

# Creating Rules in FIRO-H

From SBVR to OWL 2: A step by step guide

**Abstract.** The purpose of this document is to provide a detailed guide, a manual, a descriptive reference on the modeling of legislative rules expressed in SBVR Structured English using FIRO-H. We start by describing FIRO-H, more specifically the concept of Actions which roughly correspond to SBVR Verb Concepts. We show how to model different verbs (in/transitive, binary/n-ary) and grammatical voices (active/passive), and complements (subject, object, indirect object, etc.). In a more advanced stage we discuss the modeling of some verbs that have specific meaning (has, is for, etc.) and then we talk about roles that do not translate into complements. Finally, we show how to create rules themselves: how to specify the modality, compose actions, etc.

## Table of Contents

1	Introduction .....	3
1.1	SBVR Basic Concepts .....	3
1.2	Different Levels of Interpretation .....	4
1.3	Definitions .....	4
1.4	FIRO-H Basic Actions .....	5
2	Verbs .....	7
2.1	Intransitive Verbs .....	7
2.2	Transitive Verbs .....	8
2.3	Additional Complements .....	8
2.3.1	Indirect Object .....	9
2.3.2	Location .....	9
2.3.3	Time*Needs*To*Be*Re-discussed .....	10
2.3.4	Measure .....	11
3	Verbs with Fixed Meaning .....	12
3.1	Specialization .....	12
3.2	Comparison .....	12
4	Factors that Do Not Translate into Complements .....	12
5	Rules .....	13
5.1	1 Obligation, 1 Compliance Made of 1 Action .....	13
5.2	1 Obligation, 1 Compliance, 1 Additional Condition .....	15
5.3	Enumerations .....	16

5.4	Multiple Conditions/Compliances/Breaches .....	17
5.5	Keywords Equivalence .....	17
5.6	Notes on SBVR, Mercury, and FIRO-H.....	17
6	References .....	18

# 1 Introduction

This document is meant to be a guide for, mainly, Semantic Technology Experts (STE) showing them how to transform a rule written in SBVR SE into a rule in FIRO-H.

## 1.1 SBVR Basic Concepts

At the very basic level, SBVR defines a *Vocabulary* and a *Rulebook*. A vocabulary contains:

1. **General Noun Concepts**. They are nouns that describe classes of objects (e.g. Bank, Share, ...)
2. **Individual Noun Concepts**. They designate individual occurrences of objects (e.g. Cork is a City, however Cork is an individual instance, a single, unique concept).
3. Verb Concepts. Composed of a *verb* (technically called verb symbol, but we will use verb and one or more **Verb Concept Roles**. A Verb Concept Role is a **Noun Concept**, either **General** or **Individual**. Examples of a basic verb concepts:
  - a. **Bank transfers Asset**. *transfers* is a verb that has 2 Verb Concept Roles, namely **Bank** and **Asset**.
  - b. **Bank of Ireland issues Share**. *issues* is a verb that has 2 Verb Concept Roles, namely the Individual Noun Concept **Bank of Ireland** and the General Noun Concept **Share**.
  - c. **Bank defaults**. *defaults* is a verb with 1 Verb Concept Role, namely the General Noun Concept **Bank**.

The vocabulary is used to build the rulebook. Every rule is a combination of:

1. A modality
2. One or multiple verb concepts connected with **keywords**.

Example of a rule:

**It is obligatory that each Price reflects the Prevailing Market Condition for each Share.**

1. The modality of the rule is expressed in “**It is obligatory that**” which indicates an obligation.
2. We have 3 Noun Concepts: **Price**, **Prevailing Market Condition**, and **Share**.
3. There are 2 Verb Concept:
  - a. **Price reflects Prevailing Market Condition**.
  - b. **Share has Prevailing Market Condition**

The vocabulary only contains Noun Concepts and Verb Concepts. The universal quantifier **each** is not present in the Verb Concept, but it is present in the rule. The modality is also in the rule only. This distinction is crucial for the modeling of rules in FIRO-H. Sections 1.2 through 4 are concerned with modeling the vocabulary in FIRO-H, while Section 0 shows how to model rules by composing Verb Concepts. Therefore, until we get to Section 0, we will be only concerned with modeling the vocabulary, more specifically Verb Concepts.

It is worth noting that the only voice that is allowed to be employed when describing a rule or a verb concept is the *active voice*. (REFERENCE):

Verb concept wordings shown as vocabulary entries use singular, active forms of verbs with the exception that present participles are sometimes used for characteristics. Infinitive, subjunctive, passive, and plural forms of verbs are implicitly usable in statements and definitions. For a binary

verb concept, the implicit passive form of a verb uses the past participle of the verb preceded by the word “*is*” and followed by the preposition “*by*.” For example, the implicit passive form of ‘expression *represents* meaning’ is ‘meaning *is represented by* expression’. The same pattern holds for verb concepts with more than two roles where a verb is used between the first two placeholders. For example, the implicit passive form of ‘thing *fills* role *in* actuality’ is ‘role *is filled by* thing *in* actuality’.

## 1.2 Different Levels of Interpretation

To understand the place of FIRO-H and its use, we should introduce the different levels of interpretations:

**Rule Interpretation.** This is the first and initial level of interpretation. This is the part where a Subject-Matter Expert (SME) would interpret the law, describe the rules in SBVR SE, then hand them to the STE who would translate them to OWL using FIRO-H as a general framework to describe regulative and constitutive rules. At this level, we can also describe internal business rules, which allows FIRO-H to classify these business rules as compliant or breaching legislative rules.

**Data Interpretation.** This is the second level of interpretation. This is the part where data “is fitted” to the legislative interpretation. A mapping is necessary between existing data and artefacts that live in the legislative interpretation level. Once the mapping is done, FIRO-H can help classifying data as compliant or breaching rules.

This document is meant as a guide for Rule Interpretation, unless we mention otherwise.

## 1.3 Definitions

**Action.** An action is an abstract category of events that is defined arbitrarily. It is the result of the interpretation on the behavior required by the norm/law/rule.

Actions are defined during *Rule Interpretation*. Notice that, according to this definition, the same norm/law/rule found in the legislative text can have multiple interpretations (“*defined arbitrarily*”), which is exactly what is desirable in our methodology: we need this level of freedom in the definition of actions.

An example of an action would be:

Bank transfers Asset

This action describes in an abstract manner all the events that describe a bank transferring an asset. It doesn’t designate a specific Bank or a particular Asset. It describes the class of actions that qualifies as Bank transfers Asset.

**Event.** An event is a concrete manifestation of an abstract action.

Events are actions described in data, not the norm/law/rule. They are defined at the level of *Data Interpretation*. The classification of an event under one or more particular actions is the result of the act of interpretation + mapping done at this level.

Examples of events relating to the Action Bank transfers Asset:

Bank of Ireland transfers Share N. 0001234

Central Bank of Ireland transfers Parcel N. 0004321

Notice that these events belong to the concrete world. They are events that happened in the real world and described as data.

**Modals.** Alethic (possibility, impossibility, necessity) or Deontic (obligation, permission, prohibition) used to describe constitutive and regulative rules respectively.

In this document we will only describe regulative rules, which means we will restrict ourselves to Deontic modalities.

**Condition.** A condition is an Action used in a rule. A condition has the same properties as actions and may restrict factors by:

1. Specifying its scope or value
2. Specifying the role it plays in another condition

Let's take the example Action Bank transfers Asset to demonstrate the previously mentioned types of specifications:

1. Bank transfers at least 2 Assets is a specification of scope.
2. Central Bank of Ireland transfers Asset is a specification of value
3. Bank transfers Asset of Enterprise. Here Asset plays a role also in a second action Enterprise has Asset.

A condition can be generic or specific. Generic conditions are conditions that determine if a given event is relevant to a given rule. Specific conditions are conditions that determine if a relevant event complies/breaches a rule.

**Rule.** A rule is made of 1 modal, 1 specific condition, and 0 or more additional conditions.

**Compliance.** A compliance is a specific condition. It is the main condition of an obligation rule. It has 1 Deontically Qualified Condition: the direct object/main target of the deontic modality. We will call it a Deontic Factor. If a relevant event meets the deontic factor, it is compliant with the rule, if it does not meet it, it breaches the rule.

For example, an obligation rule could be:

It is obligatory that a Bank transfers less than \$10'000

The modal part is expressed in "It is obligatory that". This rule has 1 Compliance condition: Bank transfers less than \$10'000. The Compliance condition is based on the action Bank transfers Asset. An event is relevant to the rule if it is describing a Bank transferring an amount of money. If the amount of money is less than \$10'000, then the event is compliant with the rule, otherwise it breaches it. Therefore, the factor of \$10'000 is the critical point that determines if an event is compliant or not; it is the Deontic Factor, the direct object of the obligation.

#### 1.4 FIRO-H Basic Actions

We find in the literature a myriad of papers (REFERENCES) talking about the transformation from SBVR to OWL 2, and they are all based on the same basic notion also described in the official documentation of SBVR(REFERENCE). Table 1 summarizes this translation:

Table 1 SBVR to OWL 2: The Classical Approach

SBVR	OWL 2
<u>General Noun Concept</u>	<i>Class</i>
<u>Individual Noun Concept</u>	<i>Individual</i>
<i>Verb</i>	<i>Object or Data Property</i>

For the purposes of FIRO-H, the conversion is not the same. Table 2 shows how SBVR concepts should be translated to OWL 2:

Table 2 SBVR to OWL 2: The FIRO-H Approach

SBVR	OWL 2
<u>General Noun Concept</u>	<i>Class subclassOf <b>Factor</b></i>
<u>Individual Noun Concept</u>	<i>Individual that is of type (Class subclassOf <b>Factor</b>)</i>
<i>Verb</i>	<i>Individual that is of type <b>Verb</b></i>

FIRO-H describes 3 main classes, as shown in Figure 1:

1. **Factor**. Which corresponds to a General Noun Concept in SBVR.
2. **Verb**. Which corresponds to the class of Verbs in SBVR. A verb found in a Verb Concept is an individual of the class **Verb** and not an **Object** or **Data Property** as described in the literature.
3. **Action**. Which corresponds to a verb concept in SBVR. It is composed of one or multiple Factors.

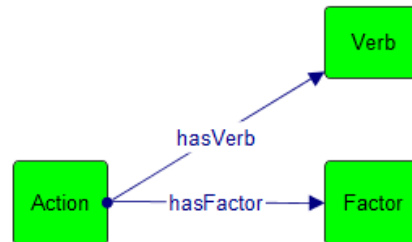


Figure 1 Basic Actions in FIRO-H

We chose a terminology that is different from that of SBVR because, although fairly simple, SBVR's terminology does not capture the legal view on rules: a rule is basically a composition conditions, and every condition is a composition of actions, not a composition of verbs or verb concepts.

Now a keen and alert reader would have some questions raised on how to actually build an action: Noun Concepts can be either classes or individuals, and verbs are always individuals! And here lies the major difference between what have been described in the literature and the FIRO-H: Actions are anonymous classes: they are described as axioms which allows us to unlock two powerful use cases:

1. Using FIRO-H to classify actions: which (composition of) actions are subsumed by others, and subsequently which (business) rules are compliant with other (legal) rules.
2. Using FIRO-H to classify real-world data: which events described in the real world correspond to which (composition of) actions, and subsequently which events breach which rules.

These use cases will be also clearer later in this document, but for the time being, the reader should be concerned by how to build basic actions. Therefore, the following sections will provide detailed examples on how to build actions from verb concepts.

## 2 Verbs

As noted earlier, we shall only consider verb concepts written in an active voice; this is a requirement of SBVR Structured English that we shall keep. In grammar, every verb has 1 **Subject**. If a verb has **0 Objects**, then the verb is *Intransitive*; if it has **1 or multiple Objects**, it is *Transitive*.

Subjects and Objects of verbs are Factors; the object properties **hasSubject** and **hasObject** are sub-properties of **hasFactor** and used to designate verb subjects and objects respectively.

### 2.1 Intransitive Verbs

General Form:

Subject *verb*

Example:

Bank *defaults*

Translation to FIRO-H:

1. Create a class **Bank** *subclassOf* **Factor**
2. Create an individual **defaults** instance of the class **Verb**
3. Create the class **Bank\_Defaults** as follows:

**Class** Bank\_Defaults:  
*Action*  
**and** (hasSubject **some** Bank)  
**and** (hasVerb **value** defaults)

Please note that there are no strict rules regarding naming an Action class, however, it is strongly recommended to create a meaningful class name for future maintenance: these action will be composed in rules and the definition of a rule should give its reader a good idea about the actions it is made of without having to delve deep into every action to comprehend the whole rule.

## 2.2 Transitive Verbs

General Form:

Subject *verb* Object

Example:

Bank *transfers* Asset

Translation to FIRO-H:

1. Create **Bank** and **Asset** classes; each is *subclassOf* **Factor**.
2. Create individual **transfers** instance of class **Verb**.
3. Create the class **Bank\_Transfers\_Asset** as follows:

**Class** Bank\_Transfers\_Asset:  
*Action*  
**and** (hasSubject **some** Bank)  
**and** (hasObject **some** Asset)  
**and** (hasVerb **value** transfers)

## 2.3 Additional Complements

Normally, verb concepts have more factors than just subjects and objects. Whether transitive or intransitive, a verb might be complemented with words to give it more meaning. In FIRO-H, we have the ability to define different kinds of complements. Table 3 shows the list of available complements:

Table 3 List of Available Complements in FIRO-H

Complement		Description	Object Property
Subject		Subject of the action	hasSubject
Object		Direct object of the action	hasObject
Indirect Object		Indirect object of the action	hasIndirectObject
Location <sup>1</sup>	Where	Where the action takes place	hasLocation
	From	Physical origin of the action	hasLocationFrom
	To	Physical destination of the action	hasLocationTo
	Through	Physical stage of the action	hasLocationThrough
Time <sup>2</sup>	Point in Time	Specific time	hasInstant
	Duration	An interval (possible semi-open)	hasDuration
Measure		Monetary value	hasAmount, hasCurrency

In the following, we will give the general form and examples for each and every complement.

<sup>1</sup> hasLocationFrom, hasLocationTo, and hasLocationThrough are sub-object properties of hasLocation

<sup>2</sup> hasInstant and hasDuration are sub-object properties of hasTime



### 2.3.1 Indirect Object

General Form:

Subject *verb* Object [*to*|*for*] Indirect Object

Example:

Bank *transfers* Asset *to* Firm

Translation to FIRO-H:

1. Create **Bank**, **Asset**, and **Firm** classes; each is *subclassOf* **Factor**.
2. Create individual **transfers** instance of class **Verb**.
3. Create the class **Bank\_Transfers\_Asset\_To\_Firm** as follows:

**Class** Bank\_Transfers\_Asset\_To\_Firm:  
*Action*  
**and** (hasSubject **some** Bank)  
**and** (hasObject **some** Asset)  
**and** (hasIndirectObject **some** Firm)  
**and** (hasVerb **value** transfers)

### 2.3.2 Location

General Form:

Subject *verb* Object [*in*|*from*|*to*|*through*]<sup>3</sup> Location

Example:

Bank *transfers* Asset *from* Ireland *to* France

Translation to FIRO-H:

1. Create **Bank** and **Asset** classes; each is *subclassOf* **Factor**.
2. Create individuals **Ireland** and **France**<sup>4</sup>.
3. Create individual **transfers** instance of class **Verb**.
4. Create the class **Bank\_Transfers\_Asset\_From\_Ireland\_To\_France** as follows:

**Class** Bank\_Transfers\_Asset\_From\_Ireland\_To\_France:  
*Action*  
**and** (hasSubject **some** Bank)  
**and** (hasObject **some** Asset)  
**and** (hasFromLocation **value** Ireland)  
**and** (hasToLocation **value** France)  
**and** (hasVerb **value** transfers)

---

<sup>3</sup> Other keywords?

<sup>4</sup> Imperatively instances of Factor itself. They could be part of a taxonomy.

### 2.3.3 Time\*Needs\*To\*Be\*Re-discussed

General Form:

Subject *verb* Object [*from*|*to*|*before*|*after*|*between*]<sup>5</sup> Time

Example:

Bank *transfers* Asset *after* the 4<sup>th</sup> of January 2015

Translation to FIRO-H:

1. Create **Bank** and **Asset** classes; each is *subclassOf* **Factor**.
2. Create individual **transfers** instance of class **Verb**.
3. Create the class **Bank\_Transfers\_Asset\_After\_04012015** as follows:

**Class** Bank\_Transfers\_Asset\_After\_04012015:

*Action*

**and** (hasSubject **some** Bank)

**and** (hasObject **some** Asset)

**and** (hasFromLocation **value** Ireland)

**and** (hasToLocation **value** France)

**and** (hasVerb **value** transfers)

---

<sup>5</sup> Other keywords?

### 2.3.4 Measure

There are different kinds of measures, but all measures have the same characteristic: they have a **Magnitude** and a **Kind**. This is the method used in the Nasa's and TopQuadrant's QUDT [REFERENCE] ontology. Examples:

1. 1000 USD is a measure of price: it has a *Magnitude* **1000** and a *Kind* **USD**
2. 24.5 meter is a measure of distance: it has a *Magnitude* **24.5** and a *Kind* **meter**.

General Form:

Subject *verb* Object [*has* | *for*]<sup>6</sup> Measure

Example:

Bank *sells* Asset *for* \$1

Translation to FIRO-H:

1. Create **Bank** and **Asset** classes; each is *subclassOf* **Factor**.
2. Create individual **sells** instance of class **Verb**.
3. Create class USD\_1 as follows

```
Class USD_1:  
Value  
and (hasMagnitude value 1)  
and (hasKind value USD)
```

4. Create the class **Bank\_Sells\_Asset\_For\_1** as follows:

```
Class Bank_Sells_Asset_For_1:  
Action  
and (hasSubject some Bank)  
and (hasObject some Asset)  
and (hasVerb value sells)  
and (hasValue value USD_1)
```

Note that we decided to model the value \$1 as General Noun Concept, thus a class USD\_1 in OWL 2, and not an Individual Noun Concept. The reason is that the rule, although specific on the amount, applies to any Asset that has a value equal to 1 USD even if we have it expressed in Euros, Lebanese Pounds, or Polish Zloty ...

---

<sup>6</sup> Other keywords?

### 3 Verbs with Fixed Meaning

#### 3.1 Specialization

#### 3.2 Comparison

FIRO-H provides a Compare class that can be used to describe comparisons such as less than or equal, equal to, etc. In order to compare two figures, one should create an action that *hasVerb* one or multiple individuals of Compare. Table 4 shows for every comparison verb, the equivalent individual in the Compare class, and what are the individuals to be used in the translation to FIRO-H:

Table 4 Comparison in FIRO-H: Verbs, their Equivalent in the class Compare, and How to Use Them in an Action

Verb	Individual in "Compare"	Individuals of an Action
<i>less than</i>	lt	lt
<i>greater to</i>	gt	gt
<i>equal to</i>	eq	eq
<i>less than or equal to</i>		lt,eq
<i>greater than or equal to</i>		gt,eq

General Form:

Subject *is* [*less than* | *less than or equal to* | *equal to* | *greater than or equal to* | *greater then*] Object

Example:

Quote Size *is less than or equal to* Market Size

Translation to FIRO-H:

1. Create **Quote Size** and **Market Size** classes; each is *subclassOf* **Factor**.
2. Create the class **QuoteSize\_LTE\_MarketSize** as follows:

```
Class QuoteSize_LTE_MarketSize:  
  Action  
  and (hasSubject some QuoteSize)  
  and (hasObject some MarketSize)  
  and (hasVerb some Compare and ({eq,lt}))
```

### 4 Factors that Do Not Translate into Complements

## 5 Rules

The first task that should be completed before modeling rules is the translation of the SBVR Vocabulary, which was covered in this document in Sections 0, 3, and 4. This part produces Actions, the building blocks of rules. Actions are abstract descriptions of real Events.

Rules are two kinds: Regulative and Constitutive. In this document we will restrict ourselves to regulative statements.

A Regulatory statement has 2 distinct components:

1. **Modality.** An obligation, prohibition, or a permission. FIRO-H provides a class *RegulativeModality* containing 3 individuals: *obligation*, *prohibition*, and *permission*.
2. **Conditions.** A configuration of 1 or multiple actions.

Conditions can be generic (simply called Conditions) or specific: Compliance or Breach. Every rule has exactly 1 specific condition and 0 or more generic Conditions.

- If an event meets the Compliance, then the event Complies to the rule. If an event meets the Breach, then the event breaches the rule.
- If an event does not meet a Condition that is not a Compliance or a Breach, then the event does not comply nor breach the rule: it is not relevant, it is not subject to the rule, ...

Conditions and Compliances/Breaches will be clearer in the upcoming examples. Figure 2 shows Regulatory Statements, RegulatoryModality, Conditions, Compliances, and Breaches.

General Forms:

Obligation + Compliance + [Condition]<sup>0...\*</sup>

Prohibition + Breach + [Condition]<sup>0...\*</sup>

### 5.1 1 Obligation, 1 Compliance Made of 1 Action

Vocabulary:

Noun Concepts	Verb Concepts
<u>Quote</u> <u>Bid Price</u>	<u>Quote</u> <i>include</i> <u>Bid Price</u>

Rule:

It is obligatory that each quote *include* at least two bid price

The first identifiable part is the Modality: an obligation expressed in 'It is obligatory that'. First of all, we should create **rule1** and individual of the class **RegulatoryStatement**, and affect to it the *obligation* modality: *rule1 hasModality obligation*.

Afterwards, noun concepts and individual noun concepts should be created: **Quote** and **BidPrice** are created as subclasses of **Factor**.

There is also 1 verb concept (Quote *include* Bid Price) that should be translated into an Action:

```

class Quote_Include_BidPrice:
  Action
  and (hasSubject some Quote)
  and (hasObject some BidPrice)
  and (hasVerb value include)

```

Notice that there are other keywords present in the rule (**each** and **at least one**); these keywords are not part of an Action, but of a Condition. And since we only have 1 condition, and rule1 is an obligation, then the condition is a Compliance:

```

class Rule1_Compliance:
  Compliance
  equivalentClass (compliesTo rule1)
  equivalentClass (Quote_Include_BidPrice and hasObject min 2 BidPrice)

```

We will provide a detailed list of equivalence between certain keywords like **at least**, **at most** with their OWL equivalents later in the document.

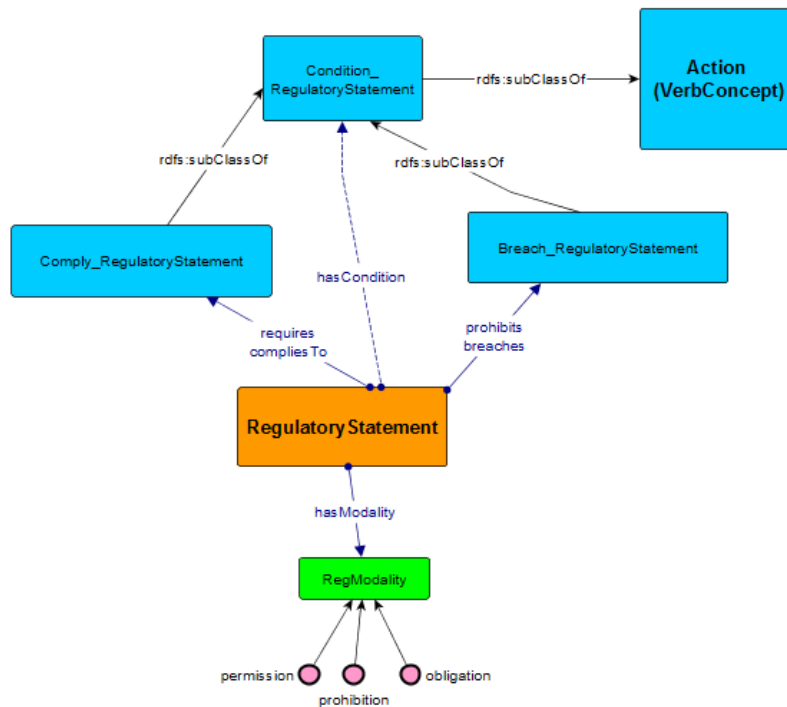


Figure 2 Condition, Compliance, Breach

## 5.2 1 Obligation, 1 Compliance, 1 Additional Condition

Vocabulary:

Noun Concepts	Verb Concepts
<u>Quote</u>	<u>Quote</u> <i>include</i> <u>Bid Price</u>
<u>Bid Price</u>	<u>Share</u> <i>traded on</i> <u>Trading Venue</u>
<u>Share</u>	<u>Share</u> <i>has</i> <u>Quote</u>
<u>Trading Venue</u>	

Rule:

It is obligatory that each quote (for each share traded on a trading venue) *include* at least two bid price

The first identifiable part is the Modality: an obligation expressed in 'It is obligatory that'. First of all, we should create **rule2** and individual of the class **RegulatoryStatement**, and affect to it the *obligation* modality: *rule2 hasModality obligation*.

Afterwards, noun concepts and individual noun concepts should be created: **Quote**, **BidPrice**, **Share**, and **TradingVenue** are created as subclasses of **Factor**.

There are 3 verb concepts that should be translated into an Action:

<b>class</b> Quote_Include_BidPrice: Action <b>and</b> (hasSubject <b>some</b> Quote) <b>and</b> (hasObject <b>some</b> BidPrice) and (hasVerb <b>value</b> include)	<b>class</b> Share_Traded_TradingVenue: Action <b>and</b> (hasSubject <b>some</b> Share) <b>and</b> (hasLocation <b>some</b> TradingVenue) <b>and</b> (hasVerb <b>value</b> trade)	<b>class</b> Share_Has_Quote Action <b>and</b> (hasSubject <b>some</b> Share) <b>and</b> (hasObject <b>some</b> Quote) <b>and</b> (hasVerb <b>value</b> has)
--	--	--

This rule has 2 conditions expressed in 2 separate actions:

1. A share should have a quote and be traded on a trading venue.
2. The quote of the aforementioned share should include at least two bid prices.

If there is an event that does not satisfy the first condition, it is not subject to the rule; the event does not constitute a breach of rule2. Therefore, the first condition is a general Condition.

However, the second condition is necessary for the rule: if an event satisfies the first condition, but does not satisfy the second one, it may constitute a breach. Otherwise, the event complies to the rule. Therefore, the second condition is a Compliance.

```
class Rule2_Codition1:  
Condition  
equivalentClass (  
  Share_Has_Quote  
  and hasSubject some  
    (Share and (subjectOf some Share_Traded_TradingVenue)))
```

```
class Rule2_Compliance:
    Compliance
    equivalentClass (compliesTo rule1)
    equivalentClass (Quote_Include_BidPrice
        and hasSubject some (Quote and (objectOf Rule2_Condition1)
        and hasObject min 2 BidPrice)
```

### 5.3 Enumerations

Let's take the example of Section 5.2:

It is obligatory that each quote (for each share traded on a trading venue) include at least two bid price

In real life, legislative rules tend to enumerate all possible conditions, so the rule might look like something like this:

It is obligatory that each quote (for each share and each depository receipt and each ETF and each certificate traded on a trading venue) include at least two bid price

Which means that Rule2\_Condition1 should be re-written:

```
class Rule2_Condition1:
    Condition
    equivalentClass (
        Share_Has_Quote
        and hasSubject some
            ((Share or DepositoryReciept or ETF or Certificate or FinancialInstrument)
            and (subjectOf some FinancialInstrument_Traded_TradingVenue)))
```

Where **FinancialInstrument\_Traded\_TradingVenue** is the action of the verb concept:

Financial Instrument *traded on* Trading Venue

And obviously **Share**, **DepositoryReciept**, **ETF** and **Certificate** are all subclasses of **FinancialInstrument**, itself a subclass of **Factor**.



## 5.4 Multiple Conditions/Compliances/Breaches

Let's take the example of Section 5.1:

It is obligatory that each quote include at least two bid price

Sometimes, a rule has a Compliance made of multiple Actions:

1. As a conjunction:

It is obligatory that each quote include and *make public* at least two bid price

2. As a disjunction:

It is obligatory that each quote include or *make public* at least two bid price

In the case of a conjunction, the rule should be split: it should be modeled in FIRO-H (and in SBVR/Mercury if possible) as distinct rules; one Action per Compliance/breach. The reason behind this approach is simple: in the case of an event that does not comply with a rule composed of a conjunction of Actions, we need to know which action was not fulfilled.

However, in the case of a disjunction of Actions, they should all be modeled as a single Compliance/Breach. The worry we expressed for conjunction of Actions is not present: if an event fulfills any of the actions, then it is compliant with the rules. It presents a breach in the sole case where all of the Actions in the disjunction are not fulfilled.

## 5.5 Keywords Equivalence

## 5.6 Notes on SBVR, Mercury, and FIRO-H

In this document we have presented a guide on the translation from SBVR to FIRO-H. Sometimes this translation is not straightforward: for example, in the Vocabulary, sometimes it is not clear if we should use a complement of location or not. Another important point of confusion comes from the rulebook, how can the STE define what is a general condition and what is the main Compliance/Breach?

If we're to use SBVR solely, then the STE confusion is inescapable, and his task will be far more complicated than it should be. This is why we developed Mercury, an extension of SBVR that builds artefacts on top of it to help the both the STE and SME to properly define the semantics of the vocabulary and the rules they are producing: hints like subject, object, condition, compliance, etc. will be available from the SME during authoring. These hints will be persisted and transferred to the STE to ensure a correct translation (and unambiguous) from SBVR to FIRO-H.

## 6 References