Making Sense of Regulations with SBVR

Marcello Ceci, Firas Al Khalil, Leona O'Brien

Governance, Risk, and Compliance Technology Center
University College Cork
13 South Mall, Cork, Ireland
{marcello.ceci,firas.alkhalil,leona.obrien}@ucc.ie

Abstract - The continuous increase in quantity and depth of regulation following the financial crisis has left the financial industry in dire need of making its compliance assessment activities more effective. The field of AI & Law provides models that, despite being fit for the representation of semantics of requirements, do not share the approach favoured by the industry which relies on business vocabularies such as SBVR. This paper presents Mercury, a solution for representing the requirements and vocabulary contained in a regulatory text (or business policy) in a SME-friendly way, for the purpose of determining compliance. Mercury includes a structured language based on SBVR, with a rulebook, containing the regulative and constitutive rules, and a vocabulary, containing the actions and factors that determine a rule's applicability and its legal effect. Mercury includes an XML persistence model and is mapped to an OWL ontology called FIRO, enabling semantic applications.

Keywords. AI & Law, GRC, SBVR, statutory interpretation, factors, fintech.

1 Introduction

Ensuring compliance with new regulatory requirements as well as with existing ones continues to be a challenge of herculean proportions for the international financial industry, driven by the proliferation and complexity of the financial regulatory environment in the aftermath of the global financial crisis. The wide acceptance in the industry that traditional Governance, Risk, and Compliance (GRC) information systems are deficient is leading to a growing interest in semantic technologies. The research at the Governance, Risk and Compliance Technology Centre (GRCTC) is a complementary set of projects tackling the challenges for the Financial Industry that arise from multi-layered, multi-jurisdictional and ever-changing regulation.

Understanding regulations is a complex task for both non-trained human agents and machines. Legal experts face a number of challenges in interpreting a regulatory text, including: following and fleshing out references and citations; identifying, delineating, and disambiguating definitions; making sense of complex sentences; clarifying ambiguities resulting from legalese; accounting for exceptions (Abi-Lahoud et al. 2014a). The research presented in this paper devises an approach to bridge the gap between the legal expertise required to interpret the regulatory text and the modeling skills required to build a semantic knowledge base. The goal is to foster compliance in the financial sector by supporting corporate lawyers, risk practitioners and compliance professionals in their role of SMEs in making law more readily consumable and comprehensible by the industry, but differently from the Ergo platform (Grosof et al. 2015). The process of translation is articulated around a Regulatory compliance Interpretation Methodology (RIM) that defines a process for transforming a regulatory text into a representation in structured English *en route* to enabling semantic applications. The process is envisaged as a collaborative one, involving the legal expert as a Subject Matter Expert (SME) and the modeler as a Semantic Technology Expert (STE) through multiple iterations.

The solution allows the SME to represent the semantics of regulatory requirements in a machine-readable format through a SME-friendly process. This is ensured through the use of SBVR, a OMG specification based on formal logics and well known to the industry. In SBVR, a requirement is re-written in Structured English, a loosely structured Controlled Natural Language where every term used in the rules is sourced and specified in a terminological dictionary. SBVR is a powerful instrument for modeling an area of business activity and for building a business vocabulary (van Haarst 2013, p. 14), but it is not suitable – as is – for the representation of legal rules; some SBVR components are not needed, or overcomplicate the task of rule representation (e.g. the logical formulation of a sentence), and some components are lacking.

To overcome this, GRCTC research has devised an artifact called Mercury, composed of a structured English based on SBVR and persisted in an XML Schema capturing rulebook and vocabulary entries. Mercury represents rule statements contained in regulations and describes the concepts used in those rules in a terminological dictionary. The most important change that Mercury brings to SBVR – and the object of this paper – is a different way to manage the logical formulation of the SBVR rule that overcomes the limitations of SBVR original logical formulation in representing legal interpretation. The financial regulations to be represented for GRC purposes are often very detailed, containing requirements rather than high-level legal principles or general prescriptions. Therefore, for their interpretation value judgments (Bench-Capon and Sartor 2003, Grabmair and Ashley 2011) play a minor role in comparison to the linguistic element. A significant part of the relations between the regulated entities is in fact specified in the rule itself, and the important concepts are explicitly defined in the same regulation. SBVR, with its vocabulary and rulebook components, seems best suited to capture this.

The current phase of the research does not involve advanced rule-based reasoning but only rule representation. The aim is to capture relevant information on the regulatory requirements and to be able to:

- Run queries on the resulting knowledge base;
- Perform abstract classification and reasoning on rules and their regulated actions (e.g. detecting which rules regulate a subset of another rule's regulated action);
- Validate data representing instances of regulated actions (events) as compliant or breaching one or more rules.

Mercury exploits technologies from the Semantic Web (SW) stack at the XML layer as well as non-SW technologies (SBVR and the RIM). It relies on upper SW layers, particularly OWL, for advanced classification and reasoning on rules and vocabulary. (Al Khalil et al. 2016) for an explanation of GRCTC's mapping from SBVR to OWL.

The rest of the paper is structured as follows: Section 2 describes statutory interpretation and the issues related to its representation with SBVR, Section 3 introduces Mercury and its main elements, and Section 4 explains its model for representing regulative rules. Section 5 concludes with final remarks, a proof of concept, and the further steps of the research.

2 The Semantics of Regulatory Requirements

The present paper addresses the AI & Law subjects of statutory interpretation and rule representation. The concept of legal interpretation, highly debated in legal philosophy (Aracziewicz 2013), consists in our account in determining the extension of statutory terms. We thus focus on the part of legal interpretation concerned with the conceptual analysis, typical of domestic law in civil law countries.

Since its inception, the AI & Law community has devoted notable efforts in modeling legal rules and regulations (Sergot et al. 1986, Pashke et al. 2007), but only few of them were aimed at devising an intermediate format for describing legal rules, the Legal Knowledge Interchange Format (LKIF) (Boer et al. 2008) being a first systematic attempt in this direction. Most of the research in the field was instead directed towards the representation of legislative documents (the speech acts rather than the provisions they contain) through XML languages, among which we cite CEN MetaLex (Boer et al. 2009) and Akoma Ntoso (Vitali and Zeni 2007). The present research relies on Akoma Ntoso to enhance the semantics of the legal source, enabling Regulatory Change Management, temporal reasoning, and standard representation of metadata such as jurisdiction and issuing authority (O'Brien et al. 2016).

Legal rules are mostly debated in AI & Law in relation to the notions of legal argumentation (Prakken 2008) and case-based legal reasoning (see Section 3.2). The problem of statutory interpretation is relatively underrepresented in AI & Law research, with recent exceptions from the civil law tradition (Araszkiewicz 2013). The research showed notable results, e.g. Gordon et al. (2009) setting the requirements for the representation of legal rules and Wyner and Peters (2011) covering the automatic extraction of rules from regulation exploiting NLP techniques. LegalRuleML has all the desired features to capture legal regulation, but it does not rely on a terminological dictionary like SBVR, and lacks specific focus on compliance.

2.1 Case-based Legal Reasoning

Legal reasoning can be classified in two main categories: *abstract* reasoning on the relationship between rules and/or the concepts they express, and *concrete* reasoning on the rules applicable to a given fact. The first type of reasoning is the *Rechtsdogmatik* and is typical (but not exclusive) of civil law systems, and is also called *doctrinal legal interpretation* (Wroblewski 1992). The second type of reasoning corresponds to *case-*

based legal reasoning and is typical (but not exclusive) of common law systems, and is also called *operative legal interpretation*.

Case-based legal reasoning has made for very prolific research in AI & Law. The original purpose, especially in the works of K. D. Ashley (1990) and his scholars (Aleven 1997) was to support legal classes providing a way to explore concepts such as analogy and precedent. Their research resulted in two computer programs aimed at representing case-based argumentation in a computable way:

- The first, HYPO, enables comparison between precedents by means of dimensions: characteristics of the precedents are thus represented as points (positions) within gradients which are not themselves built to favor one solution to the case, rather strengthening or wakening the claim of each party to the dispute according to the position in the dimension (Ashley 1990 p. 112).
- The second, CATO, includes a simplified structure of dimensions where particular points (positions) are represented as factors, unequivocally supporting one party to the case. Factors as employed in CATO are *unary*, in the sense that those factors either apply to a given case, or they do not apply (Aleven 1997).

In AI and Law, factors are seen as mere indicators that enable argumentation and comparison without necessarily leading to the promoted judicial outcome, even in absence of reasons to the contrary (Sartor 2005). More recently, research focused on formal models of legal argumentation, with analytical concepts such as dimensions and factors being replaced by more high-level figures such as values (Bench-Capon and Sartor 2003), argumentation schemes (Gordon and Walton 2006), and intermediate legal concepts (Grabmair and Ashley 2011). The use of factors to capture statutory interpretation has been already discussed (Araszkiewicz 2011, Ceci and Gangemi 2016), and the research presented in this paper builds upon that experience.

2.2 SBVR

The Object Management Group (OMG) created the Semantics of Business Vocabularies and Business Rules specification (SBVR 2013) to define business concepts and rules using a controlled natural language, SBVR Structured English (SBVR SE). It is meant to be used by business people to describe their business activities, hence its adoption in GRCTC: it allows non-technical experts (SMEs) to define rules using a controlled language (as opposed to legalese).

SBVR SE is composed of a rulebook and of a terminological dictionary (vocabulary). The rulebook contains *operative rules*, introduced by deontic modalities (*it is obligatory that*, *it is permitted that*, *it is prohibited that*), and *structural rules*, introduced by alethic modalities (*it is necessary that*, *it is possible that*, *it is impossible that*). These two categories correspond to regulative rules and constitutive rules (Searle 1969). The present paper focuses on operative rules such as the following:

It is obligatory that each market operator **that** *operates* **a** trading venue *makes public* bid price *for* share.

All the terms of the SBVR rule except **keywords** and **modality** (both in **bold** in the rule above) are declared in the SBVR vocabulary. The vocabulary contains entries for **noun concepts** (underlined) and **verb concepts**. Verb concepts include one **verb symbol** (*italic*) and one or more **verb concept roles** (underlined in the entry below) covered by noun concepts (see Figure 1).

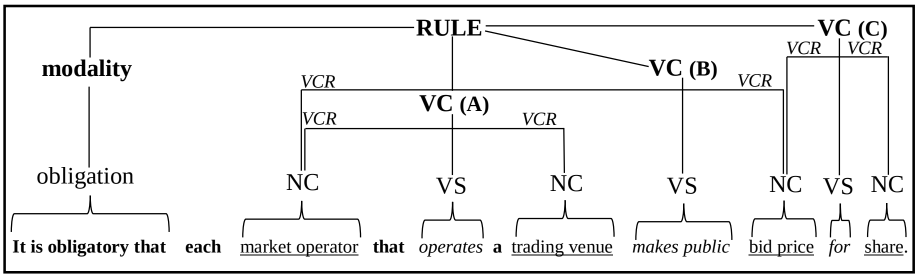

Fig. 1. – Elements of an SBVR rule. NC = Noun Concept, VS = Verb Symbol, VCR= Verb Concept Role, VC= Verb Concept (the letters refer to the table below and to Figure 6).

Let us consider for example the following verb concept entry in the vocabulary:

market operator *operates* trading venue

This verb concept includes one verb symbol, "operates", and two noun concepts, "market operator" and "trading venue", as **verb concept roles**. Each vocabulary entry contains a list of attributes (metadata) for semantic enrichment.

SBVR defines a means to capture the semantics expressed in a rule. Semantic formulations, as described in the SBVR specification, are not representations or expressions of meaning. Rather, they are structures of meaning – the logical composition of meaning. Although adequate, the logical formulation of a sentence is not suitable to represent the composition of a legal rule: it is too verbose and complex. A simpler model to capture the semantics of a legal rule can be more advantageous, since it deals with the legal nature of the rules directly, not in an abstract logical manner.

2.3 Making Sense of SBVR: Introducing Mercury

In order to design a model of legal rules for compliance, we need to define a data format that the machine can use to perform such type of reasoning, starting from what an SME can be expected to achieve: the SBVR-SE format of the rule. In Mercury (Figure 2), regulative rules are thus presented as SBVR sentences, with an operative modality and a net of verb concepts, connected by verb concepts roles shared among them. The verb concepts as used in the rule to represent the regulated actions or states of affairs, or, in other words, are the "conditions" of the rule. In the example rule the following verb concepts can be identified:

	Verb Concept Role	Verb Symbol	Verb Concept Role
A	market operator	*operates*	trading venue
B	market operator	*makes public*	bid price
C	share	*has*	bid price

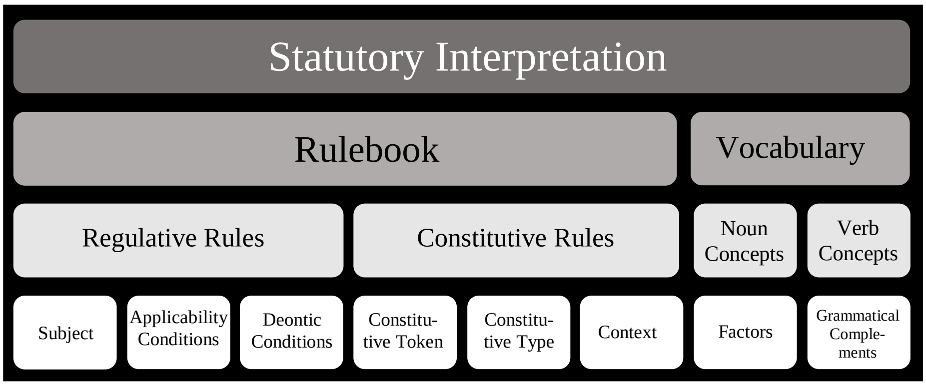

Fig. 2. – The elements of statutory interpretation as represented in Mercury-SE. *Constitutive rules* and its elements are not covered in the present paper.

Each of these verb concepts represent abstract actions that determine relevance under the rule. In LKIF (Boer et al. 2008), rules are seen as qualifiers, and regulated actions or state of affairs as qualified things. Under this perspective, the core semantics of compliance (the semantics of *compliance* and *breach*) can be treated as semantics of a "qualified" entity, as regulative rules qualify actions as "compliant to" or "breaching" the rule itself. In SBVR, a rule's semantics is specified through its verb concepts, and in the noun concept entries that correspond to those verb concepts' roles. Verb concepts thus represent the qualified ("compliant" or "breaching") actions, and verb concept roles are the links that express the semantics of this qualification.

The next section presents Mercury, a bridge language built on top of SBVR to provide the necessary hooks for legal reasoning. Mercury enriches the semantics of SBVR rulebook by introducing the concept of *condition*, a "qualified" verb concept.

3 Mercury: A Bridge Language for Regulatory Requirements

Mercury is a bridge language for translating regulatory requirements in a machine-readable way. It includes two components:

- A structured natural language called **Mercury Structured English** (Mercury-SE, based on SBVR, presented in the next section);
- A persistence model in XML called **Mercury Markup Language** (HgML, presented in Section 4.5).

3.1 Actions, Events, and Factors

Mercury-SE is composed of a rulebook and a vocabulary (terminological dictionary), both similar to their SBVR counterparts. Mercury extends the terminology of SBVR by identifying an additional layer of semantics to the vocabulary entries. Thus, the entries that are defined in SBVR as (*a*) verb concepts and (*b*) verb concept roles express abstract entities that are captured in Mercury under the labels of (*a*) **action** and (*b*) **factor** (see Figure 3).

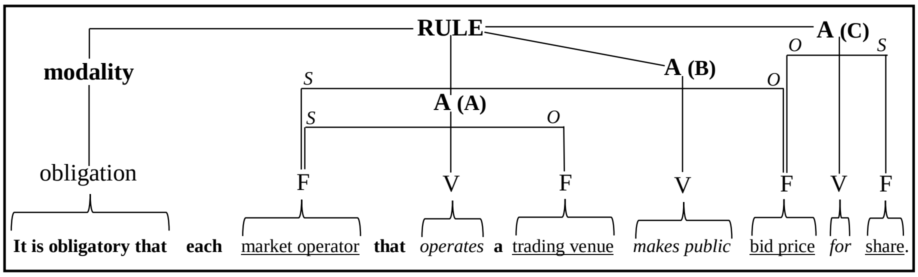

Fig. 3. – An SBVR rule with actions (A) and factors (F) as identified by Mercury. Grammatical complements are also indicated (S = Subject, O = Object).

> ***Definition of Action***: an action is an abstract category of events that is defined arbitrarily. It is the result of the interpretation on the behaviour required by the rule. Actions are expressed by SBVR verb concepts.
>
> ***Definition of Event***. an event is a concrete manifestation of an abstract action.
>
> ***Definition of Factor***: a factor is a (generic or specific) entity that plays a role in one or more actions contained in the same rule. It is a result of the interpretation of the entities involved in the rule. Factors are expressed by SBVR verb concept roles.

3.2 Grammatical Complements and Conditions

For reasons of simplicity, in the present paragraph we will describe a Mercury rule stripped of its deontic modality:

market operator **that** *operates* **a** trading venue *makes public* bid price *for* share.

Each verb symbol (in italics) corresponds to one or more verb concepts, depending on the noun concepts present as verb concept roles and on the keywords. Each of the verb concepts contained in the rule represents an abstract action. The sum of the verb concepts contained in the rule constitutes the structure diagram of that rule (Figure 4).

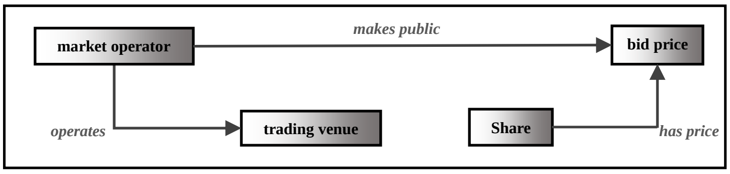

Fig. 4. – Implicit ontology in the sample rule in an SBVR structure diagram (van Haarst 2013, chapter 7). Boxes represent noun concepts; arrows represent verb symbols.

The reader will notice that some verb concept roles are present in more than one verb concept. In Mercury, we say that some factors are present in more than one action. These factors could independently identify different instances of the same category (i.e. two different market operators or two different bid prices) or instead they could identify the same instance (i.e. the same market operator must be the subject of action A and B, the same bid price must be the object of actions B and C). In the latter case, for an action or event to be classified under the more general event the same instance must appear in all factors. E. g. for a market operator (or a category of market operators) to be relevant to the statement above, it is necessary that he operates a trading venue and also that he makes public a bid price for a share. Keywords such as **same as** and **that** determine the identity of factors among different actions in a rule.

With SBVR, the logical formulation of the sentence is too verbose and complex. Being unable to extract the logical formulation, however, means being unable to determine the relevance of events or actions to rule statements. To overcome this limitation, Mercury implements its own solution of expressing the logical formulation. It introduces *grammatical complements*, an attribute of verb concept entries used to describe the grammatical role that verb concept roles play in the sentence.

> ***Definition of grammatical complement*:** a grammatical complement is an attribute of a factor expressing the grammatical role played by it in the verb concept. The grammatical complements in Mercury are: *subject*, *object*, *indirect object*, *location*, *time*, *comparison*, *value*, and *mode*.

With Mercury, every factor is specified under the perspective of the grammatical complement that it covers for that verb concept. So, for example:

	subject	**verb**	**object**
A	market operator	*operates*	trading venue
B	market operator	*makes public*	bid price
C	share	*has*	bid price

Figure 5 shows how these complements enrich the logical formulation. Please note that, despite the example being very simple (all actions having the same complements), verb concepts can vary: they can have only one factor (the subject) or more than two, and logical connectors such as **and**/**or** can duplicate them.

Mercury distinguishes between verb concepts as they appear in the vocabulary and as they appear in the rulebook: in the first case they are called Actions, while in the latter they are called *conditions*. Conditions can be seen as “qualified” actions, where one or more factors of the action correspond to factor(s) in other action(s). Keywords such as **that** or **same** are indicators that a factor is involved in more than one action.

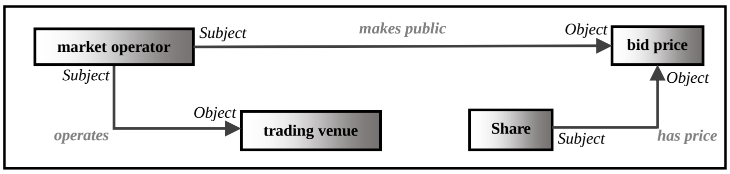

Fig. 5. – Implicit ontology in the sample rule in Mercury formulation.

Definition of Condition: a condition is an action used in a rule. A condition shares the same properties of its general action and may restrict factors by specifying:

1. its scope or value, or
2. The role it plays in another condition (*grammatical complement*).

The following Section describes how Mercury uses actions, factors and conditions to enrich the semantics of regulations.

4 Representing Requirements in Mercury

4.1 Anatomy of a Regulative Norm

According to Biagioli (2009) regulations can be seen as a set of provisions (semantics) carried by speech acts (structure). Performing the legal interpretation of a regulation means in our account to describe those provisions.

From a formal logics perspective, a regulative norm can be represented with the following formula (Kelsen 1991, Sartor 2005) (please note that in formal logics we use the operator "Forb", for "forbidden", instead of "prohibited"):

$$A\ (A_1, A_2, ..., A_n) \rightarrow {}_{[Obl;Forb;Perm]}B$$

The first element, "A", is called **applicability condition** (Gordon et al. 2009). The second element, "${}_{[Obl;Forb;Perm]}B$", is often called legal effect, but for the purposes of the present research it is called **deontic condition**. In our model, this element shares the same structure as the applicability condition, with the addition of a deontic modality.

Definition of Applicability Condition. It is a condition that determines if a given event is relevant to a given rule, or not.

Definition of Deontic Condition. It is a condition that determines if a relevant event complies/breaches a rule.

4.2 Reasoning on Compliance and Breach

In a Mercury rule, abstract compliance is represented through the verb concepts contained in the rule. As explained in Section 3.1, SBVR's verb concepts constitute the **actions** of Mercury, central to the present approach. Its abstract characteristics are described in the attributes of the verb concept in the Mercury vocabulary: an action is thus composed of a verb symbol and one or more **grammatical complements**. The action becomes a **condition** when it is used (as a "deontically qualified action") in a Mercury rule sentence. A rule sentence includes zero or more applicability conditions and one deontic condition. In this way, the abstract compliance to a rule can be described as a sum of conditions, and the way the complements of these conditions intertwine provides fundamental information that unlocks powerful reasoning on compliance, just like CATO's factors (see Section 2.1) enable reasoning on case law

Concrete events (instances of actions) can be classified under the abstract conditions of the Mercury rule using factors. An event is *relevant* for the rule (i.e. the rule applies to the event) if the factors of the event can be classified as factors of the applicability conditions of the rule. In addition, if the factors of the event can be also classified as factors of the deontic condition of the rule, the event is classified depending on the rule's deontic modality: if the rule is an obligation, the event *complies* with it; if the rule is a prohibition, the event *breaches* it. If, instead, the factors of the event do not match all applicability conditions, the event is classified as "not relevant" to that rule (see Figure 6). Concrete compliance to rule is thus similar to matching a new case (the event) to the factors of a precedent (the sum of conditions of the rule).

Because this solution attributes explicit semantics (labels such as *breach*, *compliant*, *relevant*) to the actions, the concept of factor is preferable over dimension: factors (Aleven 1997) are unary concepts, susceptible of attribution of Boolean values, and thus ideal to represent such semantics. In Mercury, factors are represented by the elements of the structured language, i.e. verb concept roles. The semantic value added to them is given not only by their actual value, or by the vocabulary entry of that term, or of the verb concept that contains it, but mostly by the role they play in other verb concepts of the same rule (see next Section).

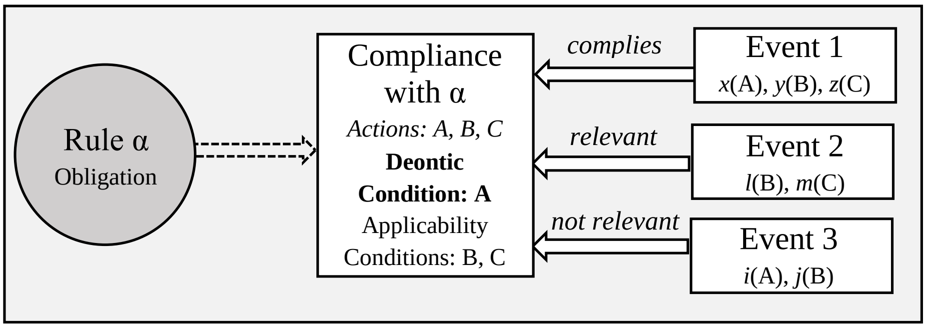

Fig. 6. – Relationship between rules and regulated events. Letters in italics represent events as instances of actions (A, B, C). Those events compose the complex events: 1, 2, 3. Event 1 complies with rule α because it matches all applicability conditions (B and C) and the deontic condition (A); event 2 is relevant but we have no information on compliance because it does not match deontic condition A (on detecting deontic opposites see Section 4.4); event 3 is not relevant (rule α is not applicable) because it does not match action C.

4.3 Representing a Mercury Rule

This section describes the approach for representing operative rules in Mercury. The approach relies on a list of rules and a terminological dictionary to explain the terms used in them. The dictionary is particularly helpful in managing the open-textured character of statutory concepts. For more on this topic see Hart (1961).

The rule is first identified in the source, and re-written in plain English, fleshing out all references, using a limited set of keywords to express logical elements (e.g. each, and, that), resolving syntactic ambiguities by referring every term in the rule to a verb or noun concept in the vocabulary, and indicating, explicitly and at the beginning of the rule, its deontic modality (Abi-Lahoud et al. 2014b). Despite any future implementation of *a contrario* reasoning (see Section 4.4), the modality chosen by the SME will constitute the “main” modality, upon which more straightforward reasoning is possible.

Following is an example of a Mercury rule, with the deontic modality in bold:

It is obligatory that each market operator that operates a trading venue makes public bid price for share.

Representing the conditions of the rule.

As explained in Section 3, **action** designates the vocabulary entry of the verb concept, while **condition** designates the verb concept as it appears in the rule. A condition specifies an action either by (*a*)specifying the value of some of its *factors* (verb concept roles), or (*b*) specifying the role, in terms of *grammatical complement,* that they must cover in other conditions.

The Deontic condition is a particular kind of condition: it describes the action that determines the compliance to the rule. The deontically relevant part of such condition resides in one (or more) of its factors. Such deontically relevant parts must be highlighted by the SME in the rule representation to allow the reasoner to distinguish between applicability conditions and deontic conditions. The part underlined in the rule above (*market operator makes public bid price*) identifies the deontic condition. In order for an event to be classified as compliant to this rule, its factors must correspond to the factors in the other conditions of the rule.

4.4 Deontic Opposites

Reasoning on the direct legal effect of a deontic statement means to detect the events that are compliant to an obligation, or that breach a prohibition, or that are allowed by a permission. In order to enable reasoning on the main deontic modality, it is not necessary to distinguish between deontic and applicability conditions. In this limited scenario, such distinction is only useful to reason about legal relevance of a rule even when the information on the deontic modality is missing (that is, we do not know if the event breaches or complies with the rule because we have no information on the deontic condition, but we know that the event is relevant for the rule because it matches all its applicability conditions). The distinction between applicability and deontic condition is instead important to enable reasoning on the deontic opposite of the rule.

We define a deontic opposite as the obligation implied by a prohibition, or a prohibition implied by an obligation or permission. In legal theory, the deontic opposites are known since Hohfeld’s definition of the deontic qualifications *obligatory* and *prohibited* as complete, in the sense that they determine the deontic status of both (*a*) the action or state of affairs they are concerned with and (*b*) the complement of such action or state of affairs. Hence, to say that the action or state of affairs A is obligatory is equivalent to saying that ¬A is prohibited, and to say that A is prohibited is equivalent to say that ¬A is obligatory (Cook 1918). To reason on a deontic opposite thus means to detect breach of an obligation or permission and compliance to a prohibition. In order to classify such events, the reasoner needs to match them to the “opposite” of the direct deontic condition, for example:

It is prohibited that each *market operator* that operates a trading venue ***does not make public bid price*** for share.

If an event matches such deontic condition, it is classified as a breach.

In the literature concerning the interpretation of statutory text using negation as failure (Sergot et al. 1986) and the *a contrario* argument (Peczenik 2008), legal theory defines the determination of those "deontic opposites" as far from automatic (Araszkiewicz 2011). The process of deriving deontic opposites must thus be a semi-automated process led by the SME. It is one of the intended future developments of the research and not covered further in the present paper.

4.5 Mercury-ML

Mercury-ML is the persistent XML model for Mercury, representing the vocabulary and the rulebook. The following XML covers a short version of our example rule:

```
<?xml version="1.0" encoding="UTF-8"?>
<grctc:mercury xmlns:grctc="http://www.grctc.com/Mercury/20160215"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
...
       <vocabulary>
              <nounConcept id="ID1">
                     <designation>Market Operator</designation>
                     ...
              </nounConcept>
              <nounConcept id="ID2">
                     <designation>Bid Price</designation>
                     ...
              </nounConcept>
              <nounConcept id="ID3">
                     <designation>Share</designation>
                     ...
              </nounConcept>
              <verbConcept id="ID4">
                     <composition>
                            <role id="ID5" ref="ID1" complement="subject"/>
                            <role id="ID6" ref="ID2" complement="object"/>
                            <verb id="ID7">makes public</verb>
                     </composition>
                     ...
              </verbConcept>
              <verbConcept id="ID8">
                     <composition>
                            <role id="ID9" ref="ID3" complement="subject"/>
                            <role id="ID10" ref="ID2" complement="object"/>
                            <verb id="ID11">has</verb>
                     </composition>
                     ...
              </verbConcept>
       </vocabulary>
       <rulebook>
              <legalRules>
                     <obligation id="ID12">
                            <expression>It is obligatory that market operator makes
public bid price for share.</expression>
                            <placeholderList>
                                   <placeholder     startingCharacterPosition="23"
endingCharacterPosition="37" signifier="ID5"/>
                                   <placeholder     startingCharacterPosition="39"
endingCharacterPosition="50" signifier="ID7" deontic=”true”/>
                                   <placeholder     startingCharacterPosition="52"
endingCharacterPosition="61" signifier="ID6"/>
```

```
                                <placeholder     startingCharacterPosition="52"
endingCharacterPosition="61" signifier="ID10"/>
                                <placeholder     startingCharacterPosition="63"
endingCharacterPosition="65" signifier="ID11"/>
                                <placeholder     startingCharacterPosition="67"
endingCharacterPosition="71" signifier="ID9"/>
                          </placeholderList>
                          ...
                    </obligation>
              </legalRules>
        </rulebook>
</grctc:mercury>
```

In this example, we show the markup used to model a rule in Mercury-ML. In the vocabulary section, we declare (through the 'id' attribute) three distinct noun concepts and two verb concepts. The verb concepts define exactly one verb symbol and multiple distinct (through the 'id' attribute) verb roles each of which referencing (through the 'ref' attribute) a previously declared noun concept. The deontic condition is identified with the attribute "deontic" attached to the verb symbol. In the rulebook section, we define a legal rule (an obligation). The rule is written in Mercury-SE (in the expression tag). A rule also has a 'placeholderList' that binds elements of the 'expression' to components of verb concepts, i.e. 'placeholderList' has a list of placeholders, each of which binds (through the 'signifier' attribute) to a verb concept role or a verb symbol.

In our example, the placeholders of rule ID12 bind 'Market Operator', 'makes public', and 'Bid Price' to the subject, verb symbol, and object (ID5, ID7, and ID6 respectively) of the verb concept 'Market Operator makes public Bid Price' (ID4). We also have similar bindings for 'Share has Bid Price' (ID8). Notice that Bid Price has two bindings: one as the object of ID4, and as the object of ID8. This example shows that we can execute, in a straightforward fashion, some complex queries: we can retrieve the rules that include certain actions, specifying how these actions must be concatenated. For example, we can ask for "all rules that have a Bid Price that is the object of [a verb concept] AND the object of [another verb concept]".

5 Conclusions

The present paper describes the legal framework behind the applied research performed by the GRC Technology Centre. The solution devised by the research is Mercury, enabling machine-readable representation of regulatory requirements in a SME-friendly way. Mercury is an extension of a subset of SBVR that allows the SME to maintain the precision of legal knowledge, its specific semantics, and the importance of the legal source when transforming the regulatory text in a format that *(a)* allows specification of the model of business activity and of the business vocabulary in compliance to SBVR, and *(b)* unlocks the potential of semantic web technologies.

On the latter, the GRCTC is currently developing a set of OWL ontologies called FIRO (Financial Industry Regulatory Ontology) to enable semantic applications such as classification, querying and reasoning. The research has also devised a mapping of all SBVR elements relevant to Mercury into OWL, to facilitate the STE in translating Mercury rulebooks and vocabularies. This mapping differs significantly from the literature and results in a different SBVR structure diagram than the one suggested in Figure

4 (van Haarst 2013, chapter 7). The reader is invited to refer to Al Khalil et al. (2016) for further details on FIRO and the mapping. More research efforts are being directed towards Natural Language Processing techniques for automatic extraction of requirements from legal texts (Asooja et al. 2015).

The solutions presented in the present paper could be applied beyond the limit of description logics and OWL. The current phase of the research, however, excludes rule-based reasoning capabilities, limiting itself to rule representation: as noted previously, the principal goal of the research is to exploit the expressive power of the SBVR vocabulary to enhance the semantics of concepts such as requirement, compliance, breach, source, legal definition, and context. Mapping of such semantics from Mercury to LegalRuleML is only partially possible because of Mercury's bipartite structure (rulebook + vocabulary) following SBVR. Mapping is quite straightforward for conditions, less so for factors and the links they establish across conditions, because LegalRuleML lacks a vocabulary.

One may argue that a mapping from the logical formulation, as defined by SBVR, to the Mercury model can be achieved: a correct, yet unpractical observation. The extraction of the legal formulation of a sentence is not evident, and the lack of tooling discouraged us from taking this route. Bypassing the logical formulation proved to be advantageous for both 1) understanding the rule by the SME and the STE due to its simplicity, and 2) mapping legal rules to FIRO (Al Khalil et al. 2016).

The Ergo platform (Grosof et. al. 2015) shares the goal of GRCTC research and has a similar approach, the main differences consisting in the use of a bridge language (Mercury, based on SBVR) and in the creation of a Regulatory Interpretation Methodology, that focuses the attention on the technique of translation as well as on the technologies, thus enabling the semi-automatic and collaborative translation process.

From a legal theory point of view, the research faces the issue of representing statutory interpretation by adopting a solution coming from another area of AI & Law, namely from the concept of factors as used in computational models for lecturing on case-based reasoning. This research therefore treats the computer as a (not particularly brilliant) legal student, which does not grasp the notions of pure legal theory (*Rechtsdogmatik*) but rather reasons in terms of classifying actions and things under abstract legal categories.

A Proof of Concept of the RIM has been conducted on MiFIR (Regulation No. 600/2014 of the European Union), as two SMEs used the protocol to transform articles 3 and 14 into Mercury. The purpose of the proof of concept was to measure the time it takes and the size of the resulting knowledge base, rather than to measure the capabilities of the resulting knowledge base. The two outputs for Article 3 were a document of 12 pages containing 13 rules and 37 vocabulary entries, and a document of 25 pages with 5 rules and 63 vocabulary entries. This difference is caused by the fact that the two SMEs used different structures to represent some complex concepts and articulated requirements. This variation in the output shows the degree of freedom that Mercury allows in the representation of legal concepts, which is clearly desirable, as the opposite would instead increase the risk of inaccurate representation of legal concepts and of loss of details, which in turn leads to (potentially costly) mistakes in the application of the law.

The next steps for the research will involve extensive testing of Mercury from the point of view of its suitability to represent any kind of regulatory statement. The testing will also involve the user experience of the supporting tool (called Ganesha, currently under development) that will accompany the SME during the process of transformation of legal requirements from regulatory text to Mercury. In the future, the application of the legal knowledge base built with Mercury will be extended further up the semantic web stack of technologies, past ontologies into defeasible rule-based reasoning such as LegalRuleML (Palmirani et al. 2011), agent-based reasoning (Boella et al. 2013), and legal argumentation relying e.g. on the Carneades engine (Gordon and Walton 2006).

6 Acknowledgements

This work is financed and supported by Enterprise Ireland (EI) and the Irish Development Authority (IDA) under the Government of Ireland Technology Centre Programme.

7 References

1. Abi-Lahoud, E., O'Brien, L., Butler, T.: On the Road to Regulatory Ontologies: Interpreting Regulations with SBVR. In: AI Approaches to the Complexity of Legal Systems (Vol.8929), Lecture Notes in Artificial Intelligence, Springer Berlin, pp. 188-201 (2014)
2. Abi-Lahoud, E., Butler, T., O'Brien, L.: A Hermeneutic Approach to Solving the Translation Problem in Designing Ontologies. In: 22nd European Conference on Information Systems (ECIS) (2014)
3. Aleven, V.: Teaching Case-Based Argumentation Through a Model and Examples. PhD Dissertation, University of Pittsburgh (1997)
4. Al Khalil, F., Ceci, M., Yapa, K., O'Brien, L.: SBVR to OWL 2 Mapping in the Domain of Legal Rules. In: RuleML 2016 (accepted) (2016)
5. Araszkiewicz, M.: Analogy, Similarity and Factors. In: Ashley, K. D., Van Engers, T. (eds.): ICAIL 2011: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Law, ACM, New York, pp. 101-105 (2011)
6. Araszkiewicz, M.: Towards Systematic Research on Statutory Interpretation in AI and Law. In: Ashley, K. (ed.): JURIX 2013: The Twenty-Sixth Annual Conference, IOS Press, Amsterdam, pp. 15-24 (2013)
7. Ashley, K. D.: Modeling Legal Argument: Reasoning with Cases and Hypotheticals. MIT Press, Cambridge, MA (1990)
8. Asooja, K., Bordea, G., Vulcu, G., O'Brien, L., Espinoza, A., Abi-Lahoud, E., Buitelaar, P., Butler, T.: Semantic Annotation of Finance Regulatory Text using Multilabel Classification. In: LeDASWAn workshop 2015, co-located with ESWC 2015, Portorož, Slovenia (2015)
9. Bench-Capon, T., Sartor, G.: A Model of Legal Reasoning with Cases Incorporating Theories and Values. In: *Artificial Intelligence* 150, pp. 97-142 (2003)
10. Biagioli, C.: Modelli Funzionali delle Leggi: Verso Testi Legislativi Autoesplicativi. European Press Academic Publishing (2009)
11. Boella, G., Janssen, M., Hulstijn, J., Humphreys, L., van der Torre, L.: Managing Legal Interpretation in Regulatory Compliance. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law, ACM, pp. 23-32 (2013)

12. Boer, A., Hoekstra, R., de Maat, E., Hupkes, E., Vitali, F., Palmirani, M., and Rátai, B.: CEN Metalex Workshop Agreement. http://www.metalex.eu/WA/proposal (2009)
13. Boer, A., Winkels, R., Vitali, F.: MetaLex XML and the Legal Knowledge Interchange Format. In: Computable Models of the Law. Lecture Notes in Artificial Intelligence 4884, Springer Verlag, Berlin, pp. 21-41 (2008)
14. Ceci, M., Gangemi, A.: An OWL Ontology Library Representing Judicial Interpretations. In: Semantic Web, IOS Press, vol. 7, no. 3, pp. 229-253 (2016)
15. Cook, W. W.: Hohfeld's Contribution to the Science of Law. In Yale Law Journal 721 (1918)
16. Gordon, T. F., and Walton, D.: The Carneades argumentation framework: Using presumptions and exceptions to model critical questions. In: 6th computational models of natural argument workshop (CMNA), ECAI, Italy, pp. 5-13 (2006)
17. Gordon, T. F., Governatori, G., Rotolo, A.: Rules and Norms: Requirements for Rule Interchange Languages in the Legal Domain. In: Governatori, G., Hall, J., Paschke, A. (eds.): RuleML 2009, LNCS, vol. 5858, Springer, Heidelberg, pp. 282-296 (2009)
18. Grabmair, M., Ashley, K.: Facilitating Case Comparison Using Value Judgments and Intermediate Legal Concepts. In: Ashley, K. D., Van Engers, T. (eds.): ICAIL 20111: proceedings of the 13th Int. Conference on A. I. and Law, ACM, New York, pp. 161-170 (2011)
19. Hart, H. L. A.: The Concept of Law. Clarendon Law Series, Oxford (1961)
20. Kelsen, H.: General Theory of Norms. Clarendon, Oxford (1991)
21. O'Brien, L., Ceci, M., and Al-Khalil, F.: An Akoma Ntoso Compliant System for Capturing Regulatory Requirements. In: Proceedings of the 1st International Akoma Ntoso Conference (IANC 2015), George Mason Univ., Arlington, VA (pending publication) (2016)
22. Palmirani, M., Governatori, G., Rotolo, A., Tabet, S., Boley H., Paschke, A.: LegalRuleML: XML-Based Rules and Norms. In: RuleML 2011, pp. 298-312 (2011)
23. Paschke, A., Kozlenkov, A., Boley, H.: A Homogeneous Reaction Rule Language for Complex Event Processing. In: Proceedings of the 2nd International Workshop on Event Drive Architecture and Event Processing Systems (2007)
24. Peczenik, A.: On Law and Reason. 2nd ed., Springer, Berlin (2007)
25. Prakken, H.: Combining Modes of Reasoning: An Application of Abstract Argumentation. In: Proceedings of the 11th European conference on Logics in Artificial Intelligence (vol. 5293), LNAI, Springer, Berlin, pp. 349-361 (2008)
26. Sartor, G.: Legal Reasoning: A Cognitive Approach to the Law. In: Pattaro, E., Rottleuthner, H., Shiner, R., Peczenik, A., Sartor, G. (eds.): A Treatise of Legal Philosophy and General Jurisprudence, volume 5, Springer (2005)
27. Searle, J. R.: Speech Acts: An Essay in the Philosophy of Language. Cambridge U. P. (1969)
28. Semantics of Business Vocabulary and Business Rules (SBVR) Version 1.2 (Apr 2013), http://www.omg.org/spec/SBVR/1.2/PDF
29. Sergot, M., Sadri, F.,Kowalski, R., Kriwaczek, F., Hammond, P., Cory, H. T.: The British Nationality Act as a Logic Program. In: Communications ACM 29, pp. 370-386 (1986)
30. van Haarst, R.: SBVR Made Easy – Business Vocabulary and Rules as a Critical Asset. Conceptual Heaven, Amsterdam (2013)
31. Vitali, F. and Zeni, F.: Towards a country-independent data format: The Akoma Ntoso experience. In Proceedings of the V legislative XML workshop. Florence, Italy: European Press Academic Publishing, pp. 67-86 (2007)
32. Wroblewski, J.: The Judicial Application of Law, Springer (1992)
33. Wyner A., Peters W.: On Rule Extraction from Regulations. In: Atkinson, K. (ed.): Legal Knowledge and Information Systems - JURIX 2011: The Twenty-Fourth Annual Conference, University of Vienna, Austria, 14th-16th December 2011. Frontiers in Artificial Intelligence and Applications 235, IOS Press (2011)