this scenario, we identify the following action verbs "disclose" and "pay" along with keywords such as "relatives" and "deceased." With these search parameters, the following relevant excerpts are found in different hierarchical and cross-referenced sections of the law:

45 CFR 164.502: Uses and disclosures of protected health information: General rules

(a) Standard. A covered entity or business associate may not use or disclose protected health information, except as permitted or required by **this subpart** or by **subpart C of part 160 of this subchapter**.

(1) Covered entities: Permitted uses and disclosures. A covered entity is permitted to use or disclose protected health information as follows:

(v) Pursuant to an agreement under, or as otherwise permitted by, **§ 164.510;**

45 CFR 164.510: Uses and disclosures requiring an opportunity for the individual to agree or to object
A covered entity may use or disclose protected health information, provided that the individual is informed in advance of the use or disclosure and has the opportunity to agree to or prohibit or restrict the use or disclosure, in accordance with the applicable requirements of this section.

(b) Standard: Uses and disclosures for involvement in the individual's care and notification purposes:

(5) Uses and disclosures when the individual is deceased. If the individual is deceased, a covered entity may disclose to a family member, or other persons identified in paragraph (b)(1) of this section who were involved in the individual's care or payment for health care prior to the individual's death,

protected health information of the individual that is relevant to such person's involvement, unless doing so is inconsistent with any prior expressed preference of the individual that is known to the covered entity.

Note: For brevity, sibling provisions, such as (2–5) in 164.502(a) or 164.510(a), are not listed in the excerpt, or considered in the rest of model. If necessary, they can be fully analyzed using related super-situations.

Step 2: Hohfeldian atomic sentence extraction and norm model transformation

Annotated elements of the atomic sentence for HIPAA 164.502(a) are shown below. This sentence includes a negation "not" after the modal "may," indicating a duty. Thus, the "**Being_obligated**" norm template is evoked. It represents a duty which is satisfied by not disclosing protected health information. But, this duty is only activated if the rights provided by the preconditions are not exercisable. In this scenario, rights are preconditions which if exercised, exclude the need to satisfy a duty.

[A covered entity or business associate]$_{subject/actor}$ [may]$_{modal}$ [not]$_{modal-negation}$ [use or disclose protected health information]$_{object-clause}$, [except as permitted or required by **this subpart** or by **subpart C of part 160 of this subchapter**.]$_{preconditions}$

After applying appropriate transformations based on Table 3, the resulting norm model for HIPAA 164.502(a) is shown in Fig. 8 as super-situation "SS_HIPAA164502a." Further, the precondition "***subpart C of part 160 of this subchapter***" is modeled as an atomic situation to limit the scope of analysis in the problem-driven slice. This is a lazy evaluation feature possible due to modular norm models.

Step 3: Repeat process for all referenced super-situations

We apply steps 1–2 repeatedly to other norms in the problem-driven slice to be able to reason further about the preconditions of 164.502(a)(1). The resulting norm models are shown in Fig. 8 as super-situations "SS_HIPAA164510" and "SS_HIPAA164510b5." These super-situations illustrate the hierarchical containment of regulations and their cross-references. From this diagram, if claim-right "HIPAA164510b5" is exercisable (Exr) then the duty expressed in HIPAA 164.502(a) is not applicable; hence, its status is compliant (Com).

7 Validation of norm models

Assessing the understandability and readability of the norm models is an important step for their adoption in practice. We conducted a series of formal experiments with students as well as an informal study with a subject matter expert (SME) panel. The study with SMEs was informal in nature due to their limited availability to participate in a controlled experiment. So controlled experiments were only conducted with two groups of students from an Information Technology (IT) college, who were enrolled in a software engineering course and an IT security policy and awareness course. The SME panel for the informal study was comprised of participants contributing to the development of the Linux Foundation Software Package Data Exchange (SPDX) open standard. This group includes lawyers and technologists with many years of experience with software licenses and enterprise-wide open source software audits.

7.1 Controlled experiments

From a computer science perspective, previously outlined specification of norm models provides a validation for the theory of domain knowledge through operationalization [43]. Specifically, our approach provides semantics and composition of norm models in the context of legal text. While the theory of domain knowledge is useful to predict and reason about abstract models used in software engineering, its validation in the context of legal specifications is limited. Domain theory approaches expertise development from the perspective of recording and recall of problem abstractions as patterns. The recurring structures in norm models should lend themselves well for pattern development and recall by experts. While this is the eventual desired state, a new method of representation and reasoning—like norm modeling—should first establish basic utility for problem solving in a given domain, before attempting to measure its contribution to the development of expertise. Our goal is also to bring the ability to analyze legal text in a given situation to IT professionals, not just legal professionals. Thus, the primary objective of our empirical assessments is to find out if and how the norm models can aid IT practitioners and stakeholders to ensure compliance with relevant regulations and policies. This objective leads us to the proposition that: Norm models are *useful* to reason about compliance with legal text in a given scenario. To further investigate this proposition, we have identified three specific research questions:

RQ1 Use of norm models *improves* the accuracy of interpretation of legal text in a given compliance scenario.

RQ2 Use of norm models *reduces* the time to respond to an inquiry about compliance with legal text in a given scenario.

RQ3 Use of norm models *increases* confidence in the interpretation of legal text in a given compliance scenario.

The following subsection describes the design of the controlled experiments conducted with student participants. We conducted two experiments—one in the Fall semester

Fig. 8 Models for HIPAA problem-driven slice
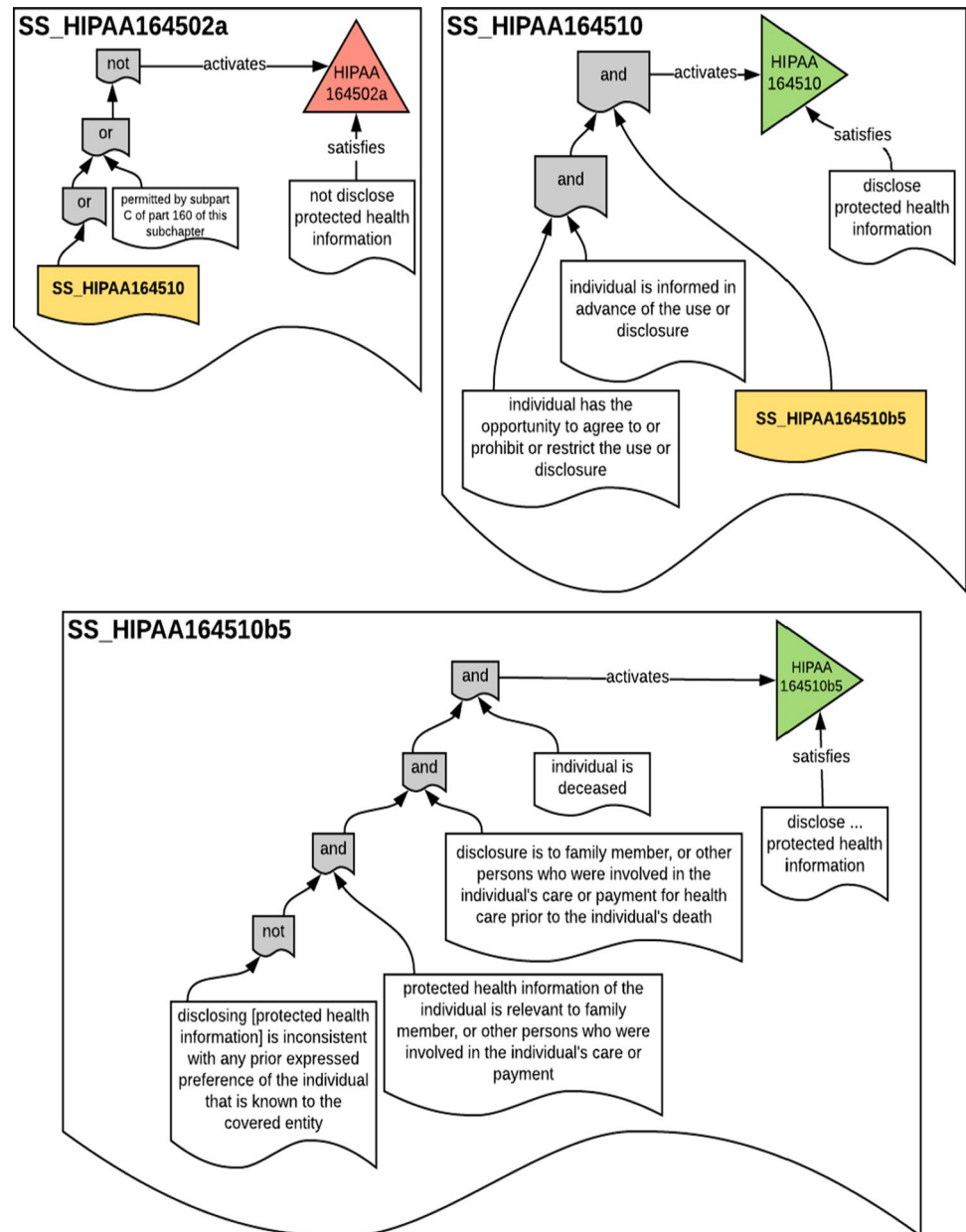
of 2016 and the other in the Spring semester of 2017. The Fall and Spring studies are similar to each other, with the exception of a few updates made in the Spring study using feedback from the Fall study. A description of the general study design of the experiments is then followed by results of the two experiments. A description of the informal study with SMEs in the SPDX group follows the results from the controlled experiments.

7.2 Experimental design

For our norm validation study, we selected a randomized, control group pretest posttest experiment design (Table 13).

Table 13 Experimental design for classroom experiments

	Control group	Treatment group
Session 1	Document only	Document only
Session 2	Document only	Document + norm model

We conducted the experiment with two groups—control and treatment—in two sessions—Session 1 and Session 2—held on two separate days of the week. The control group used the same artifacts for all tasks in both sessions. Artifacts for the control group included legal textual documents, scenario descriptions and corresponding questionnaires. The treatment group used the same artifacts as the control group for session

1, but in session 2 the treatment group used an additional artifact, which is the norm model. In the entire experiment, participants from both groups analyze five scenarios. For each scenario, the corresponding questionnaire elicits responses if the actions in the scenario comply with the provided excerpt from a privacy regulation (HIPAA) or software license (AGPL, GPL). A sample scenario used as a tutorial for the treatment group can be found here: https://robinagandhi.github.io/modularnorms/examples/yourlicense-test.html.

7.3 Experimental variables

In this section, we identify the independent variables manipulated by the experiment design and elaborate on the dependent variables collected from the participants.

7.3.1 Independent variables

The experiment manipulated these three *independent variables:*

1. *Group*—refers to the Group assigned (1 or 2, Group 2 is the treatment group that used the norm models)
2. *Session*—refers to the experiment round (1 or 2, conducted on two separate days)
3. *Legal document type*—refers to the type of legal document provided to the participants (Software License: GPL and AGPL, or Healthcare Privacy statement: HIPAA)

7.3.2 Dependent variables

Three *dependent variables* were collected from each participant based on their responses to the questionnaire. As described in Table 14, the three dependent variables are:

1. Accuracy of interpretation
2. Time of response
3. Confidence of response

Accuracy of interpretation For the accuracy of interpretation metric, we looked at the correctness of both the *answer* to the compliance question and the *sentence(s)* from the original legal text selected as relevant to the question.

Thus two scores are collected:

1. *Answer accuracy* Accuracy in answering the compliance question (**Correct**, **Wrong**, **Need More Information**(**NMI**)):

In order to measure this attribute, we counted the number of *correct answers* for the compliance question for the control group and for the treatment group working with the same scenarios. This enabled an efficient comparison of the effect and ease of use of our models in aiding users in correctly interpreting legal texts. Answer accuracy was marked in three ways—Correct (if participant chose the correct answer), Wrong (if the participant chose the incorrect answer) and NMI or Need More Information (if the participant mentioned that more information was required for him/her to answer the question).

2. *Sentence accuracy* Accuracy in selecting relevant legal sentences (**Correct**, **Wrong**):

We considered the sentence to be correct if the study participants selected the correct legal sentences relevant to the compliance question. Sentence accuracy was marked in two ways—**Correct** (if participant chose the correct set of sentences) and **Wrong** (if the any of the required sentences were missing in the set identified by the participant).

Although the interpretation of answer accuracy remains the same for the two studies, the interpretation of sentence accuracy was revised from the Fall to the Spring study. For the Fall study, sentence accuracy represents correctness in identifying relevant legal statements from the given legal text (for the control group) and identifying relevant situations from the norm models (for the treatment group). In the Spring study, on the other hand, sentence accuracy

Table 14 Dependent variables and how they are measured

Dependent variable	Measures	How quantified
Accuracy of interpretation	Accuracy in answering compliance question (Correct, Wrong, Need More Info) Accuracy in selecting relevant legal sentences (Correct, Wrong)	Composite Score that combines correctness of answer and sentence selection (see Table 15)
Time of response	Self-reported clock time (in seconds) from start of reading scenario to providing response	Control group: from the start of question till after the question is answered Treatment group: from start of question till selection of relevant situations/norm
Confidence of response	Self-reported confidence in answering (Very Confident, Semi-confident, Guessing)	Very = 2, Semi = 1, Guessing = 0

represents correctness in identifying relevant legal statements from given legal document (for the control group) and identifying the most relevant starting norm from the norm models (for the treatment group). These differences will be further elaborated during our subsequent discussion on the different scoring mechanisms in the respective study sections. Both studies, however, utilized the same scoring table (Table 15) for determining a score for answer and sentence accuracy. The relevant scores for this table have been adapted from Klymkowsky et al. [31].

Time of response (in seconds) The ability of developers to understand our model quickly and efficiently is essential to the practical adoption of this modeling approach. We asked the participants to record the start time and the stop time for their response for each of the scenario questions during the experiment. This enabled us to measure if and by what extent it is easier to interpret legal text with models as opposed to only be given the entire legal paragraph.

To measure the time of response, in Session 1, we measured the time recorded from the start of each question till the time recorded after the question is answered by the participant. This analysis was performed for each participant for each of the questions in the scenario in both the groups. In Session 2, for the control group, the total time was measured similar to the Session 1 total time. For the treatment group, however, the total time was given by the time recorded for answering the question on selection of relevant situations/norm for the given scenario. The time recorded for answering the actual compliance question was not taken into account. This technique was utilized since we aimed to measure the time taken for using norm models as opposed to the plain legal text, and identifying the relevant situations by the treatment group equated to the time taken to search through the text and answer the final question in case of the control group.

Confidence of response (very, semi, guess) We measured if using the models instills a sense of surety or confidence in the participants for their responses to the scenario questions. We asked the participants to self-report their level of confidence in their answer to each compliance question on a three-point scale: *Very Confident*, *Semi-confident*, and *Guessing*. This measure contributed to assessing whether using norm models makes system developers more confident in interpreting the given complex legal text.

Table 15 Composite scoring chart

	Sentence correct	Sentence wrong
Answer correct	$+4$	$+2$
Answer wrong	(-2)	(-4)
Need more information	$+2$	(-2)

7.3.3 Subject variables

We also collected demographic information through a pre-test questionnaire to all the participants. This contained questions regarding the background and experience of the participants. The demographic test questions were:

- Have you developed for open source software before? (*OSSExp*)
- Have you developed software professionally before? (*ProExp*)
- Do you have prior experience with reading and building models (e.g., UML) for software engineering? (*ModelExp*)
- Do you have a background in Computer Science? (*CSBg*)
- Do you have a background in Law? (*LawBg*)
- Have you taken a course in Data Structures? (*DStructExp*)
- Is English your first language? (*EnglProf*)

7.3.4 Exit survey

At the end of the study, we conducted an exit survey to gather information regarding the perception of using the "models" versus using the "text only" for answering the scenario questions. The survey provided us with added insights into the readability and understandability of our model-driven approach of legal text interpretation. The responses were collected in a binary format of agree/disagree. Some of the survey questions for the treatment group were along the following lines:

- Do you feel the models helped you?
- Do you feel you are more confident in your responses for the models vs. the text?
- Do you feel you obtained added guidance in interpreting the legal text with the help of the models?

We had corresponding questions for control group who completed the experiment with only the original legal text. The questions for this group were of a separate nature.

- Was the text easy to understand to answer the questions?
- Do you feel you needed some extra guidance in some form to understand the text better?
- Were you absolutely confident in your responses?

7.4 General analysis strategy

We administered our experiment questions and the surveys through the research tool Qualtrics. The results obtained from the study are analyzed using the R statistical tool.

Our analysis was based on comparing the performance of participants from Session 1 to Session 2 on each of the three dependent variables described in the previous section:

- Accuracy of Interpretation
- Time of Response
- Confidence of Response

For each dependent variable, we summed the measures obtained from each participant per session. We then subtracted the Session 2 total of each participant with their Session 1 performance. This difference was plotted using a boxplot to show the distribution of the improvement (or lack thereof).

Statistical analysis We used the Shapiro–Wilk normality test to find out if the distribution of the summed values was normal or not. If it was found to be normal, we performed the *t* test (used for parametric analysis) to determine if the difference is statistically significant. If the distribution was found to be not normal, we performed the Wilcoxon test (used for nonparametric analysis).

To gain further insights in explaining the accuracy results, we performed analysis of variance (ANOVA) to estimate the effects of subject variables on the observed accuracy of a participant.

7.5 Analysis of results: Fall 2016 experiment

The experiment was carried out in Fall 2016 with a cohort of Masters' Computer Science students taking a software engineering class. The students in this class have enough background to identify software related topics, but do not have too much knowledge on natural language extraction or norm modeling to introduce any bias into the study. The experiment was mentioned in the syllabus on the first day of class. Information about the experiment's objectives was announced in class shortly before the start of the experiments.

We had 32 participants. For Session 2, the treatment and control groups consisted of 16 participants each. We received 15 valid responses for the treatment group, while for the control group, we obtained 16 valid responses.

7.5.1 Accuracy of interpretation

We summed the composite accuracy score per participant per session. The Shapiro–Wilk test for normality of the participant accuracy totals was not significantly different from
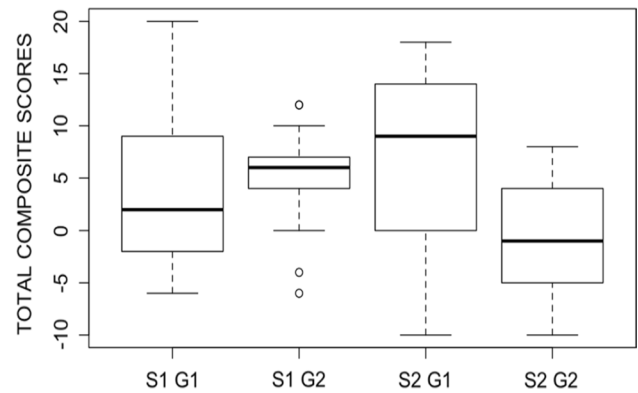

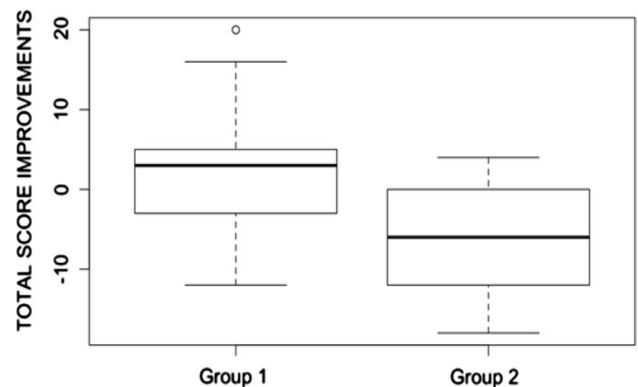**Fig. 9** Aggregate responses by individual and session (Fall 2016)

Fig. 10 Composite score difference from session 1 to session 2 (Fall 2016)

normal ($p > 0.1$). Our primary test will then be parametric (paired *t* test).

The paired *t* tests indicate that Group 1 (control group) accuracy score did not have a significant difference from Session 1–2 while Group 2 (treatment group) was significantly different ($p < 0.01$). The boxplot Fig. 9 further indicates that Group 2's performance was significantly worse in Session 2 using the norm models. The decrease in performance was confirmed by the difference between each participant's Session 2 and Session 1 accuracy scores. Figure 10 shows a drop in accuracy for Group 2.

ANOVA To help us understand why the performance for Group 2 got worse when the participants used the norm models, we examined other factors such as demographics and other subject variables that could help explain the accuracy score values by creating several analysis of variance (ANOVA) models.

Our initial ANOVA model for accuracy included independent variables and the dependent variables Time of Response and Confidence of Response. We included these dependent variables as it is possible that there is a causal

relationship between the time it takes to respond and the participant's confidence in his/her response to the accuracy of the response. The ANOVA model added all the subject variables described in Sect. 7.3.3 to identify individual factors that might have contributed to the accuracy result. ANOVA produced the results in Table 16.

Several subject variables were unbalanced; participants overwhelmingly reported high modeling experience (ModelExp), CS background (CSBg) and data structures experience (DStructExp), and low background in law (LawBg). These variables were dropped, resulting in Table 17. The interaction between Session and Group (Session:Group) was significant ($p < 0.05$) as expected. We also observe that English Proficiency (EnglProf) was significant ($p < 0.05$), indicating that those participants who have English as their first language tended to have higher accuracy. ANOVA reported that Confidence was significant ($p < 0.05$), but this goes away when EnglProf and Session:Group are first accounted for. We revisit the role of confidence in a later discussion.

7.5.2 Time of response

We summed the time of response per participant per session. The Shapiro–Wilk test for normality indicates that the participant time totals were different from normal ($p < 0.001$). Thus, we use paired Wilcoxon test to compare the time for both groups.

We subtracted the Session 2 time total of each participant with their Session 1 total. This difference is the time improvement for each student from session 1 to session 2. The resulting boxplot in Fig. 11 shows that Group 1 has a greater time reduction (improvement) compared to Group 2.

The Wilcoxon test conducted indicated that the total time difference for Group 2 is greater than the total time

Table 16 Significance of factors from ANOVA—full model (Fall 2016)

Factors	df	Sum Sq.	Mean Sq.	F value	Pr(>F)
Session	1	52.56	52.56	1.0779	0.30406
Group	1	138.06	138.06	2.8313	0.09855**
Time	1	0.26	0.26	0.0053	0.94209
Confidence	1	207.61	207.61	4.2577	0.04418*
OSSExp	1	11.88	11.88	0.2435	0.62378
ProExp	1	4.37	4.37	0.0897	0.76577
ModelExp	1	2.44	2.44	0.0500	0.82392
CSBg	1	111.24	111.24	2.2812	0.13712
LawBg	1	0.56	0.56	0.0116	0.91472
DStructExp	1	0.17	0.17	0.0035	0.95331
EnglProf	1	344.95	344.95	7.0742	0.01043*
Session:Group	1	311.45	311.45	6.3871	0.01464*
Residuals	51	2486.88	48.76		

*$p < 0.05$, **$p < 0.1$

Table 17 Significance of factors from ANOVA—simplified model (Fall 2016)

Factors	df	Sum Sq.	Mean Sq.	F value	Pr(>F)
Session	1	52.56	52.562	1.0454	0.31104
Group	1	138.06	138.063	2.7459	0.10320
Time	1	0.26	0.260	0.0052	0.94295
Confidence	1	207.61	207.615	4.1292	0.04699*
OSSExp	1	11.88	11.876	0.2362	0.62890
ProExp	1	4.37	4.374	0.0870	0.76914
EnglProf	1	203.11	203.109	4.0396	0.04936*
Session:Group	1	289.20	289.202	5.7519	0.01989*
Residuals	55	2765.38	50.280		

*$p < 0.05$, **$p < 0.1$

difference for Group 1 ($p < 0.001$). Thus participants from the treatment group (Group 2) used more time in Session 2 to complete the experiment than they did in Session 1, while the Control group remained relatively consistent from Session 1–2. This is likely explained by the initial complexity of the norm models overwhelming the participants, causing them to take more time to understand and navigate the models before they could answer the questions.

7.5.3 Confidence of response

We examined if use of norm models led to increased confidence in the participants' perceived correctness of responses. As with time, we summed the confidence score per participant per session and then subtracted each participant's Session 2 total from Session 1. This difference is the improvement in confidence scores for each participant from session 1–2. The resulting boxplot in Fig. 12 shows that Group 2 participants had lower confidence in their answers when using the norm models.

The Shapiro–Wilk normality test indicates that Group 2's distribution is significantly different from normal ($p < 0.05$);
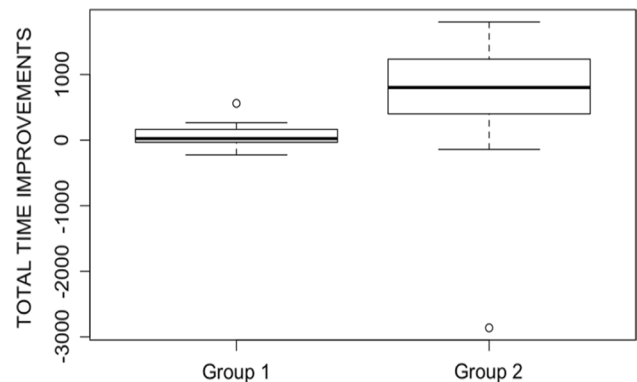

Fig. 11 Difference of total time per participant from session 1 to session 2 (Fall 2016)
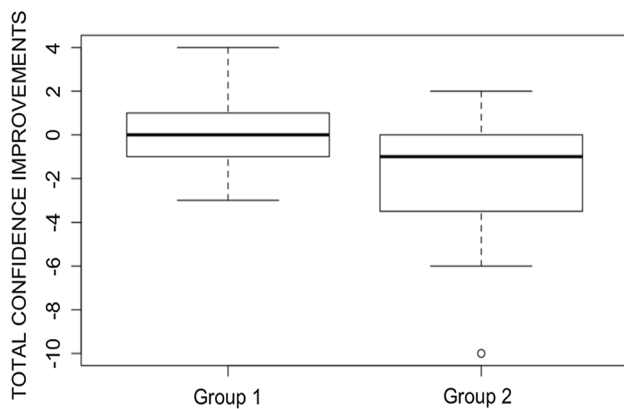
Fig. 12 Difference of total confidence score per participant from session 1 to session 2 (Fall 2016)

thus, we use the Wilcoxon test to compare the improvement of Group 1 to Group 2. The Wilcoxon test showed a p value < 0.1; thus, Group 2 showed a deterioration in confidence in Session 2 as compared to Group 1.

This is likely attributed to the fact that the participants initially faced with the daunting task of understanding and navigating the models on paper and trying to relate it to the legal text given, would not have much confidence in their first ever usage of the models. Their confidence scores thus reflect their lack of certainty about using a brand new method for the first time in the experiment.

7.5.4 Observations and adjustments

The Fall study results and comments from exit surveys helped us gain certain significant insights. These insights informed adjustments to the administration and artifact presentation for a subsequent study. Table 18 outlines the important insights gained, changes made in the Spring study, and the rationale for them.

7.6 Analysis of results: Spring 2017 experiment

The second experiment was carried out in Spring 2017 with a cohort of BS cybersecurity students taking an IT security policy and awareness course. Session 1 had 34 valid responses. Session 2 had 16 valid responses for the treatment group, and 17 valid responses for the control group.

7.6.1 Accuracy of interpretation

Unlike the Fall 2016 experiment, the Shapiro–Wilk test for normality of the participant accuracy totals was significantly different from normal ($p < 0.05$). Thus, we use paired Wilcoxon test to compare the scores for both groups. The results of the paired Wilcoxon test indicate

that Group 1 (control group) accuracy score did not have a significant difference from Sessions 1–2, while Group 2 (treatment group) had a statistically significant difference ($p < 0.05$). In session 1, the control group (Group1) performed better than the treatment group (Group 2). However, the boxplot in Fig. 13 indicates that Group 2's performance was significantly better in Session 2 using the norm models. The increase in performance was confirmed by taking the difference between each participant's Session 2 and Session 1 accuracy scores. Figure 14 shows a clear rise in accuracy for Group 2.

ANOVA Our initial ANOVA model for accuracy for the Spring participant data included independent variables and the dependent variables Time of Response and Confidence of Response. We included these dependent variables as it is possible that there is a causal relationship between the time it takes to respond and the participant's confidence in his/her response to the accuracy of the response. The ANOVA model added all the subject variables described in Sect. 7.3.3 to identify individual factors that might have contributed to the accuracy result. ANOVA produced the results in Table 19. Several subject variables were unbalanced, that is, participants overwhelmingly reported low background in law (LawBg) and CS (CSBg). These variables were dropped. Next, we dropped other subject variables that were not significant in the resulting model: professional experience (ProExp), open source experience (OSSExp), and model experience (ModelExp), resulting in Table 20. We see that Time, Confidence and English Proficiency are significant, $p < 0.05$. The interaction between Session and Group (Session:Group) was not significant in this case. Our observations of English Proficiency (EnglProf) being significant ($p < 0.05$) indicates that those participants who had English as their first language tended to have higher accuracy. ANOVA reported that Confidence and Time were significant ($p < 0.05$), indicating that participants who spent more time and reported more confidence tend to also get higher accuracy scores.

7.6.2 Time of response

The Shapiro–Wilk normality test indicates that both groups' time data are significantly different from normal ($p < 0.001$); thus, we use the Wilcoxon test to compare the improvement of Group 1 to Group 2. The results for time of response (see Fig. 15) are similar to the Fall 2016 experiment, with participants in Group 2 (treatment group) using significantly more time during Session 2 while participants in Group 1 (control group). The Wilcoxon test also confirms it ($p < 0.001$). This indicates that though there is automated support for norm reasoning, the manual task of exploring a nontrivial norm model in order to identify applicable situations remains highly time-intensive.

Table 18 Summary of observations from the Fall study and adjustments made for the Spring study

|	Fall study observations		Spring study adjustments	
Insights	Control group	Treatment group	Adjustments needed	Rationale for adjustment
Mode of interaction	Interaction with computer-based artifacts only: Computer-based legal document, scenario and questionnaire	Interaction with computer and paper-based artifacts: Paper-based norm models and truth tables, and computer-based document, scenario and questionnaire. Requires constant switching	For the Treatment group, norm models need to be laid out in a single webpage to keep interaction similar to the original model printout. Model elements made interactive to represent compliance values	Have a similar mode of interaction (Computer only) for both groups
Context for situations	Situations can be read in the context of the entire legal statement in a document. Pronouns easily resolved	Concise listing of situations to simplify visual presentation lacked context of the entire sentence as well as the previous sentences in the legal document. Pronouns hard to resolve	For the Treatment group, missing context, subjects and objects, and pronouns need to be fully qualified in norm model situations	Give both groups access to similar contextual information to understand and interpret situations
Identification of norms relevant to scenarios	A familiar text interface intuitively enables keyword-based search. While this allows a quick listing matching legal statements, the reader may miss synonyms or the larger context of the situation	A visual, modular and hierarchical artifact on paper intuitively leads to manual inspection of the norm model when searching. This leads to fixations to limited parts of the model and may potentially miss other relevant norms	For the control group include a prompt to identify all relevant paragraphs that support their answer. Similarly, the Treatment group identifies all relevant starting norms in the models that support their answer	The prompts encourage the subjects from both groups to examine the other norms. Additionally, it allows better traceability of the reasoning behind answers provided
Wording of scenarios and questions	The exit survey pointed to some minor issues with the readability and interpretation of scenarios and questions. Same scenarios and questions were given to both groups		Make needed corrections in the scenario and questions	Avoid confusions related to the expression of the scenarios and questions
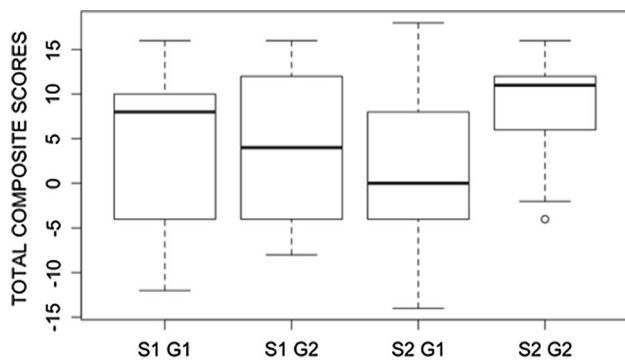
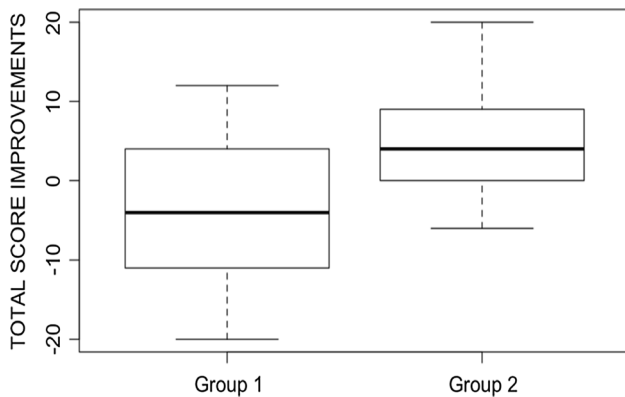**Fig. 13** Aggregate responses by individual and session (Spring 2017)

Fig. 14 Composite score difference from session 1 to session 2 (Spring 2017)

Table 19 Significance of factors from ANOVA—full model (Spring 2017)

Factor	df	Sum Sq.	Mean Sq.	F value	Pr(> F)
Session	1	7.2	7.20	0.1213	0.72893
Group	1	178.4	178.35	3.0067	0.08842**
Time	1	258.5	258.45	4.3571	0.04142*
Confidence	1	389.0	389.01	6.5581	0.01316*
OSSExp	1	3.3	3.25	0.0548	0.81575
ProExp	1	62.1	62.09	1.0468	0.31065
ModelExp	1	25.2	25.22	0.4252	0.51703
CSBg	1	55.9	55.91	0.9426	0.33578
LawBg	1	115.3	115.33	1.9442	0.16872
DStructExp	1	119.7	119.74	2.0186	0.16092
EnglProf	1	75.1	75.05	1.2653	0.26545
Session:Group	1	117.4	117.43	1.9797	0.16494
Residuals	56	3321.8	59.32		

*$p < 0.05$, **$p < 0.1$

7.6.3 Confidence of response

The Shapiro–Wilk normality test indicates that both groups' confidence data are significantly different from normal

Table 20 Significance of factors from ANOVA—simplified model (Spring 2017)

Factor	df	Sum Sq.	Mean Sq.	F value	Pr(> F)
Session	1	7.2	7.20	0.1282	0.72151
Group	1	178.4	178.35	3.1784	0.07959**
Time	1	258.5	258.45	4.6059	0.03585*
Confidence	1	389.0	389.01	6.9326	0.01071*
DStructExp	1	74.1	74.05	1.3197	0.25513
EnglProf	1	287.6	287.57	5.1247	0.02716*
Session:Group	1	111.3	111.25	1.9826	0.16419
Residuals	61	3422.9	56.11		

*$p < 0.05$, **$p < 0.1$
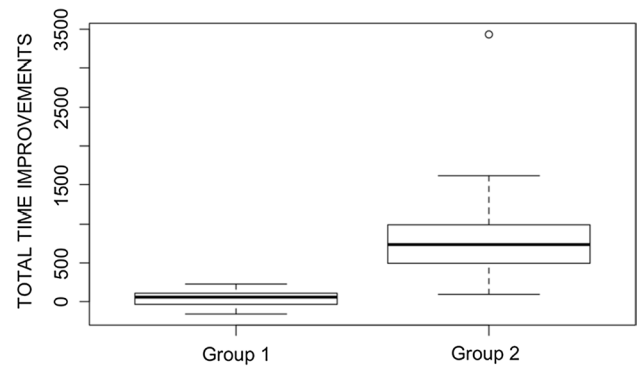
Fig. 15 Difference of total time per participant from session 1 to session 2 (Spring 2017)

($p < 0.01$); thus, we use the Wilcoxon test to compare the improvement of Group 1 to Group 2. Results for confidence of response (see Fig. 16) indicate no statistical difference in improvement from Session 1 to Session 2 for either group. There is, however, a slight improvement over the Fall 2016 experiment which showed a significant deterioration on confidence for the treatment group. Compared with the Fall study, it can be seen that the use of the interactive tool made the participants more confident than the last experiment so much as to become at par with the control group. Thus even though the treatment group was faced with the norm models for the first time, the interactivity and ease of use of the tool may have contributed to their confidence values not dropping lower than the control group.

7.6.4 Discussion

We briefly discuss the findings from the Spring 2017 experiment. This was carried out after making improvements in the instruments used, in particular, introducing an interactive norm model exploration interface and revising the text within the models so as to provide more context to each situation. The design of the experiment remained the same
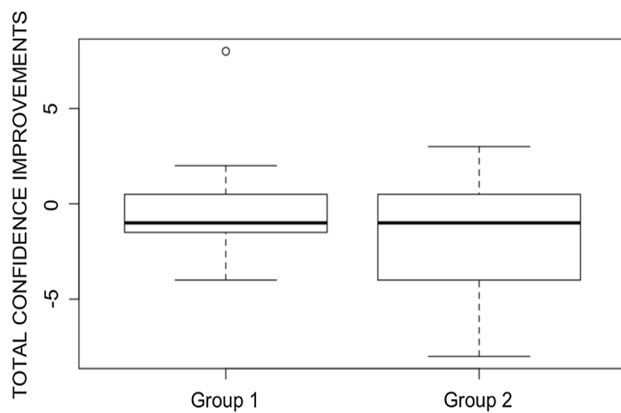
Fig. 16 Difference of total confidence score per participant from session 1 to session 2 (Spring 2017)

as the Fall 2016 experiment. We examine each research question in turn.

Research question 1: norm models and accuracy

With the improved interactive experience as well as clearer explanation of the models and their elements, we find that using norm models improves the ability of experiment participants in interpreting the compliance question for each scenario. Thus, RQ1 is "yes." The exit surveys indicate that Group 2 participants in general liked the interactive interface, though many expressed concern that the models were too expansive and complicated to understand, requiring the reader to pan and zoom across a very large space. In our ongoing work, we are attempting to address this by better utilizing the natural modularity of our norm models to present only a single module at a time.

Research question 2: norm models and time to respond

The use of norm models improves the accuracy of interpretation, but at the cost of increasing analysis time. Thus, RQ2 is "no." We expect users to become more efficient with models with continued usage and hence require less time. On the exit survey, some respondents wished there was additional practice in analyzing the norm models, before being asked to evaluate an entire model.

Research question 3: norm models and confidence

The use of norm models did not improve confidence in answering compliance questions. Thus RQ3 is also "no." Nevertheless, confidence data were not statistically different between Groups 1 and 2, which is an improvement from the Fall 2016 study where confidence actually went down for Group 2.

7.7 Informal expert study

We had planned to demonstrate our norm models to legal and open source experts. We were able to work with six participants from the SPDX community working group

meetings. Their expertise stems from practical knowledge about open source license compatibility issues, direct involvement in legal cases and application scenarios that require consideration of software licenses, and extensive experience in open source communities.

At first, the experts observed the norm modeling tutorial designed for the treatment group participants of our user study in session 2. It was in the form of a remote screen share presentation, with the experts asking questions and clarifications. At the end of the tutorial, we showed a live demonstration of the simple scenario and model used in the tutorial as well as the AGPLv1.0 model and corresponding scenarios and questions used in the study. We did not ask our experts to individually solve the questions, rather we walked them through the scenarios, the norm model and the solution steps in a think-aloud manner. We answered questions about model construction, connections to the legal text and reasoning and propagation using a given scenario. Comprehensive running notes were taken during the demonstration of the tutorial and scenario solutions. At the end, we asked the experts to complete the questions from the exit survey provided to the treatment group of our user study and recorded their responses. We now briefly discuss a summary of their responses.

Using simple yes and no responses, five out of six experts expressed that the norm models were more readable compared to legal text and that the norm models helped interpret the legal text better. Four out of six experts expressed that they were more confident in their interpretation of the license text using the models. The experts provided free-form comments regarding what they liked about norm models and what could be improved. Only one representative quote is displayed here for brevity.

The experts were generally impressed with our approach.

"Visually lays out the compliance process"

At the same time, the need for a better formed UI was reflected in many comments.

"There are probably better ways to visually display the interface"

Direct interaction with the models was perceived to provide structure to the development workflows which would be useful in the professional world for collaboration between developers, designers and legal teams.

"To developers and legal teams, the barebones interaction with the model would be great."

The models were deemed helpful in identifying complexities and variabilities in legal textual documents and to demonstrate how the technological decisions were based on legal text. They were generally satisfied with the conclusions derived from the formalized reasoning with satisfiability and

applicability values. But cautioned against labeling a norm as "compliant," as this is best left up to the legal team and judicial proceedings to determine.

> "Very interesting project—structuring legal and compliance information in formal models is a field ripe for disruption and potentially automated/scalable risk decisions. What I like best is the logical and hierarchical connection of factual situations to compliance conclusions. At the same time, opinions about compliance stray very close to the "practice of law" so disclaimers or softer language about "likely compliant" would probably be a good idea. But even with those softeners, it can be very powerful to link up facts to intermediate conclusions ("likely conclusions") to ultimate conclusions, so I found the multilayered scenario model quite interesting."

Interaction between multiple software licenses was also suggested as much needed feature for compliance analysis. Reuse of situations and their compliance values across multiple licenses was identified as a time-saving feature to have.

> "Expand to a collection of licenses that operate on shared code - would require including a property about what copyrightable element the license applies"

This feedback from subject matter experts provides face validity for the general utility of modular norm model representation of legal text in compliance and development workflows, as well as the understandability of these models.

8 Related work

Interpretation of law tends to be subjective. The open-texture problem [8] describes this recurring issue where a legally binding interpretation of applicability is left to the legal system to decide based on circumstances of individual cases. While it takes a trained eye to directly extract case relevant information from legal text [15, 37, 38], identifying grammatical parts of a sentence is more straightforward. In our work, the mapping of sentence parts to "Being_Obligated" and "Capability" semantic frames is mechanized using lookup tables, instead of NLP tools. We expect to use NLP tools like SEMAFOR in the future, but the current manual annotation process is simple enough to avoid NLP tool usage. Even if fully automated sentence parsing were possible using NLP tools, classification errors would still require manual review. Since for widely used regulatory texts and licenses, the modeling activities would be performed much less frequently than using the model for analysis, the impact of NLP for sentence parsing would be limited. Instead, we have emphasized automation in reasoning with norm models, which could be much more error-prone if done manually.

The use of problem-driven slicing and a "lazy learning" approach [3], where we explore the interpretation of a text as needed by the scenario being examined, also allows initial manual sentence parsing to be practical and scalable. Finally, we use an intermediate (JSON) representation that allows the models to be encoded visually and syntactically using a variety of formats.

Several researchers have proposed techniques for extracting concepts from legal text. Breaux and Anton's FBRAM [13] enables the systematic extraction of semi-formal representations of requirements from regulations using custom frames. A manual annotation process is used to annotate a regulatory document, and a tool is used to parse the annotations to extract the corresponding regulatory requirements. Cerno [30] and its extension GaiusT [48] use a structural pattern matching language to add semantic annotations to legal sentences and identify rights and obligations. Breaux et al. [12] have previously extracted rights and obligations from legal texts using three restricted natural language statements. These and others (e.g., Biasiotti, et al. [11], Biagioli, et al. [10], etc.) indicate that extraction of legal concepts can be achieved by using a small set of underlying semantic structures, modeled around provisions that are mainly obligations, permissions and prohibitions. These semantic structures can also be represented as FrameNet frames, as in Venturi, et al. [44]. Camilleri, et al. have created a controlled natural language to verbalize contract-oriented diagrams which highlight the hierarchical and sequential dependencies among contract clauses [16]. In our work, we rely on the occurrence of lexical units in legal text to simplify the identification of appropriate semantic frames for each atomic sentence. Mapping of semantic frames to norm model elements limits the expertise required for modeling activities. Rather than using the frames directly from frame-semantic parsing, we incorporated the frame elements into simpler norm model templates and mainly used the frame elements to guide the identification of norm model elements. Furthermore, each atomic legal sentence can be modeled and reviewed in the context of a single module. Practical representation and analysis of contractual rights and duties has been an overarching priority of this work to allow its democratization.

Legal and regulatory documents present some opportunities not commonly present in generic natural language text. Lau [33] observes that regulatory documents are hierarchical, are heavily cross-referenced and have essential terms clearly defined. The hierarchical nature implies a tree-like organization of the text, with each subpart providing a finer-grained specification of the statement above it. Cross-referencing allows a level of separation of concerns, with details not relevant to the concerns of the current section being usually referred to other parts of the document by their authors. The definition of essential terms makes it possible

to use and refer to such terms, e.g., "*covered entity*" in HIPAA, consistently throughout a document. We exploited the hierarchical structure of regulatory documents to derive the modular extension to Nòmos 2, connecting a norm to its subparts as antecedent and consequent super-situations. Cross-referencing is handled by tagging a situation with the associated cross-reference identifier. This identifier can then be expanded by the modeler as needed as a separate super-situation. We also made use of the terminology definitions to identify potential actors who can play the holder and beneficiary roles within each norm.

A key goal of analysis of legal text is to transform it into a model or language that is amenable for automated analysis. The Nòmos framework [28] modeled laws to determine applicability and satisfiability. Maxwell and Antón [36] use Prolog production rules to model HIPAA regulations to check the validity or implication of various assertions. Breaux and Gordon's LRSL [14] is used to facilitate recognizing regulatory specification patterns and analyze the complexity of writing styles for legal documents. With modular norm models, our goal is to support incremental reasoning about compliance with duties or exercisable capability of rights as stated in legal text. By systematically avoiding the use of relationships that lead to non-monotonic reasoning, we have enabled reasoning capabilities using widely used OWL representations and rule engines. Our empirical studies with novices and experts show that our proposed reasoning truth tables are easily understood and interpreted.

There is recent work in expressing contractual obligations directly in programming language, for example, smart contracts [6]. Norm models can be used here to help understand the contractual policies being encoded in smart contracts. Conversely, semantic analysis of smart contracts [9, 23], whose goal is vulnerability detection and correctness, can be applied to norm models to identify conditions not covered by the policies.

9 Conclusion

We presented a modular approach for practical representation and analysis of contractual rights and obligations. This method is targeted toward developers and other stakeholders in the software development lifecycle. Using specific problem-driven slices of legal documents, we present a modularized representation of their norms pertinent to the query at hand. The mappings developed between linguistic structures, norm meta-model elements and atomic fragments of Law support a streamlined process of converting legal text into structured models. To better understand the computational characteristics of the developed models, we formalized its logic using two different automated reasoning systems. From an empirical standpoint, we examined the applicability of

modular norm models in the open source domain and privacy domains. Controlled experiments examine the accuracy, time taken and level of confidence in interpreting a legal excerpt by novice developers using modular norm models. Although it takes longer to interpret the norm models, we observed increased accuracy and equivalent confidence levels. The tool pipeline developed for this study is open sourced and available to other researchers and practitioners to replicate and extend. Finally, we have collected valuable feedback from a group of practitioners including lawyers and technologists with many years of experience with software licenses and enterprise-wide open source software audits.

The experiments show promise in the web-based and automated reasoning tool, in the context of open source and privacy domains. These domains have been the target of a number of research efforts in the requirements engineering community, where we expect our results to be useful. Previously, Hohfeldian primitives [25], Nòmos 2 norm models [28, 41] and norm extraction rules by Ghanavati et al. [21] have been applied to a much broader set of legal documents. By extension, we expect the modular norm modeling method to also be broadly applicable. The logical and inference consistency demonstrated by OWL + SWRL reasoning as well as DLV-based implementation discussed in Sect. 5 provides additional assurances.

Lessons learned from formalization, controlled experiments and expert feedback continue to drive our ongoing and future work. Not in any particular order, these include (1) usability studies that inform better interactions with norm models in the context of the corresponding legal text; (2) examine the application of additional computational linguistics approaches and the use of NLP tools to assist analysts in the extraction of norm models; (3) explore the applicability of modular norm models to other types of Hohfeld legal rights (power, immunity and privilege); and (4) further experimentation with a larger and more diverse user population and with a larger corpus of legal documents.

Appendix A: semantic web-based formalization using SWRL rules

We make use of Semantic Web representations for the formalization of norm models. In Sect. 5, we described the use of OWL and SWRL as a way to operationalize the truth tables in Sect. 4. SWRL rules are written in the form of an implication between an antecedent (body) and consequent

Table 21 SWRL rules for duty and right compliance values

Duty		Satisfiability		
		ST	SF	SU
Applicability	AT	Com	Vio	Vio
	AF	Com	Com	Com
	AU	Inc	Inc	Inc

Let *?z* be a SWRL variable representing a norm:

$Duty(?z), applicable(?z, "AT"), satisfied(?z, "ST") \rightarrow Com(?z, true)$

$Duty(?z), applicable(?z, "AT"), satisfied(?z, "SF") \rightarrow Vio(?z, true)$

$Duty(?z), applicable(?z, "AT"), satisfied(?z, "SU") \rightarrow Vio(?z, true)$

$Duty(?z), applicable(?z, "AF") \rightarrow Com(?z, true)$

$Duty(?z), applicable(?z, "AU") \rightarrow Inc(?z, true)$

Right		Satisfiability		
		ST	SF	SU
Applicability	AT	Exr	Exr	Exr
	AF	Nex	Nex	Nex
	AU	Inc	Inc	Inc

$Right(?z), applicable(?z, "AT") \rightarrow Exr(?z, true)$

$Right(?z), applicable(?z, "AF") \rightarrow Nex(?z, true)$

$Right(?z), applicable(?z, "AU") \rightarrow Inc(?z, true)$

Table 22 SWRL rules for computing satisfiability and applicability of a norm from its related situations

Situation A −activate→	Norm
ST	AT
SF	AF
SU	AU

Let *?z* and *?a* represent a norm and a situation, respectively:

$Norm(?z), Situation(?a), activate(?a, ?z), satisfied(?a, "ST"), DifferentFrom (?a, ?z) \rightarrow applicable(?z, "AT")$

$Norm(?z), Situation(?a), activate(?a, ?z), satisfied(?a, "SF"), DifferentFrom (?a, ?z) \rightarrow applicable(?z, "AF")$

$Norm(?z), Situation(?a), activate(?a, ?z), satisfied(?a, "SU"), DifferentFrom (?a, ?z) \rightarrow applicable(?z, "AU")$

Situation A −satisfy→	Norm
ST	ST
SF	SF
SU	SU

$Norm(?z), Situation(?a), satisfy(?a, ?z), satisfied(?a, "ST"), DifferentFrom (?a, ?z) \rightarrow applicable(?z, "ST")$

$Norm(?z), Situation(?a), satisfy(?a, ?z), satisfied(?a, "SF"), DifferentFrom (?a, ?z) \rightarrow applicable(?z, "SF")$

$Norm(?z), Situation(?a), satisfy(?a, ?z), satisfied(?a, "SU"), DifferentFrom (?a, ?z) \rightarrow applicable(?z, "SU")$

(head), that is, if the body is true, the head is true. The OWL elements in Table 8 are used as predicates in the rules, as explained in the following tables. The rules demonstrate how the satisfiability, applicability and compliance values propagate through inference (Tables 21, 22, 23 and 24).

Appendix B: JSON-based representation of norm models

For a tool-friendly representation, we encode each super-situation in modular norm models as a JSON object [29] with cross-references to other super-situations as needed. Figure 17 shows the JSON object representation of SS_AGPL2 super-situation described in Sect. 3. This JSON object has an additional "id" property to facilitate cross-referencing. The first element in the precondition array identifies a logical expression that combines all other preconditions stated in the sentence. This list may include other super-situations or atomic situations. In Fig. 17, precondition P1 is an atomic situation as evidenced by its simple text description only. In contrast, P2–P5 are super-situations that are fully defined elsewhere and referenced using their ids prefixed with a dollar symbol. As discussed before, if required, P1 may be modeled as a super-situation that contains other norms in Section 1 of AGPL.

We also include an additional property, "negative," in the JSON object. This field indicates if the modal has a negation

Table 23 SWRL rules for computing satisfiability of AND, OR and NOT logical situations

Situation A	Situation B	Logical Situation (AND)
ST	ST	ST
ST	SF	SF
ST	SU	SU
SF	SF	SF
SF	SU	SF
SU	SU	SU

Let ?a, ?b represent situations and ?o represent a logical situation:

Situation(?a), Situation(?b) , AND(?o), and_(?a, ?o), and_(?b, ?o), satisfied(?a, "ST"), satisfied(?b, "ST"),
DifferentFrom (?a, ?b), DifferentFrom(?a, ?o), DifferentFrom(?b, ?o) → satisfied(?o, "ST")

Situation(?a), Situation(?b) , AND(?o), and_(?a, ?o), and_(?b, ?o), satisfied(?a, "ST"), satisfied(?b, "SF"),
DifferentFrom (?a, ?b), DifferentFrom(?a, ?o), DifferentFrom(?b, ?o) → satisfied(?o, "SF")

Situation(?a), Situation(?b) , AND(?o), and_(?a, ?o), and_(?b, ?o), satisfied(?a, "ST"), satisfied(?b, "SU"),
DifferentFrom (?a, ?b), DifferentFrom(?a, ?o), DifferentFrom(?b, ?o) → satisfied(?o, "SU")

Situation(?a), Situation(?b) , AND(?o), and_(?a, ?o), and_(?b, ?o), satisfied(?a, "SF"), satisfied(?b, "SF"),
DifferentFrom (?a, ?b), DifferentFrom(?a, ?o), DifferentFrom(?b, ?o) → satisfied(?o, "SF")

Situation(?a), Situation(?b) , AND(?o), and_(?a, ?o), and_(?b, ?o), satisfied(?a, "SF"), satisfied(?b, "SU"),
DifferentFrom (?a, ?b), DifferentFrom(?a, ?o), DifferentFrom(?b, ?o) → satisfied(?o, "SF")

Situation(?a), Situation(?b) , AND(?o), and_(?a, ?o), and_(?b, ?o), satisfied(?a, "SU"), satisfied(?b, "SU"),
DifferentFrom (?a, ?b), DifferentFrom(?a, ?o), DifferentFrom(?b, ?o) → satisfied(?o, "SU")

Situation A	Situation B	Logical Situation (OR)
ST	ST	ST
ST	SF	ST
ST	SU	ST
SF	SF	SF
SF	SU	SU
SU	SU	SU

Situation(?a), Situation(?b) , OR(?o), or_(?a, ?o), or_(?b, ?o), satisfied(?a, "ST"), satisfied(?b, "ST"),
DifferentFrom (?a, ?b), DifferentFrom(?a, ?o), DifferentFrom(?b, ?o) → satisfied(?o, "ST")

Situation(?a), Situation(?b) , OR(?o), or_(?a, ?o), or_(?b, ?o), satisfied(?a, "ST"), satisfied(?b, "SF"),
DifferentFrom (?a, ?b), DifferentFrom(?a, ?o), DifferentFrom(?b, ?o) → satisfied(?o, "ST")

Situation(?a), Situation(?b) , OR(?o), or_(?a, ?o), or_(?b, ?o), satisfied(?a, "ST"), satisfied(?b, "SU"),
DifferentFrom (?a, ?b), DifferentFrom(?a, ?o), DifferentFrom(?b, ?o) → satisfied(?o, "ST")

Situation(?a), Situation(?b) , OR(?o), or_(?a, ?o), or_(?b, ?o), satisfied(?a, "SF"), satisfied(?b, "SF"),
DifferentFrom (?a, ?b), DifferentFrom(?a, ?o), DifferentFrom(?b, ?o) → satisfied(?o, "SF")

Situation(?a), Situation(?b) , OR(?o), or_(?a, ?o), or_(?b, ?o), satisfied(?a, "SF"), satisfied(?b, "SU"),
DifferentFrom (?a, ?b), DifferentFrom(?a, ?o), DifferentFrom(?b, ?o) → satisfied(?o, "SU")

Situation(?a), Situation(?b) , OR(?o), or_(?a, ?o), or_(?b, ?o), satisfied(?a, "SU"), satisfied(?b, "SU"),
DifferentFrom (?a, ?b), DifferentFrom(?a, ?o), DifferentFrom(?b, ?o) → satisfied(?o, "SU")

Situation A	Logical Situation (NOT)
ST	SF
SF	ST
SU	SU

Situation(?z), LNOT(?o), satisfied(?a, "ST"), not_(?a, ?o) → satisfied(?o, "SF")
Situation(?z), LNOT(?o), satisfied(?a, "SF"), not_(?a, ?o) → satisfied(?o, "ST")
Situation(?z), LNOT(?o), satisfied(?a, "SU"), not_(?a, ?o) → satisfied(?o, "SU")

present ("yes") or not ("no"). For the sentence in Fig. 17, the modal "may" does not have a negation, so the value of the "negative" field is "no." The modal field combined with the negative field help justify associate with the appropriate semantic frame. The modal "may," with no negation indicates a claim-right. P1 is an atomic situation, but P2–P5 represent super-situations with "duty" norms in sentences (a) through (d). These represent super-situations "SS_AGPL2a," "SS_AGPL2b," "SS_AGPL2c," and "SS_AGPL2d" as shown in Fig. 6. Using the JSON representation, we have developed scripts that generate a GraphViz [20] visualization as well an OWL (Web Ontology Language) [7] representation automatically. The transformed Graphviz and OWL files can be found here: (https://github.com/robinagandhi/modularnorms) (Fig. 17).

Table 24 SWRL rules for computing truth value of a super-situation

Duty —contained by→	Super-Situation	Let *?z* and *?a* represent a duty norm and a super-situation, respectively:
		$Duty(?z), SuperSituation(?a), contains(?a, ?z), Com(?z, true), DifferentFrom (?a, ?z) \rightarrow satisfied(?a, "ST")$
		$Duty(?z), SuperSituation(?a), contains(?a, ?z), Vio(?z, true), DifferentFrom (?a, ?z) \rightarrow satisfied(?a, "SF")$
Com	ST	$Duty(?z), SuperSituation(?a), contains(?a, ?z), Inc(?z, true), DifferentFrom (?a, ?z) \rightarrow satisfied(?a, "SU")$
Vio	SF	
Inc	SU	
Right —contained by→	Super-Situation	$Right(?z), SuperSituation(?a), contains(?a, ?z), Exr(?z, true), DifferentFrom (?a, ?z) \rightarrow satisfied(?a, "ST")$
		$Right(?z), SuperSituation(?a), contains(?a, ?z), Nex(?z, true), DifferentFrom (?a, ?z) \rightarrow satisfied(?a, "SF")$
Exr	ST	$Right(?z), SuperSituation(?a), contains(?a, ?z), Inc(?z, true), DifferentFrom (?a, ?z) \rightarrow satisfied(?a, "SU")$
Nex	SF	
Inc	SU	

[-] Object, 1 property

table	[-] Object, 8 properties	
	id	AGPL2
	sentence	2. You may modify your copy or copies of the Program or any portion of…
	actor	You
	modal	may
	negative	no
	verb	copy
	action	modify your copy or copies of the Program or any portion of it, … an…
	preconditions	[-] Array, 6 items

0	P1 and (P2 and (P3 and (P4 and P5)))
1	P1: under the terms of Section 1
2	P2: $AGPL2a
3	P3: $AGPL2b
4	P4: $AGPL2c
5	P5: $AGPL2d

Fig. 17 AGPL Section 2 statement JSON object

References

1. Abend O, Rappoport A (2017) The state of the art in semantic representation. In: Proceedings of the 55th annual meeting of the association for computational linguistics, pp 77–89
2. Affero (2002) Affero general public license v1.0. https://spdx.org/licenses/AGPL-1.0.html. Accessed 15 Aug 2018
3. Ashley KD, Rissland EL (2003) Law, learning and representation. Artif Intell 150(1–2):17–58
4. Baker CF, Fillmore CJ, Lowe JB (1998) The Berkeley FrameNet project. In: Proceedings of the 17th international conference on computational linguistics-volume 1. Association for Computational Linguistics, 1998
5. Banarescu L, Bonial C, Cai S, Georgescu M, Griffitt K, Hermjakob U, Knight K, Koehn P, Palmer M, Schneider N (2013) Abstract meaning representation for sembanking. In: Proceedings of the 7th linguistic annotation workshop and interoperability with discourse, pp 178–186
6. Bartoletti M, Pompianu L (2017) An empirical analysis of smart contracts: platforms, applications, and design patterns. In: Brenner M, Rohloff K, Bonneau J, Miller A, Ryan PYA, Teague V, Bracciali A, Sala M, Pintore F, Jakobsson M (eds) Financial cryptography and data security. Springer, Cham, pp 494–509
7. Bechhofer S (2009) OWL: web ontology language. In: Liu L, Özsu MT (eds) Encyclopedia of database systems. Springer, Boston, pp 2008–2009
8. Bench-Capon TJ, Visser PR (1997) Open texture and ontologies in legal information systems. In: Proceedings, eighth international workshop on database and expert systems applications. IEEE, pp 192–197
9. Bhargavan K, Delignat-Lavaud A, Fournet C, Gollamudi A, Gonthier G, Kobeissi N, Kulatova N et al (2016) Formal verification of smart contracts: short paper. In: Proceedings of the 2016 ACM workshop on programming languages and analysis for security. ACM, pp 91–96
10. Biagioli C, Francesconi E, Passerini A, Montemagni S, Soria C (2005) Automatic semantics extraction in law documents. In: Proceedings of the 10th international conference on artificial intelligence and law. ACM, pp 133–140
11. Biasiotti MA, Tiscornia D (2011) Legal ontologies: the linguistic perspective. In: Sartor G, Casanovas P, Biasiotti M, Fernández-Barrera M (eds) Approaches to legal ontologies. Springer, Netherlands, pp 143–166
12. Breaux TD, Vail MW, Anton AI (2006) Towards regulatory compliance: extracting rights and obligations to align requirements with regulations. In: Proceedings of the 14th IEEE

international requirements engineering conference (RE'06), Minneapolis/St. Paul, MN, pp 49–58

13. Breaux TD, Antón AI (2007) A systematic method for acquiring regulatory requirements: a frame-based approach. In: RHAS-6, Delhi, India

14. Breaux TD, Gordon DG (2013) Regulatory requirements traceability and analysis using semi-formal specifications. In: REFSQ 2013, pp 141–157

15. Brüninghaus S, Ashley KD (2001) Improving the representation of legal case texts with information extraction methods. In: Proceedings of the 8th international conference on artificial intelligence and law. ACM, pp 42–51

16. Camilleri JJ, Paganelli G, Schneider G (2014) A CNL for contract-oriented diagrams. In: Workshop on controlled natural language (CNL 2014), volume 8625 of LNCS. Springer, pp 135–146

17. Das D et al (2014) Frame-semantic parsing. Comput Linguist 40(1):9–56

18. Edmundson WA (2004) An introduction to rights, series Cambridge introductions to philosophy and law. Cambridge University Press, Cambridge

19. Fillmore CJ (1982) Frame semantics. In: The Linguistic Society of Korea (ed) Linguistics in the morning calm, pp 111–137

20. Gansner ER, North SC (2000) An open graph visualization system and its applications to software engineering. Softw Pract Exp 30(11):1203–1233

21. Ghanavati S, Amyot D, Rifaut A (2014) Legal goal-oriented requirement language (legal GRL) for modeling regulations. In: 6th international workshop on modeling in software engineering. ACM, pp 1–6

22. Gildea D, Jurafsky D (2002) Automatic labeling of semantic roles. Comput Linguist 28(3):245–288

23. Grishchenko I, Maffei M, Schneidewind C (2018) A semantic framework for the security analysis of Ethereum smart contracts. In: International conference on principles of security and trust. Springer, pp 243–269

24. HIPAA (1996) Health insurance portability and accountability act of 1996, Pub. L. No. 104-191, 110 Stat. 1936

25. Hohfeld WN (1913) Some fundamental legal conceptions as applied in judicial reasoning. Yale Law J 23:16–59

26. Horrocks I, Patel-Schneider PF, Boley H, Tabet S, Grosof B, Dean M (2004) SWRL: a semantic web rule language combining OWL and RuleML. W3C Memb Submiss 21:79

27. Horrocks I (2008) Ontologies and the semantic web. Commun ACM 51(12):58–67

28. Ingolfo S, Siena A, Perini A, Susi A, Mylopoulos J (2013) Modeling laws with Nòmos 2. In: 6th international workshop on RE and LAW (RELAW), 2013

29. JSON, "Introducing JSON," http://www.json.org/

30. Kiyavitskaya N, Zeni N, Cordy JR, Mich L, Mylopoulos J (2009) Cerno: light-weight tool support for semantic annotation of textual documents. Data Knowl Eng 68(12):1470–1492

31. Klymkowsky MW, Taylor LB, Spindler SR, Garvin-Doxas RK (2006) Two dimensional implicit confidence tests as a tool for recognizing student misconceptions. J Coll. Sci. Teach 36(3):44

32. Knublauch H, Fergerson RW, Noy NF, Musen MA (2004) The Protégé OWL plugin: an open development environment for semantic web applications. In: International semantic web conference, pp 229–243

33. Lau G, Law KH, Wiederhold G (2005) Analyzing government regulations using structural and domain information. IEEE Comput 38(12):70–76

34. Leone N, Pfeifer G, Faber W, Eiter T, Gottlob G, Perri S, Scarcello F (2006) The DLV system for knowledge representation and reasoning. ACM Trans Comput Logic (TOCL) 7(3):499–562

35. Lewis M, Steedman M (2013) Combined distributional and logical semantics. Trans Assoc Comput Linguist 1:179–192

36. Maxwell JC, Antón A (2009) Checking existing requirements for compliance with law using a production rule model. In: RELAW

37. McCarty LT (2007) Deep semantic interpretations of legal texts. In: Proceedings of the 11th international conference on artificial intelligence and law. ACM, pp 217–224

38. Peters W, Wyner AZ (2016) Legal text interpretation: identifying Hohfeldian relations from text. In: Proceedings of the tenth international conference on language resources and evaluation LREC 2016. http://www.lrec-conf.org/proceedings/lrec2016/summaries/253.html

39. Mandal S, Gandhi R, Siy H (2015) Semantic web representations for reasoning about applicability and satisfiability of federal regulations for information security. In: 2015 IEEE eighth international workshop on requirements engineering and law (RELAW). IEEE, pp 1–9

40. Reddy S, Täckström O, Collins M, Kwiatkowski T, Das D, Steedman M, Lapata M (2016) Transforming dependency structures to logical forms for semantic parsing. Trans Assoc Comput Linguist 4:127–140

41. Siena A, Jureta I, Ingolfo S, Susi A, Perini A, Mylopoulos J (2012) Capturing variability of law with Nòmos 2. In: Conceptual modeling (ER 2012), LNCS 7532. Springer Berlin, pp 383–396

42. Sirin E, Parsia B, Grau BC, Kalyanpur A, Katz Y (2007) Pellet: a practical OWL-DL reasoner. Web Semant Sci Serv Agents World Wide Web 5(2):51–53

43. Sutcliffe A, Maiden N (1998) The domain theory for requirements engineering. IEEE Trans Softw Eng 24(3):174–196

44. Venturi G et al (2009) Towards a FrameNet resource for the legal domain. In: LOAIT, pp 67–76

45. Waddell W (1961) Structure of laws as represented by symbolic methods. Arts and Crafts Press, Tacoma

46. Wenar L (2015) "Rights", The Stanford Encyclopedia of Philosophy (Fall 2015 edition). In: Zalta E (ed) http://plato.stanford.edu/archives/fall2015/entries/rights/. Accessed 15 Aug 2018

47. Zeni N, Seid EA, Engiel P, Mylopoulos J (2018) NómosT: building large models of law with a tool-supported process. Data Knowl Eng 117:407–418

48. Zeni N, Kiyavitskaya N, Mich L, Cordy JR, Mylopoulos J (2013) GaiusT: supporting the extraction of rights and obligations for regulatory compliance. Requir Eng 20:1–22