FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Logical Formalization of Semantic Business Vocabulary and Rules

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Master of Science (Computational Logic) (MSc)

im Rahmen des Studiums

Computational Logic (Erasmus-Mundus)

eingereicht von

Dmitry Solomakhin
Matrikelnummer 0818286

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: O. Univ. Prof. Dr. Thomas Eiter
Mitwirkung: Prof. Dr. Enrico Franconi
Univ.-Ass. Dr. Mantas Simkus

Wien,

(Unterschrift Verfasser) (Unterschrift Betreuung)

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Logical Formalization of Semantic Business Vocabulary and Rules

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science (Computational Logic) (MSc)

in

Computational Logic (Erasmus-Mundus)

by

Dmitry Solomakhin
Registration Number 0818286

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: O. Univ. Prof. Dr. Thomas Eiter
Assistance: Prof. Dr. Enrico Franconi
Univ.-Ass. Dr. Mantas Simkus

Vienna,

(Signature of Author) (Signature of Advisor)

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

Erklärung zur Verfassung der Arbeit

Dmitry Solomakhin
viale Druso 299b, 39100 Bolzano, Italien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)	(Unterschrift Verfasser)

Abstract

Artifacts such as laws, regulations and policies, use deontic constructs expressed in natural language to articulate human behavior, as well as alethic rules, describing what the organization takes things to be and how do the members of the community agree on the understanding of the domain. In most cases such artifacts have a rather ambiguous semantics, which can hardly be formalized. But sometimes, in order to satisfy the organizations' need of automatic reasoners, a logical framework can be built to provide such tools for validation and verification of the artifacts. The most promising candidate for such a framework is a recently approved standard for specifying Semantics of Business Vocabulary and Business Rules (SBVR). Despite existence of formally grounded notations, e.g. ORM2 specification language, up to now SBVR still lacks a sound and consistent logical formalization which would allow developing automated solutions able to automatically test business models for consistency of business rules.

This work reports on the attempt to provide logical foundations for the SBVR standard by the means of defining a specific first-order deontic-alethic logic (FODAL) designed to capture the desired semantics of business rules and their interaction. We also report on the logical properties of the aforementioned formalization, such as completeness and soundness. Moreover, we studied its connections with other logical formalisms, including satisfiability reduction to the first-order logic and the description logic $\mathcal{ALCQI}$, which allows to obtain decidability results for a certain fragment of FODAL logic. As a result, a special tool providing automated support for consistency checks of a set of $\mathcal{ALCQI}$-expressible deontic and alethic business rules was implemented. This tool was developed in the context of the ONTORULE FP7 project and became a part of the public deliverable. Addressing the objective of the project to combine rules and ontologies, we also implemented a translation of the aforementioned class of business rules into an OWL2 ontology, which facilitates integration with other modeling tools and fulfilment of the demand for automated solutions with reasoning support.

Kurzfassung

Im Kontext der Geschäftsprozessmodellierung verwenden Verordnungen und Richtlinien deontische und in natürlicher Sprache verfasste Formulierungen, um menschliches Verhalten zu artikulieren. Andererseits werden alethischen Regeln verwendet, um die Struktur eines Business-Bereiches zu beschreiben. Da Geschäftsmodelle stetig komplizierter werden, ist der Bedarf für ein logisches Rahmenwerk gegeben, welches automatische Schlußfolgerung mit Geschäftsregeln ermöglicht. Der vielversprechendster Kandidat für solch einen Rahmenwerk ist der Standard für Semantics of Business Vocabulary and Business Rules (SBVR).
Obwohl es eine formale Syntax für SBVR gibt, zum Beispiel die ORM2 Spezifikationssprache, fehlt dem SBVR Standard zur Zeit noch immer eine solide und konsistente logische Formalisierung, die eine automatisierte Konsistenzprüfung von Geschäftsregeln im Geschäftsmodellen ermöglichen würde.

Diese Arbeit berichtet über den Aufbau der logischen Grundlagen für den SBVR Standard, nämlich die deontisch-alethische Prädikatenlogik FODAL. Diese wurde entwickelt, um die erwünschte Semantik und Interaktion von Geschäftsregeln zu erfassen. Wir begutachten die logischen Eigenschaften der FODAL Formalisierung wie Vollständigkeit und Korrektheit. Darüber hinaus untersuchten wir Verbindungen von FODAL mit anderen logischen Formalismen einschließlich Prädikatenlogik und der $\mathcal{ALCQI}$ Beschreibungslogik. Wir zeigten auch die Entscheidbarkeit eines bestimmten FODAL-Fragments und entwickelten daraufhin ein Programm zur Konsistenzprüfung von deontischen und alethischen Geschäftsregeln, die in $\mathcal{ALCQI}$ ausdruckbar sind. Wir haben weiters eine Übersetzung der oben genannten Klasse von Geschäftsregeln in eine OWL2 Ontologie umgesetzt. Dieses Programm wurde im Rahmen des FP7 ONTORULE Projekt entwickelt und wurde in einem öffentlichen ONTORULE Bericht publiziert.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Automated support to enterprise modeling has increasingly become a subject of interest for organizations seeking solutions for storage, distribution and analysis of knowledge about business processes [11]. One of the most common approaches for describing business and the information used by that business is the rule-based approach [17]. It consists in identifying and articulating the rules that define the structure and control the operation of an enterprise [37]. The main expectation from automated solutions implementing this approach is the ability to automatically determine consistency of business rules in a business model.

Rule bases for describing complex processes may explode in size and become intractable by humans, thus resulting in modeling mistakes, or in the choice to useless formal approaches. For this reason, the Object Management Group (OMG) has recently approved a standard for specifying business objects and rules in natural language. The *Semantics of Business Vocabulary and Business Rules* (SBVR) [38] standard provides means for describing the structure of the meaning of rules, so called "semantic formulation", expressed in one of the intuitive notations, including the natural language that business people use [5] and Object-Role Modeling (ORM2) diagrams [18].

SBVR is supposed to describe structural (alethic) aspects of processes, as well as the policies that should guide agents behavior in certain situations. As such, the language is provided by the capability to specify deontic prescriptions, which in turn implies the possibility to introduce "deviation" in the model – i.e., the possibility that individual behave differently from what is prescribed by rules. However, the use of deontic rule formulations raises the problem of formally defining their semantics for automatic reasoning.

Several attempts have been made so far in order to provide a logical formalization for structural and behavioral rules in SBVR and its notations. Some certain semantic and logical foundations were defined already in the specification of the standard [38]. The most significant related work includes the first-order translation of the SBVR Structured English notation [16] and several formalizations of the ORM2 notation, including translation to first-order predicate logic [21] and some description logics ([25], [31] and [23]). However, none of the existing approaches enables consistency checks for a combined set of possibly interacting alethic and deontic business rules.

Another approach widely used to capture complex business semantics is ontology-based approach [40], which seeks to model basic business logic and meta-knowledge about business domain using ontologies. While the rule-based approach mostly focuses on the operational procedures of a business model, the ontology-based approach serves the purpose of capturing the rationale of the underlying business logic as well as providing means for business models interoperability.

This thesis research was conducted in the context of the "ONTORULE: Ontologies meet Business Rules" FP7 project [3], which aims to combine rule-based and ontology-based [40] approaches to business modeling in order to enable system interoperation and collaboration over business processes.

1.2 Objectives

The purpose of this thesis is to investigate the problem of logical formalization of SBVR, which provides means for automated solutions with reasoning support and enables integration of business rules with ontologies. In order to meet this purpose we address the following objectives:

- Study the specification of the SBVR standard, in particular, its semantic and logical foundations and identify the weak points.

- Examine the possible methods of formalization of business rules, taking into account the deontic part and develop such a formalization.

- Study the logical properties of the defined formal semantics and its connections with other logical formalisms, including those that facilitate integration with ontologies.

- Investigate the problem of developing automated solutions with reasoning support for business rules modeling and business processes monitoring.

1.3 Contribution of the thesis

Addressing the issues described above, this thesis bridges the gap between informal specification of the standard and a logical formalism for business rules with a clear formal semantics which allows automated reasoning on a combined set of structural and behavioral business rules.

In particular, we obtained the following results:

- We defined a multimodal *first-order deontic-alethic logic (FODAL)* with sound and complete axiomatization that captures the desired semantics of and interaction between business rules.
- We also showed that satisfiability in FODAL logic may be reduced to a standard first-order satisfiability for a class of formulas restricted to atomic modal sentences that may be used to express the majority of real-world rules. In attempt to overcome undecidability of FODAL which follows from undecidability of first-order logic, we also showed that under further restrictions FODAL satisfiability problem may be reduced to that of satisfying some formula in the $\mathcal{ALCQI}$ description logic.
- Moreover, in order to establish a relationship with a standard logical formalism, we defined a truth-preserving translation from a fragment of bimodal FODAL into quantified monomodal logic $\boldsymbol{QK}$, that can be used to facilitate the transfer of decidability results from well-studied fragments of predicate modal logics to FODAL.
- Finally, a special tool was implemented which provides an automated support for consistency checks of a set of $\mathcal{ALCQI}$-expressible business rules along with its translation to OWL2 ontology [32].

1.4 Outline of the work

The content of the thesis is divided into 5 chapters. In the second chapter an overview of the SBVR standard and its logical foundations is given, as well as the analysis of discovered shortcomings of its formal semantics. Third chapter describes the proposed logical formalization in terms of first-order deontic-alethic logic (FODAL) along with its syntax, semantics and complete and sound axiomatization. It also contains the study of the relevant logical properties of FODAL. The fourth chapter gives an overview of the tool developed to provide automated support for consistency checks together with translation to OWL2.

CHAPTER 2

Modeling business rules in SBVR

2.1 General overview

The *Semantics of Business Vocabulary and Business Rules* (SBVR) is a standard recently approved by the Object Management Group (OMG) [38] that provides means for specifying business objects and rules in natural language that business people use [5].

SBVR is supposed to describe structural (alethic) aspects of business processes, as well as the policies that should guide agents behavior in certain situations. A core idea of business rules formally supported by SBVR is the following: "Rules build on facts, and facts build on concepts as expressed by terms. Terms express business concepts; facts make assertions about these concepts; rules constrain and support these facts" [38]. The notions of terms and facts of this "business rules mantra" correspond to SBVR *noun concepts* and *verb concepts* (or *fact types*) respectively [7].

In this section we give a short overview of building blocks of Business Vocabulary and Business Rules as defined by SBVR standard approved by the Object Management Group (OMG) [38] and provide some examples in order to illustrate the subject.

2.1.1 Noun and verb concepts

Definition 1. A *noun concept* is "concept that is the meaning of a noun or noun phrase", which can be one of the following: an *object type*, an *individual concept* or a *fact type role*.
An object type is a "noun concept that classifies things on the basis of their common properties".
An individual concept is "a concept that corresponds to only one object [thing]".
A fact type role is a "noun concept that corresponds to things based on their playing a part, assuming a function or being used in some situation".

Definition 2. A *verb concept* (or a *fact type*) represents the notion of relations and is defined as "a concept that is the meaning of a verb phrase".
A fact type can have one (*characteristic*), two (*binary*) or more fact type roles.

Example 2.1.1. A noun concept <u>`rental car`</u> is an object type corresponding to those cars that are rented. A noun concept <u>`EU-Rent Vienna Downtown Office 1`</u> is an individual concept whose one and the only instance is an individual city branch in Vienna.
An example of a fact type in SBVR Structured English notation [38, annex C] is the following:

<u>`rental car`</u> `is stored at` <u>`branch`</u>

where roles <u>`rental car`</u> and <u>`branch`</u> are two fact type roles of a given verb concept.
The instances of this particular fact type are all actualities of rental cars being stored at branches.

2.1.2 Business rules

The SBVR standard defines two types of business rules: *structural (definitional) rules* and *operative behavioral rules* (See Figure 2.1).

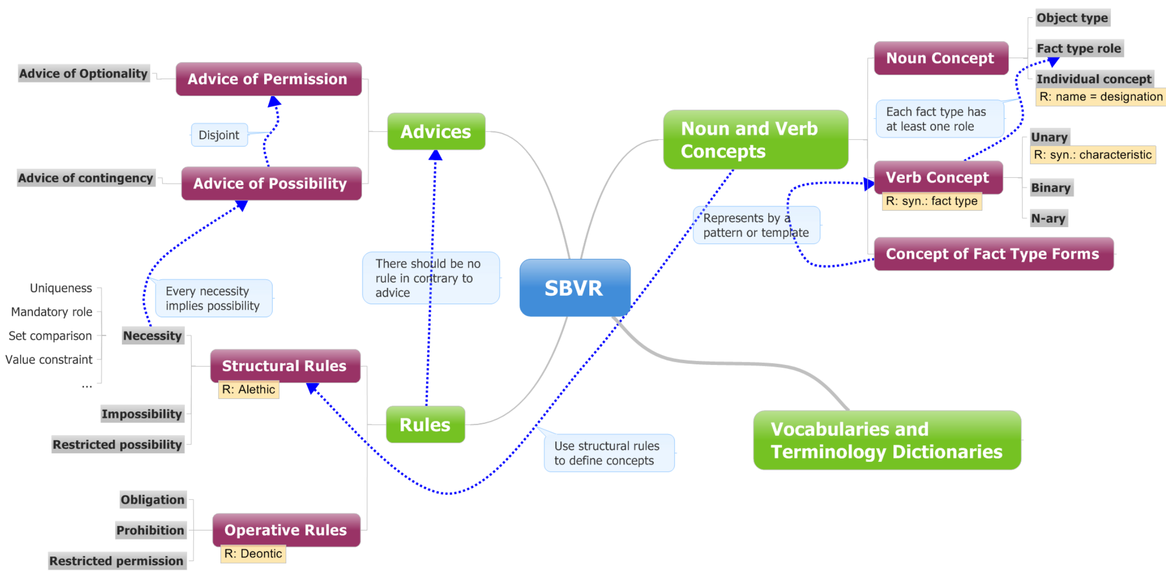


Figure 2.1: SBVR rules and advices

Definition 3. A *structural (alethic) rule* specifies what the organization takes things to be, how do the members of the community agree on the understanding of the domain [12]. It defines the characteristics of noun concepts and puts constraints on verb concepts, being in fact a constraint that is true by definition and hence can not be broken.

A structural business rule can be expressed using either *necessity*, *impossibility* or *restricted possibility* modalities.

Definition 4. An *operative (deontic) rule* is intended to describe business processes in organization (e.g. working instructions) and can be either ignored or violated by people.
An operative business rule can be expressed using either *obligation*, *prohibition* or *restricted permission* modalities.

N.B. It should be noted that one should not confuse the common notion of a "logical rule" understood as an inference rule and the notion of "rule" used hereafter in this thesis to denote a business rule.

2.2 Conceptual modeling in SBVR

In this section we recapitulate some definitions of the basic concepts from [21], introduce the basic background and terminology used in the standard [38] and provide some examples in order to illustrate the matter.

2.2.1 Semantical foundations

For any given business, the term "business domain" is usually used to indicate those aspects of the area that are of interest for the business modeling community. *Facts*, which are of a particular *Fact Type* are the main building blocks of SBVR and are considered to be propositions taken to be true by the business. The lowest-level logical unit in SBVR is an *Atomic Formulation*, which is a logical formulation based directly on a particular fact type, without any logical operations, quantifiers or operators.

Definition 5 (Conceptual model)**.**
An SBVR *conceptual model* $CM = \langle S, F \rangle$ is a structure intended to describe a business domain, where S is a *conceptual schema*, declaring fact types and rules relevant to the business domain, and F is a *population of facts* that conform to this schema.

Definition 6 (Ground facts)**.**
We use $f \in F$ to denote a *ground fact* from a fact population F in SBVR.
An *existential fact* f_{ex} is an assertion of the existence of an individual.
An *elementary fact* f_{el} is a declaration that an individual has some property, or that one or more individuals participate in a relationship.

A conceptual model, also referred to as *fact model*, may cover any time span and contain facts that describe past, present and future. Any change to a conceptual model, including changes regarding fact population and individual facts, produces a different fact model. Although in practice the conceptual schema as a part of a model can evolve over time, it is usually considered as fixed and it is the fact population which is considered to be highly variable. Different populations of the fact model with fixed conceptual schema are often referred to as *states of fact model* and include a set of ground facts, which can be either elementary or existential.

Business rules defined in the conceptual schema S can be considered as high-level facts (i.e., facts about propositions) and play a role of ***constraints***, which are used to impose restrictions concerning fact populations and may be either *static* (define which populations are possible or permitted) or *dynamic* (imposes restriction on transition between populations). When classifying the rule as a static constraint, it is asserted that it is true for each state of the fact model, taken individually, while dynamic constraints restrict transitions between states of the model. Hereafter in this work only static constraints will be taken into consideration.

2.2.2 Expressing business rules with modalities

The SBVR standard provides means for describing business facts and business rules that may be expressed either informally (in terms of un-interpreted comments) or formally (in terms of fact types of pre-defined schema and certain logical operators, quantifiers, etc.). For the main objectives of the standard its description focuses on formal statements of rules, since only those may be transformed into logical formulations, which can in turn be used for exchange with other rules-based software tools.

Such logical rule formulations are equivalent to formulae in 2-valued, first-order predicate calculus with identity [38]. In addition to standard universal ($\forall$) and existential ($\exists$) quantifiers, for the sake of convenience, SBVR standard allows logical formulation to use some pre-defined [18] numeric quantifiers, such as *at-most-one* ($\exists^{0..1}$), *exactly-n* ($\exists^{n}, n \geq 1$) and others.

In order to express the structural or operational nature of a business rule, the corresponding rule formulation use any of the basic alethic or deontic modalities shown in Table 2.1. Structural rule formulations use alethic operators: $\Box$ = *it is necessary that* and $\Diamond$ = *it is possible that*; while operative rule formulations use deontic modal operators $\boldsymbol{O}$ = *it is obligatory that*, $\boldsymbol{P}$ = *it is permitted that*, as well as $\boldsymbol{F}$ = *it is forbidden that*.

Definition 7. Rule formulation $\Box p$ having a necessity modal operator as a main modality is called a *necessity claim*. *Possibility, obligation* and *permission claims* are defined similarly as rule formulations having the main modal operator of possibility, obligation or permission, respectively.

Every rule formulation may be expressed using positive, negative or default verbalization, that can be obtained by applying the standard modal negation equivalences:

- *it is not necessary that* $\equiv$ *it is possible that not* $(\neg\Box p \equiv \Diamond\neg p)$;
- *it is not possible that* $\equiv$ *it is necessary that not* $(\neg\Diamond p \equiv \Box\neg p)$;
- *it is not permitted that* $\equiv$ *it is obligatory that not* $(\neg\boldsymbol{P}p \equiv \boldsymbol{O}\neg p)$;
- *it is not obligatory that* $\equiv$ *it is permitted that not* $(\neg\boldsymbol{O}p \equiv \boldsymbol{P}\neg p)$.

The reason for allowing both positive and negative verbalizations is that they are useful for validating the constraints by business domain experts, for example, when producing counter-examples. However, for the needs of automation tools, modal negation equivalences may be used together with double negation in order to end up with one alethic (e.g. $\Box$) and one deontic (e.g. $\boldsymbol{O}$) operator.

Modality		Modal Expression		applying modal negation equivalences = (Logically Equivalent) Modal Expression	
		Expression	Reading (Verbalized as):	Expression	Reading (Verbalized as):
alethic	necessity	$\Box p$	It is necessary that p	$\neg\Diamond\neg p$	It is not possible that not p
	non-necessity: the negation of necessity	$\neg\Box p$	It is not necessary that p	$\Diamond\neg p$	It is possible that not p
	possibility	$\Diamond p$	It is possible that p	$\neg\Box\neg p$	It is not necessary that not p
	impossibility: the negation of possibility	$\neg\Diamond p$	It is not possible that p It is impossible that p	$\Box\neg p$	It is necessary that not p
	contingency	$\Diamond p \wedge \neg\Box p$	It is possible but not necessary that p	$\neg(\neg\Diamond p \vee \Box p)$	It is neither impossible nor necessary that p
deontic	obligation	$\boldsymbol{O}p$	It is obligatory that p	$\neg\boldsymbol{P}\neg p$	It is not permitted that not p
	non-obligation: the negation of obligation	$\neg\boldsymbol{O}p$	It is not obligatory that p	$\boldsymbol{P}\neg p$	It is permitted that not p
	permission	$\boldsymbol{P}p$	It is permitted that p	$\neg\boldsymbol{O}\neg p$	It is not obligatory that not p
	prohibition: the negation of permission	$\neg\boldsymbol{P}p$ $\boldsymbol{F}p$	It is not permitted that p It is forbidden that p It is prohibited that p	$\boldsymbol{O}\neg p$	It is obligatory that not p
	optionality	$\boldsymbol{P}p \wedge \neg\boldsymbol{O}p$	It is permitted but not obligatory that p	$\neg(\neg\boldsymbol{P}p \vee \boldsymbol{O}p)$	It is neither prohibited nor obligatory that p

Table 2.1: Basic alethic and deontic modalities

The modality of an SBVR rule is determined by the main modal operator applied explicitly or implicitly in the corresponding rule formulation. If no modality is explicitly defined, then an alethic operator $\Box$ is often assumed. For example, rule formulation *"**Each** Method has **at***

most one *QualityAssurance"* implies necessity and can be thus explicitly verbalized as *"It is necessary that* ***each*** *Method has* ***at most one*** *QualityAssurance"*. The same constraint can be reformulated in a negative verbalization as *"It is impossible that* ***the same*** *Method has* ***more than one*** *QualityAssurance"*.

2.2.3 Notations for business vocabulary and rules

There are several common means of expressing facts and business rules in SBVR, namely through statements, diagrams or any combination of those, each serving best for different purposes. While graphical notations are helpful for demonstrating how concepts are related, they are usually impractical when defining vocabularies or expressing rules.

Definition 8. We use r to denote a business rule in SBVR regardless the particular format in which it is written. For the sake of readability we will denote any necessity claim as $r_{\Box}$, possibility claim as $r_{\Diamond}$, obligation claim as $r_{\boldsymbol{O}}$ and permission claim as $r_{\boldsymbol{P}}$.

The SBVR standard suggests three different verbalization approaches:

1. *SBVR Structured English* approach [38, Annex C], which defines one such way of using English that maps English words and structures to SBVR concepts. Business rules are introduced using a number of keyword prefixes for each modality, which are then applied to Structured English statement, representing the body of a rule. An example of a behavioral business rule of obligation expressed in SBVR Structured English notation is the following:

   r = **`It is obligatory that`** `the rented car of the rental is stored at the pick-up branch of the rental.`

   One of the advantages of using such controlled English is that its non-modal statements can be easily be translated into first-order logic [16]. We will hereafter denote by $\phi_{\hat{r_i}}$ a first-order representation of a non-modal Structured English sentence $\hat{r_i}$ from a rule r.

   N.B.: Since the nature of business rules implies the absence of uncertainty, it means that the resulting first-order formulae will not contain free variables, i.e. will be closed formulae. Then an SBVR rule may be represented by an expression resulted from application of modalities and boolean connectives to a set of closed first-order formulae $\phi_{\hat{r_i}}$.

2. *RuleSpeak* [34], which is an existing business rule notation that is widely used by business experts dealing with large-scale projects. RuleSpeak uses basically the same expression forms as Structured English, however instead of rule prefixes it uses embedded (mixfix)

keywords conveying a modality. The following example illustrates the syntax:

$r =$ `The` <u>`driver`</u> `of a` <u>`rental`</u> **`must`** `be` <u>`qualified`</u>.

3. *Object-Role Modeling (ORM2)*, which is a conceptual modeling approach combining both formal, textual specification language and formal graphical modeling language [19]. ORM2 specification language applies to mixfix predicates of any arity and contains predefined patterns covering a wide range of constraints typical for business domains. An example of a structural rule expressed as necessity statement in ORM2 specification language is the following:

   $r =$ `Each` <u>`visitor`</u> `has at most one` <u>`passport`</u>

   An example illustrating ORM2 graphical notation [20] is introduced on Figure 2.2.

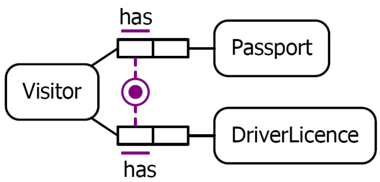


Figure 2.2: Example of ORM2 diagram

The difference between ORM2 and other notions is that it is a formal language per se, featuring rich expressive power, intelligibility, and semantic stability [22]. There exist several translations from non-modal ORM2 expressions to standard logics, including translation to first-order logic ([21]) and some description logics ([25], [31], [15]). Similar to $\phi_{\hat{r}}$, we will denote by $\phi_{\hat{r}}^{DL}$ a description logic representation of a non-modal ORM2 expression $\hat{r}$.

3a. *Cognition-enhanced Natural language Information Analysis Method (CogNIAM)* is one of the equivalent graphical variants of ORM2 notation, developed by one of the ONTORULE project partners in order to facilitate machine processing of ORM2 diagrams [38, annex L].

2.3 Model-theoretic semantics of SBVR

In this section we recapitulate the existing approach defined by SBVR standard [38, Section 10.1.1] which provides a model-theoretic semantics for business rule formulations. We then observe existing shortcomings, present some inaccuracy of the given semantics and ground the demands for a well-defined, noncontradictory formal semantics for both alethic and deontic business rules.

2.3.1 Alethic constraints

Structural rule formulations representing static alethic constraints may use the following alethic operators: $\Box$ = *it is necessary that*; and $\Diamond$ = *it is possible that*. Usually most statements of business rules include only one modal operator which is the main (preceding) operator of the whole statement. However, "to make life interesting" [38, p. 102] SBVR allows rule formulations to include multiple occurrences of modal operators as well as their nesting and embedding in the formulation.

Definition 9 (Normalized rule)**.**
A *normalized rule formulation* is a logical formulation having its modal operators in the front of the formula.

In order to provide a set of transformation rules to obtain the normalized form, the standard proposes to use modal negation equivalences, Barcan formulae (2.1) as well as other equivalences which hold in standard modal logics, e.g. in $\boldsymbol{S4}$ (2.2).

$$\forall x \Box Fx \equiv \Box \forall x Fx$$
$$\exists x \Diamond Fx \equiv \Diamond \exists x Fx \tag{2.1}$$
$$\Box\Box p \equiv \Box p; \quad \Diamond\Diamond p \equiv \Diamond p; \quad \Box\Diamond\Box\Diamond p \equiv \Box\Diamond p; \quad \Diamond\Box\Diamond\Box p \equiv \Diamond\Box p \tag{2.2}$$

In most cases the aforementioned transformations may lead to obtaining an equivalent constraint with one and the only modal operator which is placed at the front of the rule formulation. In this case, the SBVR model theory suggests omitting the preceding operator and tagging the corresponding rule as a necessity or a possibility respectively.
Then the implementation impact of a necessity tag is that any attempted change of the fact population that would cause the resulting fact model to violate the tagged constraint gets rejected. The semantics of a possibility tag **is not explicitly defined** by SBVR standard, however, assuming the intended semantics of the possibility operator $\Diamond$ in terms of possible worlds, one could consider the possibility tag as a guidance for potential changes of the fact populations.

In the rest of the cases, when there are multiple occurrences of modal operators or there is an embedded alethic modality that cannot be eliminated by transformation, SBVR considers adopting the formal semantics of the modal logic $\boldsymbol{S4}$ [6, p. 192]. All constraints are interpreted in terms of *possible world semantics* introduced by Saul Kripke [27], where a possible world corresponds to a state of the fact model that might exist at some point in time. Thus, a necessity static constraint is satisfied if and only if the proposition under the necessity modality is true in all possible states of the fact model. Similarly, a proposition is possible if and only if it is true in at least one possible state of the model. And a proposition is impossible if and only if it is false in all possible states.

2.3.2 Deontic constraints

Operative rule formulations represent static deontic constraints using standard deontic operators ($\boldsymbol{O}$ = *it is obligatory that*, $\boldsymbol{P}$ = *it is permitted that*) as well as $\boldsymbol{F}$ = *it is forbidden that*.
The approach to defining a formal semantics for deontic constraints is similar to the one for alethic rule formulations. Thus, a certain procedure is proposed to end up with a normalized rule formulation, suggesting usage of deontic modal negation equivalences, deontic counterparts to the Barcan formulae (2.3) as well as other equivalences such as (2.4).

$$\forall x \boldsymbol{O} Fx \equiv \boldsymbol{O} \forall x Fx$$

$$\exists x \boldsymbol{P} Fx \equiv \boldsymbol{P} \exists x Fx \tag{2.3}$$

$$p \rightarrow \boldsymbol{O} q \equiv \boldsymbol{O}(p \rightarrow q) \tag{2.4}$$

If the resulting normalized rule formulation contains one and the only preceding basic modal operator (i.e. $\boldsymbol{O}p$ or $\boldsymbol{P}p$), then this operator is omitted and the remaining first-order formula p is tagged as obligation or permission respectively (rather then necessity or possibility in the alethic case). The informal semantics behind this tag is that *it is ought (permitted) to be the case that* p, for all future states of the fact model until the rule is canceled or changed.
From the implementation perspective rules tagged as obligation may be violated in some states of the fact model and in which case some appropriate action ought to be taken to either eliminate the violation or to impose a penalty. However, such actions are considered to be outside of the scope of SBVR and remain unspecified. The idea behind tagging a rule as a permission is also undeclared, however based on the notion of obligation tag, it can be considered as guidance to be respected when imposing new obligation constraints.

Definition 10 (Model in SBVR)**.**
According to the model-theoretic perspective given in [38, p. 103], a model M in SBVR "is an interpretation where each *non-deontic* formula evaluates to true, and the model is classified as a *permitted model* if p in each deontic formula (of the form $\boldsymbol{O}p$) evaluates to true, otherwise the model is a *forbidden model* (though still a model)".

There can be the case when a formulation of a constraint may contain an embedded deontic modality that cannot be eliminated by the transformations (2.3-2.4). In this case SBVR doesn't commit to any particular deontic modal logic as with alethic modalities. Instead, it suggests to apply the *objectification approach* - each deontic modality get substituted by a respective service predicate with a reserved name which is defined at the business domain level, e.g. *"is forbidden"*, *"is permitted"*, etc. Additionally, in order to capture the desired semantics of the deontic modalities (e.g. modal negation equivalences), some complementary rules are declared, like: *"it is forbidden that"* ≡ *not "it is permitted that"*.

2.3.3 Ambiguity of the formal semantics of SBVR

While the logical foundations and definition of formal semantics given in the standard may actually serve for its purposes, the utility of such formalization for development of automated business modeling systems is arguable, since we can point out several shortcomings of such definition [36].

Independent treatment of modalities

We note that one of the main shortcomings of the SBVR's formal semantics for business rules is that both structural (alethic) and behavioral (deontic) rules are considered independently, without taking into account possible inconsistencies between each other. For example, some rule can be declared necessary with an alethic rule formulation and at the same time the negation of the logical formulation used in that rule may participate in some deontic rule, for instance, consider this trivial example:

It is necessary to breathe
It is permitted not to breathe

Although advices of permission are not meant to remove any degree of freedom, this set of rules is intuitively fallacious. Which is, however, not always the case when dealing with rules with contradictory bodies. Assume another trivial example:

It is obligatory not to smoke
It is possible to smoke

As a matter of fact, this set of rules is clearly consistent – indeed, although it is physically possible to smoke, it may be prohibited to smoke in public areas.

Moreover, the formal semantics based on tags, proposed by SBVR, is not able to capture the dependencies and contradictions even between rules of similar nature. However, the task of detecting possible contradictions and inconsistencies within model is very important from business modeling perspective. If there is no specific procedure to define whether the set of business rules is consistent, it is highly probable for the resulting models to be useless, as they will not correctly represent the intended business domain.

Incompatibility with formal semantics of classical deontic logics

We also note that another serious problem with the formalization defined by the standard is that its way of dealing with behavioral rules has serious discrepancies with formal semantics of existing deontic logics. The core source of such a conflict is the claim [38, p. 103] that in order to normalize a deontic rule formulation one could use equivalence (2.4):

$$p \rightarrow \boldsymbol{O}q \equiv \boldsymbol{O}(p \rightarrow q)$$

As a matter of fact, it can be observed that this transformation rule contradicts with the intended semantics of behavioral business rules as long as it doesn't preserve the validity of a given rule formulation with respect to the semantics of traditional deontic logics.

Let's assume that the proposed rule transformation is valid in some classical deontic logic $\boldsymbol{D}^*$. Then, according to the Kripke-style modal semantics of $\boldsymbol{D}^*$, the equivalence (2.4) should be valid in the class of serial frames [30] and therefore should be valid in every model based on a serial frame [6]. In particular this applies for the *Standard Deontic Logic* $\boldsymbol{SDL}$ [30].
Now let's assume the following model based on a serial frame represented on Figure 2.3.

$$\bullet \longrightarrow \bullet_p \longrightarrow \bullet_p \longrightarrow \bullet_{p,q} \longrightarrow \bullet_{p,q} \longrightarrow \ldots$$

Figure 2.3: Counter-example to the direct implication

It is obvious that the formula (2.4) is not satisfied at the first state of this model: while $p \rightarrow \boldsymbol{O}q$ evaluates to *true*, $\boldsymbol{O}(p \rightarrow q)$ evaluates to *false*.
Thus, the direct implication of the equivalence (2.4) is not valid in this model and therefore not valid in the class of serial frames.

Therefore it is showed that the transformation rule $p \rightarrow \boldsymbol{O}q \equiv \boldsymbol{O}(p \rightarrow q)$ does not preserve the validity of rule formulations and, as a result, the normalization procedure for deontic rule formulations defined by the standard's specification can not be applied.

Incompatibility with formal semantics of standard deontic logic SDL

$\boldsymbol{SDL}$ is one of the most prominent and widely used deontic logics whose formal semantics is adopted in order to formalize notions of obligation and permission. However, it can be shown that the model-theoretic semantics introduced in the SBVR specification becomes contradictory when adopting the formal semantics of $\boldsymbol{SDL}$.

Let's assume the following conceptual schema $\mathcal{S}$ and its interpretation $\mathcal{I}$ which satisfies p and falsifies q at some accessible world:

$$\boldsymbol{O}(p \to q) \tag{2.5}$$
$$\boldsymbol{O}(\neg q) \tag{2.6}$$
$$\boldsymbol{O}(p) \tag{2.7}$$
$$p \tag{2.8}$$
$$\neg q \tag{2.9}$$

Then, from a model-theoretic perspective of SBVR, the given interpretation is classified as forbidden model, because each non-deontic formula evaluates to true, while the first deontic rule is violated.

However, assuming formal semantics of $\boldsymbol{SDL}$, one could obtain the following results via natural deduction:

$$\boldsymbol{O}(p \to q) \to (\boldsymbol{O}p \to \boldsymbol{O}q) \qquad axiom\ of\ \boldsymbol{SDL} \tag{2.10}$$
$$\boldsymbol{O}p \to \boldsymbol{O}q \qquad modus\ ponens\ of\ 2.5\ and\ 2.10 \tag{2.11}$$
$$\boldsymbol{O}q \qquad modus\ ponens\ of\ 2.11\ and\ 2.7 \tag{2.12}$$
$$\boldsymbol{O}q \to \neg\boldsymbol{O}(\neg q) \qquad axiom\ of\ \boldsymbol{SDL} \tag{2.13}$$
$$\neg\boldsymbol{O}(\neg q) \qquad modus\ ponens\ of\ 2.12\ and\ 2.13 \tag{2.14}$$
$$\bot \qquad contradiction\ of\ 2.6\ and\ 2.14 \tag{2.15}$$

Therefore the conceptual schema (2.5)–(2.7) is inconsistent, hence it is counter-intuitive to claim that it has a model, that contradicts with the definition of a "forbidden model" given by the standard's specification.

Ambiguity of model-theoretic semantics of SBVR

As a matter of fact, since the model-theoretic semantics given by SBVR standard specification mentions only obligatory deontic formulae, it can be observed that a serious ambiguity or even fallacy of this semantics is caused when dealing with deontic formulae of the form $\boldsymbol{P}p$, that represent the notion of permission.

Assume that $\{\boldsymbol{O}p, \boldsymbol{P}(\neg p)\}$ is a rule set of some SBVR conceptual schema $\mathcal{S}$, and some fact population $\mathcal{I}$ of $\mathcal{S}$ is such that it satisfies p. Then, from a model-theoretic perspective of SBVR, the given fact population is classified as permitted model.

However, the conceptual schema $\mathcal{S}$ is inconsistent as it contains contradicting deontic rule formulations (since $\boldsymbol{O}p \equiv \neg\boldsymbol{P}(\neg p)$). Therefore it is counter-intuitive to claim that it has a model.

As a matter of fact, the introduced example not only proves the fallacy of model-theoretic semantics of SBVR operative rules, but also justifies the urgent need for well-defined, noncontradictory formal semantics for both alethic and deontic business rules in order to provide business expert with automated modeling solutions preventing inconsistent business models.

CHAPTER 3

Proposed logical formalization of SBVR

3.1 First-order deontic-alethic logic (FODAL)

The basic formalisms we use to model business rule formulations are standard deontic logic ($\boldsymbol{SDL}$) and normal modal logic $\boldsymbol{S4}$, which are both propositional modal logics. We then construct a *first-order deontic-alethic logic* (FODAL) – a multimodal logic, as a first-order extension of a combination of $\boldsymbol{SDL}$ and $\boldsymbol{S4}$ to be able to express business constraints defined in SBVR. In order to construct the first-order extension for the combined logic we follow the procedure described in [14].

3.1.1 Language

The alphabet of FODAL contains the following symbols:

- a set of *propositional connectives*: $\neg, \wedge$.
- a universal quantifier: $\forall$ (*for all*).
- an infinite set $\mathcal{P} = \{P_1^1, P_2^1, ..., P_1^2, P_2^2, ..., P_1^n, P_2^n, ...\}$ of n-place *relation symbols* (also referred to as *predicate symbols*).
- an infinite set $\mathcal{V} = \{v_1, v_2, ...\}$ of *variable* symbols.
- alethic modal operator: $\Box$ (*necessity*).
- deontic modal operator: $\boldsymbol{O}$ (*obligation*).

Definition 11 (Atomic formula)**.**
An *atomic formula* is any expression of the form $R(x_1, x_2, ..., x_n)$, where R is an n-place relation symbol and $x_i, i = \overrightarrow{1 \dots n}$, are variables.

Definition 12 (FODAL formulae)**.**
The *formulae* of FODAL are defined inductively in the following way:

- Every atomic formula is a formula.
- If X is a formula, so is $\neg X$.
- If X and Y are formulae, then $X \wedge Y$ is a formula.
- If X is a formula, then so are $\Box X$ and $\boldsymbol{O}X$.
- If X is a formula and v is a variable, then $\forall v X$ is a formula.

The notions of *free and bound variables* and their occurrences are defined in a usual way [14].
The existential quantifier ($\exists$) as well as other propositional connectives ($\vee, \rightarrow, \leftrightarrow$) are defined as usual, while additional modal operators ($\Diamond, \boldsymbol{P}, \boldsymbol{F}$) are defined in the following way:

$$\Diamond\phi \equiv \neg\Box\neg\phi \tag{3.1}$$

$$\boldsymbol{P}\phi \equiv \neg\boldsymbol{O}\neg\phi \tag{3.2}$$

$$\boldsymbol{F}\phi \equiv \boldsymbol{O}\neg\phi \tag{3.3}$$

Definition 13 (Closed formula)**.**
A FODAL formula with no free variable occurrences is called a *closed formula* or a *sentence*.

Definition 14 (Objective formula)**.**
An *objective formula* is a FODAL formula which does not contain any modalities.

Definition 15 (Modal sentence)**.**
A *modal sentence* is a sentence whose main logical operator is a modal operator.
An *atomic modal sentence* is a modal sentence which contains one and the only modal operator.

3.1.2 Semantics

Semantics for propositional multimodal logics are usually defined using Kripke *n-frames* [28].

Definition 16 (N-frame)**.**
An *n-frame* is a relational structure $\mathfrak{F} = \langle \mathcal{W}, R_1, ..., R_n \rangle$, where $n > 0$ is a natural number, $\mathcal{W}$ is a non-empty set and $R_i, i = \overrightarrow{1 \dots n}$, is a binary relation on $\mathcal{W}$.

Definition 17 (Path)**.**
A *path* from point $x \in \mathcal{W}$ to point $y \in \mathcal{W}$ in an n-frame $\mathfrak{F}$ is a sequence $\langle x_0, x_1, ..., x_k \rangle$ such that $x_i \in \mathcal{W}, x_0 = x, x_k = y$ and $x_i R_j x_{i+1}$ for each $i = \overrightarrow{0..k}$ and some $j \in [1; n]$.

Definition 18 (Rooted n-frame)**.**
An n-frame $\mathfrak{F}$ is called *rooted* if there exists some $x \in \mathcal{W}$, called *root*, such that for every $y \in \mathcal{W}, y \neq x$ there is a path from x to y.

Definition 19 (Depth of a frame)**.**
An n-frame $\mathfrak{F}$ is said to *be of depth k* if k is the length of the longest path in $\mathfrak{F}$. If no such path exists, then the n-frame is said to be of *infinite* length.

In propositional multimodal logics models are defined by a frame $\mathfrak{F} = \langle \mathcal{W}, R_1, ..., R_n \rangle$ and a relation $\vDash$ between propositional letters and worlds. In the first-order case, the relation $\vDash$ is substituted by a first-order interpretation function and an n-frame is complemented with a definition of domain, which can be either constant or world-dependent.

Since SBVR itself interprets constraints in terms of Kripke semantics where possible worlds correspond to states of the fact model, i.e. different fact populations, the choice of varying domain semantics is intuitively justified [38]. Also, since SBVR rule formulations may include only two types of modalities: deontic and alethic, - hereafter in this work only two-layer frames are considered as a particular case of n-frames with $n = 2$ and accessibility relations $R_{\boldsymbol{O}}$ and $R_{\Box}$ respectively.

Definition 20 (Augmented frame)**.**
A varying domain augmented bimodal frame is a relational structure $\mathfrak{F}_{var} = \langle \mathcal{W}, R_{\boldsymbol{O}}, R_{\Box}, \mathfrak{D} \rangle$, where $\langle \mathcal{W}, R_{\boldsymbol{O}}, R_{\Box} \rangle$ is a two-layer frame and $\mathfrak{D}$ is a domain function mapping worlds of $\mathcal{W}$ to non-empty sets. A *domain of a possible world* w is then denoted as $\mathfrak{D}(w)$ and a *frame domain* is defined as $\mathfrak{D}(\mathfrak{F}) = \bigcup \{\mathfrak{D}(w_i) | w_i \in \mathcal{W}\}$.

Definition 21 (Interpretation)**.**
An *interpretation* $\mathfrak{I}$ in a varying domain augmented frame $\mathfrak{F}_{var} = \langle \mathcal{W}, R_{\boldsymbol{O}}, R_{\Box}, \mathfrak{D} \rangle$ is a function which assigns to each m-place relation symbol P and to each possible world $w \in \mathcal{W}$ some m-place relation on the domain $\mathfrak{D}(w)$ of that world.
$\mathfrak{I}$ can be also interpreted as a function that assigns to each possible world $w \in \mathcal{W}$ some first-order interpretation $\mathfrak{I}(w)$.

Definition 22 (Model)**.**
A *varying domain first-order model* is a structure $\mathfrak{M} = \langle \mathcal{W}, R_{\boldsymbol{O}}, R_{\Box}, \mathfrak{D}, \mathfrak{I} \rangle$, where $\langle \mathcal{W}, R_{\boldsymbol{O}}, R_{\Box}, \mathfrak{D} \rangle$ is a varying domain augmented frame and $\mathfrak{I}$ is an interpretation in it.

The satisfiability relation between varying domain first-order models and formulae is then defined in the usual way, using the notion of valuation which maps variables to elements of the domain.

Definition 23 (Valuation)**.**
A *valuation* σ in a varying domain model $\mathfrak{M}$ is a mapping that assigns to each free variable x some member of the domain of the model $\mathfrak{D}(\mathfrak{M})$. A valuation σ' is an x*-variant* of a valuation σ when σ' and σ agree on all variables except x.

Definition 24 (Truth in a model)**.**
Let $\mathfrak{M} = \langle \mathcal{W}, R_{\boldsymbol{O}}, R_{\Box}, \mathfrak{D}, \mathfrak{I} \rangle$ be a varying domain first-order model, X, Y and Φ be FODAL formulae. Then for each possible world $w \in \mathcal{W}$ and each valuation σ on $\mathfrak{D}(\mathfrak{M})$ the following holds:

- if P is a m-place relation symbol, then $\mathfrak{M}, w \vDash_{\sigma} P(x_1, ..., x_m)$ if and only if $(\sigma(x_1), ..., \sigma(x_m)) \in \mathfrak{I}(P, w)$ or, equivalently, $\mathfrak{I}(w) \vDash_{\sigma}^{FOL} P(x_1, ..., x_m)$,
- $\mathfrak{M}, w \vDash_{\sigma} \neg X$ if and only if $\mathfrak{M}, w \nvDash_{\sigma} X$,
- $\mathfrak{M}, w \vDash_{\sigma} X \wedge Y$ if and only if $\mathfrak{M}, w \vDash_{\sigma} X$ and $\mathfrak{M}, w \vDash_{\sigma} Y$,
- $\mathfrak{M}, w \vDash_{\sigma} \forall x \Phi$ if and only if for every x-variant σ' of σ at w, $\mathfrak{M}, v \vDash_{\sigma} \Phi$,
- $\mathfrak{M}, w \vDash_{\sigma} \exists x \Phi$ if and only if for some x-variant σ' of σ at w, $\mathfrak{M}, v \vDash_{\sigma} \Phi$,
- $\mathfrak{M}, w \vDash_{\sigma} \Box X$ if and only if for every $v \in \mathcal{W}$ such that $wR_{\Box}v$, $\mathfrak{M}, v \vDash_{\sigma} X$,
- $\mathfrak{M}, w \vDash_{\sigma} \Diamond X$ if and only if for some $v \in \mathcal{W}$ such that $wR_{\Box}v$, $\mathfrak{M}, v \vDash_{\sigma} X$,
- $\mathfrak{M}, w \vDash_{\sigma} \boldsymbol{O}X$ if and only if for every $v \in \mathcal{W}$ such that $wR_{\boldsymbol{O}}v$, $\mathfrak{M}, v \vDash_{\sigma} X$,
- $\mathfrak{M}, w \vDash_{\sigma} \boldsymbol{P}X$ if and only if for some $v \in \mathcal{W}$ such that $wR_{\boldsymbol{O}}v$, $\mathfrak{M}, v \vDash_{\sigma} X$.

The notions of validity and truth are defined classically:

- If for all valuations σ we have that $\mathfrak{M}, w \vDash_{\sigma} \Phi$, then the FODAL formula Φ is said to be *true at the world* w of a model, which is abbreviated as $\mathfrak{M}, w \vDash \Phi$.
- A FODAL formula Φ is *valid in a model* $\mathfrak{M}$, denoted as $\mathfrak{M} \vDash \Phi$, if it is true at every world $w \in \mathcal{W}$ of $\mathfrak{M}$.
- A FODAL formula Φ is said to be *valid in the class of frames* $\mathfrak{F}$, denoted as $\vDash_{\mathfrak{F}} \Phi$, if it is valid in every model of the given class of frames $\mathfrak{F}$.

In order to correctly capture the behavior and interaction of the alethic and deontic modal operators it is necessary to constrain the corresponding accessibility relations: the alethic accessibility is usually taken to be a reflexive and transitive relation ($\boldsymbol{S4}$) [6], while the behavior of a deontic modality is classically considered to be captured by a serial relation ($\boldsymbol{KD}$) [29].

Moreover, since one of the objectives of formalization under development is to define the consistency of the set of business rules, it should also take into account the existing interaction between alethic and deontic modalities. The desired interaction can be verbalized as *"Everything which is necessary is also obligatory"* and then expressed as a following FODAL formula:

$$\Box X \rightarrow \boldsymbol{O}X \tag{3.4}$$

It can be shown that the modal formula 3.4 defines a certain class of bimodal frames for which the following restriction on the accessibility relations $R_\Box$ and $R_{\boldsymbol{O}}$ holds:

$$R_{\boldsymbol{O}} \subseteq R_\Box \tag{3.5}$$

Proposition 3.1.1. *The modal formula $\Box X \rightarrow \boldsymbol{O}X$ defines the class of augmented bimodal frames $\mathfrak{F} = \langle \mathcal{W}, R_{\boldsymbol{O}}, R_\Box, \mathfrak{D} \rangle$ such that $R_{\boldsymbol{O}} \subseteq R_\Box$, where $R_\Box$ is a preorder and $R_{\boldsymbol{O}}$ is serial. We then call such frame a* **FODAL** *frame.*

Proof. Recall that a modal formula ϕ defines a class of frames **FODAL** if and only if it is valid on precisely the frames in **FODAL**, i.e. (a) ϕ must be valid on every frame in **FODAL** and (b) it should the case that ϕ is falsified on any frame that is not in **FODAL** [33].

Suppose $\mathfrak{M}$ is a model based on the bimodal frame of the class under consideration, $w \in \mathcal{W}$ is an arbitrary possible world and σ is an arbitrary valuation. For let $\Box X$ be true at w: $\mathfrak{M}, w \vDash_\sigma \Box X$. Then, by definition of satisfiability:

$$\forall v \in \mathcal{W} \quad \text{such that} \quad wR_\Box v: \quad \mathfrak{M}, v \vDash_\sigma X \tag{3.6}$$

Now assume $\boldsymbol{O}X$ does not hold at w. This means that

$$\exists v' \in \mathcal{W} \quad \text{such that} \quad wR_{\boldsymbol{O}}v': \quad \mathfrak{M}, v' \nvDash_\sigma X \tag{3.7}$$

However $R_{\boldsymbol{O}} \subseteq R_\Box$, which means that $(w, v') \in R_\Box$ and thus, by assumption 3.6, $\mathfrak{M}, v' \vDash_\sigma X$, which contradicts with 3.7. Thus $\boldsymbol{O}X$ is also true at w. Hence, whenever $\Box X$ holds at world w, $\boldsymbol{O}X$ also holds, irrespective to the possible world w and valuation σ. Then $\Box X \rightarrow \boldsymbol{O}X$ is valid in a class of frames with property 3.5, which proves first part of the definability claim 3.1.1.

To complete the proof it is necessary to show that $(\Box X \rightarrow \boldsymbol{O}X)$ can be falsified on any bimodal frame that does not belong to the considered class. Suppose $\mathfrak{M}' = \langle \mathcal{W}, R_{\boldsymbol{O}}, R_\Box, \mathfrak{D}, \mathfrak{I} \rangle$ is a model which is not based on the bimodal frame from introduced class **FODAL**, so $R_{\boldsymbol{O}} \nsubseteq R_\Box$. This means that for some possible world $w \in \mathcal{W}$ there is another world $v \in \mathcal{W}$, such that

$(w,v) \in R_{\boldsymbol{O}}$ but $(w,v) \notin R_{\Box}$. For let σ be such a valuation that makes X false at v and true everywhere else, so also at w. Then $\mathfrak{M}', w \vDash_{\sigma} X$ and $\mathfrak{M}', w \vDash_{\sigma} \Box X$ by the reflexivity of $R_{\Box}$ as a $\boldsymbol{S4}$ accessibility relation. Hence $\Box X$ is true at w under valuation σ. However $\boldsymbol{O}X$ does not hold at w because there exists $v, (w,v) \in R_{\boldsymbol{O}}$ for which $\mathfrak{M}', v \nvDash_{\sigma} X$. Therefore the formula $(\Box X \to \boldsymbol{O}X)$ is falsified and the definability claim under consideration is proved. $\Box$

Definition 25 (**FODAL** model)**.**
A **FODAL** *model* is a varying domain first-order model $\mathfrak{M}_{\mathbf{FODAL}} = \langle \mathcal{W}, R_{\boldsymbol{O}}, R_{\Box}, \mathfrak{D}, \mathfrak{I} \rangle$ such that $R_{\boldsymbol{O}}$ is serial, $R_{\Box}$ is a preorder relation and $R_{\boldsymbol{O}} \subseteq R_{\Box}$.

Hereafter in this work only **FODAL** models are taken into consideration.

3.1.3 Axiomatization

As SBVR rules formulations may include only closed formulae, it might seem that taking only FODAL sentences into consideration is sufficient for the formalization being developed. However from the logical perspective proofs of those sentences would involve formulae with free variables. In the following $\Phi(x)$ denotes a FODAL formula in which the variable x may have free occurrences and the notion of substitutability if defined in a classical way:

Definition 26 (Substitutability)**.**
A free variable y is *substitutable* for a variable x in $\Phi(x)$ provided that no free occurrences of x in $\Phi(x)$ is within the scope of $\forall y$.

An $FODAL$ axiom system for first-order alethic-deontic logic is defined following the approach presented in [14] and is obtained by combining the axiom systems for the propositional modal logics $\boldsymbol{S4}$ and $\boldsymbol{KD}$ and extending the resulting combination with additional axiom schemas and the axiom 3.4 reflecting desired interaction between alethic and deontic modalities.

Definition 27 (Axioms)**.**
All the formulae of the following forms are taken as axioms.

(Tautologies $\boldsymbol{S4}$)	Any FOL substitution-instance of a theorem of $\boldsymbol{S4}$ logic	(3.8)
(Tautologies $\boldsymbol{KD}$)	Any FOL substitution-instance of a theorem of $\boldsymbol{KD}$ logic	(3.9)
(Vacuous $\forall$)	$\forall x\phi \equiv \phi$, provided x is not free in ϕ	(3.10)
($\forall$ Distributivity)	$\forall x(\phi \to \psi) \to (\forall x\phi \to \forall x\psi)$	(3.11)
($\forall$ Permutation)	$\forall x \forall y\phi \to \forall y \forall x\phi$	(3.12)
(Restricted $\forall$ Elimination)	$\forall y(\forall x\phi(x) \to \phi(y))$	(3.13)
(Necessary Obligation)	$\Box\phi \to \boldsymbol{O}\phi$	(3.14)

Definition 28 (Rules of inference)**.**

$$\textit{(Modus Ponens)} \quad \frac{\phi \quad \phi \rightarrow \psi}{\psi} \tag{3.15}$$

$$\textit{(Alethic Necessitation)} \quad \frac{\phi}{\Box\phi} \tag{3.16}$$

$$\textit{(Deontic Necessitation)} \quad \frac{\phi}{\boldsymbol{O}\phi} \tag{3.17}$$

$$\textit{(}\forall\textit{ Generalization)} \quad \frac{\phi}{\forall x\phi} \tag{3.18}$$

Theorem 3.1.2. *The $FODAL$ axiom system is complete and sound with respect to the class of **FODAL** frames.*

Proof. The resulting multimodal logic can be considered as a simple fusion of quantified modal logics $\boldsymbol{QS}4$ and $\boldsymbol{QKD}$ enriched with an additional inclusion axiom 3.14. Both basic logics $\boldsymbol{QS}4$ and $\boldsymbol{QKD}$ are known to be sound and complete with respect to the varying domain semantics [24]. Moreover, it was recently shown in [35] that soundness and completeness properties can be transferred from monomodal first-order logics $\boldsymbol{QS}4$ and $\boldsymbol{QKD}$ to their multimodal join. Therefore, in order to prove the soundness of the $FODAL$ axiomatization it is required to state the validity of the additional *Necessary Obligation* axiom with respect to the FODAL semantics. The transfer of completeness result from the simple join $\boldsymbol{QS}4 \oplus \boldsymbol{QKD}$ to the resulting $FODAL$ system is then proved in a manner described in [13] utilizing the method of canonical models.

The validity of $\Box\phi \rightarrow \boldsymbol{O}\phi$ follows as a result of Proposition 3.1.1, hence soundness is straightforward.

Definition 29 (Canonical model)**.**
$\mathfrak{M}^c = \langle \mathcal{W}^c, R^c_{\boldsymbol{O}}, R^c_{\Box}, \mathfrak{D}^c, \mathfrak{I}^c \rangle$ is the *canonical model* for the $FODAL$ axiom system, if $\mathfrak{D}^c$, $\mathfrak{I}^c$ and $\mathcal{W}^c$ – the set of all maximal consistent sets of formulae, are defined as in [24] and for all $w, w' \in \mathcal{W}^c$, $wR^c_\circ w'$ if and only if $\{\phi \mid \circ\,\phi \in w\} \subseteq w'$, where $\circ \in \{\Box, \boldsymbol{O}\}$.

The crucial fact concerning model $\mathfrak{M}^c$ is that the following *truth lemma* holds for every formula Φ and possible world $w \in \mathcal{W}^c$ [24]:

$$\Phi \in w \quad \text{if and only if} \quad \mathfrak{M}^c, w \vDash_\sigma \Phi \tag{3.19}$$

It was also shown in [13] that if the canonical model for some axiomatization L happens to be an $\boldsymbol{L}$-model, then completeness follows since if some Φ is not a theorem of L, there is an $\boldsymbol{L}$-model in which Φ fails – the canonical one. Thus, completeness proof can be concluded by verifying that constructed canonical model $\mathfrak{M}^c$ is actually a **FODAL** model.

Since the simple join $\boldsymbol{QS4} \oplus \boldsymbol{QKD}$ is complete, it follows that $R^c_{\boldsymbol{O}}$ is serial and $R^c_{\Box}$ is a preorder. Therefore it is left to show that $R^c_{\boldsymbol{O}} \subseteq R^c_{\Box}$.

Now suppose $w \in \mathcal{W}^c$. Assume $\Box X \in w$. Since $\Box\phi \to \boldsymbol{O}\phi$ is an axiom of FODAL and possible worlds of the canonical model $\mathfrak{M}^c$ are maximally consistent, so deductively closed, it follows that $\boldsymbol{O}X \in w$. Then, by definition of accessibility relations in the canonical model, $X \in v_i$, for each v_i such that $wR^c_{\boldsymbol{O}}v_i$. Therefore $\{X \mid \Box X \in w\} \subseteq v_i$. By definition of $R^c_{\Box}$ it follows that $wR^c_{\Box}v_i$, which proves that $R^c_{\boldsymbol{O}} \subseteq R^c_{\Box}$.

Hence, the constructed canonical model $\mathfrak{M}^c$ is actually a **FODAL** model what proves the completeness of the $FODAL$ axiom system with respect to the class of **FODAL** frames. □

3.2 Modeling SBVR vocabulary and rules with FODAL

3.2.1 The meaning of SBVR conceptual schema

Given an SBVR conceptual schema $\mathcal{S}$ – as informally introduced in Section 2.2 – we define the following translation $\tau(\cdot)$ from elements of $\mathcal{S}$ to notions of first-order deontic-alethic logic:

- For each noun concept A from $\mathcal{S}$, $\tau(A)$ is an unary predicate in FODAL.
- For each verb concept R from $\mathcal{S}$, $\tau(R)$ is an n-ary predicate in FODAL ($n \geq 2$).
- Recall that an SBVR business rule may be represented by an expression resulted from application of modalities and boolean connectives to a set of closed first-order formulae $\phi_{\hat{r}_i}$, as defined in Section 2.2.3. Then for each SBVR rule r from $\mathcal{S}$, its FODAL formalization $\tau(r)$ is defined inductively as follows:
  - $\tau(\hat{r}) = \phi_{\hat{r}}$, where $\hat{r}$ is an non-modal SBVR expression and $\phi_{\hat{r}}$ is its first-order translation,
  - $\tau(\Box\hat{r}) = \Box\tau(\hat{r})$,
  - $\tau(\boldsymbol{O}\hat{r}) = \boldsymbol{O}\tau(\hat{r})$,
  - $\tau(\neg r) = \neg\tau(r)$,
  - $\tau(r_1 \circ r_2) = \tau(r_1) \circ \tau(r_2), \circ \in \{\wedge, \vee, \to, \leftrightarrow\}$, where r_1 and r_2 are rule formulations.

Example 3.2.1. Assume the following set of business rules, expressed in Structured English:

r_1 = `Each` <u>`car rental`</u> `is insured by exactly one` <u>`credit card`</u>`.`

r_2 = `Each` <u>`luxury car rental`</u> `is a` <u>`car rental`</u>`.`

r_3 = **`It is obligatory that`** `each` <u>`luxury car rental`</u> `is insured by at least two` <u>`credit cards`</u>`.`

Then the corresponding FODAL formulas are the following:

$$\tau(r_1) = \forall x \exists^1 y(CarRental(x) \land Insured(x,y)),$$

$$\tau(r_2) = \forall x(LuxuryCarRental(x) \rightarrow CarRental(x)),$$

$$\tau(r_3) = \boldsymbol{O}(\forall x \exists^{\geq 2} y(LuxuryCarRental(x) \land Insured(x,y))).$$

While our $FODAL$ formalization of SBVR rules provides logical mechanism supporting rule formulations with multiple occurrences of modalities, SBVR standard mostly focuses on normalized business constraints [38, p.108] that may be expressed by rule statements of the form of atomic modal sentences or by statements reducible to such a form via mechanisms provided by $FODAL$ axiomatization. As a matter of fact, restricting the domain of interest only to such atomic modal rule formulations allows to obtain some useful results concerning satisfiability reduction and connection to standard logics, as will be shown later.

If, however, a rule statement can not be normalized using $FODAL$ axioms, then it might be either reformulated manually by a business domain expert so that to end up with a normalized semantically-equivalent constraint, or, in case it is neither possible, the rule can be retained *as is* and the whole SBVR conceptual schema should then be considered purely with respect to $FODAL$ formal semantics.

Hereafter in this work we will only consider SBVR rules expressible in one of the following forms of atomic modal sentences:

$$\Box\phi \qquad \Diamond\phi \qquad \boldsymbol{O}\phi \qquad \boldsymbol{P}\phi \tag{3.20}$$

where ϕ is any closed well-formed formula of first-order logic.

In the case of having negation in front of the modal operator, we assume application of the standard modal negation equivalences in order to obtain the basic form of the initial rule.

Definition 30 (FODAL regulation)**.**

A *FODAL regulation* Σ is a set of FODAL atomic modal sentences formalizing structural and operational rules of an SBVR conceptual schema S. We introduce the following designations:

$$\tau(r_\Box) = \Box\eta, \quad \tau(r_\Diamond) = \Diamond\pi,$$
$$\tau(r_{\boldsymbol{O}}) = \boldsymbol{O}\theta, \quad \tau(r_{\boldsymbol{P}}) = \boldsymbol{P}\rho,$$
$$\Sigma = \{\Box\eta_1, ..., \Box\eta_k, \Diamond\pi_1, ..., \Diamond\pi_l, \boldsymbol{O}\theta_1, ..., \boldsymbol{O}\theta_m, \boldsymbol{P}\rho_1, ..., \boldsymbol{P}\rho_n\} \tag{3.21}$$

The regulation Σ can also be represented as a conjunction of all formulae of the set:

$$\Sigma_\land = \bigwedge_{i=1}^{k} \Box\eta_i \land \bigwedge_{i=1}^{l} \Diamond\pi_i \land \bigwedge_{i=1}^{m} \boldsymbol{O}\theta_i \land \bigwedge_{i=1}^{n} \boldsymbol{P}\rho_i \tag{3.22}$$

where every $\eta_i, \pi_i, \theta_i, \rho_i$ is a closed first-order logic formula.
A set $\Sigma_\Box = \{\Box\eta_1, ..., \Box\eta_k, \Diamond\pi_1, ..., \Diamond\pi_l\}$ is called a *FODAL structural regulation*.
A set $\Sigma_{\boldsymbol{O}} = \{\boldsymbol{O}\theta_1, ..., \boldsymbol{O}\theta_m, \boldsymbol{P}\rho_1, ..., \boldsymbol{P}\rho_n\}$ is called a *FODAL operational regulation*.

3.2.2 Reasoning tasks

The final objective of the proposed formalization is to provide an automation solution with reasoning support for SBVR business modeling and business processes monitoring. When reasoning about some particular universe of discourse, consistency is essential, since once a logical inconsistency is accepted, it is possible to deduce anything (including rubbish) from it – an extreme case of the GIGO (Garbage In Garbage Out) principle.

According to Halpin [21], there are two types of garbage: logical and factual. Inconsistent designs of the schema S contain logical garbage. For example, one might declare two business rules that contradict one another. On the other hand, factual errors may arise while populating the schema S with a set of ground facts describing the current state of business world. For example, a declared rule that

`Each rental car is owned by at most one branch`

may be violated by some fact population that assigns some `rental car` to more than one `branch`.

In the following section we define several tasks which address described issues.

Definition 31 (Consistency of a set of business rules)**.**
Assume a FODAL regulation Σ representing a set of structural and operative business rules. The *task of consistency check for* Σ is defined as procedure which analyzes the given set Σ and decides whether none the rules contradict with each other, i.e. there is no formula ψ such that $\Sigma \vdash \psi$ and $\Sigma \vdash \neg\psi$, i.e. $\Sigma \nvdash \bot$.
A FODAL regulation Σ is called *internally inconsistent* when the specified constraints do not contradict each other when the system is populated.
A *minimal inconsistent set* $\Sigma_\bot \subseteq \Sigma$ is a subset of Σ such that $\Sigma_\bot \vdash \bot$ and $\forall \Delta \subset \Sigma_\bot, \Delta \nvdash \bot$.

According to the completeness of the FODAL logic we have that $\Sigma \nvdash \psi$ if and only if there exists a **FODAL** model $\mathfrak{M}$ and a possible world w in it, such that $\mathfrak{M}, w \vDash \Sigma \wedge \neg\psi$.

Definition 32 (Types of inconsistency)**.**
We distinguish several types of inconsistency depending on types of modalities of rules involved.
The set Σ is called *alethic inconsistent* if it is inconsistent and the minimal inconsistent set $\Sigma_\bot$ contains formulae of only alethic nature, i.e. $\Sigma_\bot \subseteq \Sigma_\Box$.
The set Σ is called *deontic inconsistent* if it is inconsistent and the minimal inconsistent set $\Sigma_\bot$

contains formulae of only deontic nature, i.e. $\Sigma_\perp \subseteq \Sigma_{\boldsymbol{O}}$.
Otherwise, if $\Sigma_\perp \subseteq \Sigma_\Box \cup \Sigma_{\boldsymbol{O}}$, the set Σ is called *cross inconsistent*.

Examples of possible sources of inconsistency are the following:

- *"something is obligatory and at the same time it is not possible"*: $\boldsymbol{O}X \wedge \Box\neg X$, cross inconsistency;
- *"something is permitted and forbidden at the same time"*: $\boldsymbol{P}X \wedge \boldsymbol{O}\neg X$, deontic inconsistency;
- other combinations of conflicting rules: $\Box(P \to Q) \wedge \boldsymbol{O}P \wedge \boldsymbol{O}\neg Q$, cross inconsistency.

Since Theorem 3.1.2 proves the completeness of the FODAL logic, in order to check the consistency of a set of rules it is sufficient to state the satisfiability of the conjunction of all formulae of the set:

$$\Sigma_\wedge = \bigwedge_{i=1}^{k} \Box\eta_i \wedge \bigwedge_{i=1}^{l} \Diamond\pi_i \wedge \bigwedge_{i=1}^{m} \boldsymbol{O}\theta_i \wedge \bigwedge_{i=1}^{n} \boldsymbol{P}\rho_i$$

Definition 33 (Maximal consistent subset)**.**
Assume a FODAL regulation Σ. Then the *task of finding a maximal consistent subset of* Σ is defined as procedure which determines a possible maximal subset $\Sigma_{max} \subseteq \Sigma$, such that:

- Σ_{max} is consistent,
- Σ_{max} is maximal with respect to $\subseteq$,

Therefore this task may also be reduced to the satisfiability of a subformula of $\Sigma_\wedge$ also being a conjunction of FODAL atomic modal sentences.

Definition 34 (Conformity of fact population)**.**
Assume a FODAL regulation Σ representing a set of structural and operative business rules S and a knowledge base KB – a set of ground facts representing some fact population F. KB expresses the current state of business world, which is defined as an interpretation of a business model defined in terms of SBVR concepts. Then *the task of conformity of fact population* is a procedure which determines whether the set KB is compliant with all the business rules of the model, i.e. whether all the necessity claims are valid and no obligations are broken.

For the conformity check we consider only those business rules which possess some restrictions on the fact populations, namely, necessity and obligation claims:

$$\Sigma^{restrict} = \{\Box\phi_i \mid \Box\phi_i \in \Sigma\} \cup \{\boldsymbol{O}\phi_i \mid \boldsymbol{O}\phi_i \in \Sigma\} \tag{3.23}$$

We denote the set of first-order formulae under the scope of modal operators as

$$\Lambda^{restrict} = \{\phi_i \mid \text{either } \Box\phi_i \in \Sigma^{restrict} \text{ or } \boldsymbol{O}\phi_i \in \Sigma^{restrict}\}, \tag{3.24}$$

Then, since the set of ground facts KB may be seen as a first-order interpretation at some possible world w, the task of conformity check consists in asserting the claim that KB satisfies each formula $\phi_i \in \Lambda^{restrict}$.

Definition 35 (Types of nonconformity)**.**
The knowledge base KB is called *strongly nonconforming* with respect to a FODAL regulation Σ if there is a FODAL necessity rule $\Box\psi \in \Sigma$ such that ψ evaluates to $false$ under KB.
The knowledge base KB is called *nonconforming with respect to obligation(s)* of a FODAL regulation Σ if there is a FODAL obligation rule $\boldsymbol{O}\psi \in \Sigma$ such that ψ evaluates to $false$ under KB.

3.3 Satisfiability of FODAL regulations

3.3.1 Reduction to satisfiability in first-order logic

In this section we will show that if we restrict the domain of interest to business rules expressed as FODAL atomic modal sentences, we can reduce the satisfiability problem (and thus consistency) of such fragment of FODAL to first-order satisfiability, what by-turn enables the transfer of logical results from well-studied reduction classes for satisfiability of predicate first-order logic [8].

Definition 36.
A FODAL regulation Σ expressed as a FODAL formula 3.22 is satisfiable with respect to a valuation σ, denoted $Sat_\sigma(\Sigma)$, if and only if there is a model $\mathfrak{M} = \langle \mathfrak{F}, \mathfrak{I} \rangle$, based on a **FODAL** frame $\mathfrak{F} = \langle \mathcal{W}, R_{\boldsymbol{O}}, R_\Box, \mathfrak{D} \rangle$, and a possible world $w \in \mathcal{W}$ such that

$$\mathfrak{M}, w \models_\sigma \bigwedge_{i=1}^{k} \Box\eta_i \wedge \bigwedge_{i=1}^{l} \Diamond\pi_i \wedge \bigwedge_{i=1}^{m} \boldsymbol{O}\theta_i \wedge \bigwedge_{i=1}^{n} \boldsymbol{P}\rho_i \tag{3.25}$$

Since FODAL atomic modal sentences, being members of the regulation Σ, may only contain closed first-order formulae and thus do not contain any free variables, if 3.25 holds with respect to some valuation σ, then it holds for any other valuation σ'. Hence, hereafter we will omit mentioning σ in computations.

Now by definition of satisfiability of conjunction in FODAL we have that

$Sat(\Sigma) = true$ if and only if there exists a **FODAL** frame $\mathfrak{F} = \langle \mathcal{W}, R_{\boldsymbol{O}}, R_{\Box}, \mathfrak{D} \rangle$, an interpretation $\mathfrak{I}$ and a possible world $w \in \mathcal{W}$, such that

$$\langle \mathfrak{F}, \mathfrak{I} \rangle, w \vDash \bigwedge_{i=1}^{k} \Box \eta_i \text{ and} \tag{3.26a}$$

$$\langle \mathfrak{F}, \mathfrak{I} \rangle, w \vDash \bigwedge_{i=1}^{l} \Diamond \pi_i \text{ and} \tag{3.26b}$$

$$\langle \mathfrak{F}, \mathfrak{I} \rangle, w \vDash \bigwedge_{i=1}^{m} \boldsymbol{O} \theta_i \text{ and} \tag{3.26c}$$

$$\langle \mathfrak{F}, \mathfrak{I} \rangle, w \vDash \bigwedge_{i=1}^{n} \boldsymbol{P} \rho_i. \tag{3.26d}$$

Proceeding with expanding the conjunctions and applying Definition 24 for modal operators, it can be shown that

$Sat(\Sigma) = true$ if and only if $\exists \mathfrak{F} = \langle \mathcal{W}, R_{\boldsymbol{O}}, R_{\Box}, \mathfrak{D} \rangle, \exists \mathfrak{I}, \exists w \in \mathcal{W}$ such that

$$\forall v \in \mathcal{W} \text{ such that } wR_{\Box}v,\ \langle \mathfrak{F}, \mathfrak{I} \rangle, v \vDash \bigwedge_{i=1}^{k} \eta_i \text{ and} \tag{3.27a}$$

$$\exists v_j \in \mathcal{W} \text{ such that } wR_{\Box}v_j,\ \langle \mathfrak{F}, \mathfrak{I} \rangle, v_j \vDash \pi_j,\ \forall j = \overrightarrow{1 \ldots l} \text{ and} \tag{3.27b}$$

$$\forall v \in \mathcal{W} \text{ such that } wR_{\boldsymbol{O}}v,\ \langle \mathfrak{F}, \mathfrak{I} \rangle, v \vDash \bigwedge_{i=1}^{m} \theta_i \text{ and} \tag{3.27c}$$

$$\exists u_j \in \mathcal{W} \text{ such that } wR_{\boldsymbol{O}}u_j,\ \langle \mathfrak{F}, \mathfrak{I} \rangle, u_j \vDash \rho_j,\ \forall j = \overrightarrow{1 \ldots n}. \tag{3.27d}$$

Then, taking into account the properties of accessibility relations of the **FODAL** frame $\mathfrak{F}$ (in particular, $R_{\boldsymbol{O}} \subseteq R_{\Box}$) and using the definition of an interpretation in first-order deontic-alethic logic, we have that

$$Sat(\Sigma) = true \quad \text{if and only if} \quad \exists \mathfrak{F} = \langle \mathcal{W}, R_{\boldsymbol{O}}, R_{\Box}, \mathfrak{D} \rangle, \exists \mathfrak{I}, \exists w \in \mathcal{W} \quad \text{such that}$$

$$\forall v \in \mathcal{W} \quad \text{such that} \quad wR_{\Box}v, \; \mathfrak{I}(v) \vDash^{FOL} \bigwedge_{i=1}^{k} \eta_i \;\; \text{and} \tag{3.28a}$$

$$\forall v \in \mathcal{W} \quad \text{such that} \quad wR_{\boldsymbol{O}}v, \; \mathfrak{I}(v) \vDash^{FOL} \bigwedge_{i=1}^{m} \theta_i \wedge \bigwedge_{i=1}^{k} \eta_i \;\; \text{and} \tag{3.28b}$$

$$\exists v_j \in \mathcal{W} \quad \text{such that} \quad wR_{\Box}v_j, \; \mathfrak{I}(v_j) \vDash^{FOL} \pi_j \wedge \bigwedge_{i=1}^{k} \eta_i, \;\; \forall j = \overrightarrow{1 \dots l} \;\; \text{and} \tag{3.28c}$$

$$\exists u_j \in \mathcal{W} \quad \text{such that} \quad wR_{\boldsymbol{O}}u_j, \; \mathfrak{I}(u_j) \vDash^{FOL} \rho_j \wedge \bigwedge_{i=1}^{m} \theta_i \wedge \bigwedge_{i=1}^{k} \eta_i, \;\; \forall j = \overrightarrow{1 \dots n} \tag{3.28d}$$

Therefore we showed that Σ is satisfiable if and only if for some **FODAL** frame $\mathfrak{F}^*$ and some world w there exists a set of first-order logic interpretations $\mathfrak{I}^*(w')$, assigned to each possible world w', such that (3.28a - 3.28d) holds. Which proves the following theorem:

Theorem 3.3.1. *A FODAL regulation* $\Sigma_\wedge = \bigwedge_{i=1}^{k} \Box\eta_i \wedge \bigwedge_{i=1}^{l} \Diamond\pi_i \wedge \bigwedge_{i=1}^{m} \boldsymbol{O}\theta_i \wedge \bigwedge_{i=1}^{n} \boldsymbol{P}\rho_i$ *is FODAL-satisfiable if and only if each of the following formulae* $\mathcal{N}, \mathcal{O}, \mathcal{Q}_j, \mathcal{P}_j$ *is independently first-order satisfiable:*

$$\mathcal{N} = \bigwedge_{i=1}^{k} \eta_i \tag{3.29a}$$

$$\mathcal{O} = \bigwedge_{i=1}^{m} \theta_i \wedge \bigwedge_{i=1}^{k} \eta_i \tag{3.29b}$$

$$\mathcal{Q}_j = \pi_j \wedge \bigwedge_{i=1}^{k} \eta_i, \;\; \forall j = \overrightarrow{1 \dots l} \tag{3.29c}$$

$$\mathcal{P}_j = \rho_j \wedge \bigwedge_{i=1}^{m} \theta_i \wedge \bigwedge_{i=1}^{k} \eta_i, \;\; \forall j = \overrightarrow{1 \dots n} \tag{3.29d}$$

Observe that satisfiability of $\mathcal{N}$ follows naturally from satisfiability of any $\mathcal{Q}_j$. The same holds for $\mathcal{O}$ and $\mathcal{P}_j$ respectively. However, the satisfiability checks for 3.29a and 3.29b should be examined explicitly, since Σ may only contain necessity and obligation rules. Moreover, such definition allows to detect the actual source of unsatisfiability of the FODAL regulation Σ.

Modularity of the approach: It should be noted that the developed approach of satisfiability reduction possesses a property of *modularity*, i.e. it does not depend on the formalism behind the rule bodies η_i, θ_i, π_i and ρ_i, as long as formalism-specific satisfiability relation is provided.

3.3.2 Canonical model for FODAL regulations

This section outlines a method to construct a *canonical pointed model* $\mathfrak{M}^\Sigma = \langle \mathfrak{F}^\Sigma, \mathfrak{I}^\Sigma, w_\Box \rangle$ for a given FODAL regulation Σ, which can be used for satisfiability checking.

Definition 37 (Σ-canonical pointed model)**.**
Assume a FODAL regulation Σ, expressed as a conjunction of FODAL formulae:

$$\Sigma_\wedge = \bigwedge_{i=1}^{k} \Box\eta_i \wedge \bigwedge_{i=1}^{l} \Diamond\pi_i \wedge \bigwedge_{i=1}^{m} \boldsymbol{O}\theta_i \wedge \bigwedge_{i=1}^{n} \boldsymbol{P}\rho_i \tag{3.30}$$

Then a Σ*-canonical pointed model* $\mathfrak{M}^\Sigma = \langle \mathfrak{F}^\Sigma, \mathfrak{I}^\Sigma, w_\Box \rangle$ is constructed in the following way:

- $\mathfrak{F}^\Sigma = \langle \mathcal{W}^\Sigma, w_\Box, R_\Box^\Sigma, R_{\boldsymbol{O}}^\Sigma, \mathfrak{D} \rangle$ is a pointed **FODAL** frame with root $w_\Box \in \mathcal{W}^\Sigma$.
- $\mathcal{W}^\Sigma = \{w_\Box, w_{\boldsymbol{O}}, v_1, ..., v_l, u_1, ..., u_n\}$ is the set of all possible worlds.
- $R_\Box^\Sigma = \{(w_\Box, v) \mid v \in \mathcal{W}^\Sigma\}^*$ is a necessity accessibility relation and $\{\cdot\}^*$ denotes the operation of reflexive transitive closure.
- $R_{\boldsymbol{O}}^\Sigma = \{(w_\Box, w_{\boldsymbol{O}})\} \cup \{(w_\Box, u_j) \mid \forall j = \overrightarrow{1 \dots n}\} \cup R_{\boldsymbol{O}}^{\boldsymbol{KD}}$ is an obligation accessibility relation and $R_{\boldsymbol{O}}^{\boldsymbol{KD}}$ is any complement set ensuring the seriality of $R_{\boldsymbol{O}}^\Sigma$, for instance: $R_{\boldsymbol{O}}^{\boldsymbol{KD}} = \{(u_j, u_j) \mid \forall j = \overrightarrow{1 \dots n}\} \cup \{(w_{\boldsymbol{O}}, w_{\boldsymbol{O}})\}$.
- $\mathfrak{D}$ is some domain function mapping worlds of $\mathcal{W}^\Sigma$ to non-empty sets.
- $\mathfrak{I}^\Sigma = \{(v, \mathfrak{I}_v) \mid \forall v \in \mathcal{W}^\Sigma\}$ is a FODAL interpretation such that

$$\mathfrak{I}_{w_\Box} \models \bigwedge_{i=1}^{k} \eta_i \tag{3.31a}$$

$$\mathfrak{I}_{w_{\boldsymbol{O}}} \models \bigwedge_{i=1}^{m} \theta_i \wedge \bigwedge_{i=1}^{k} \eta_i \tag{3.31b}$$

$$\mathfrak{I}_{v_j} \models \bigwedge_{i=1}^{k} \eta_i \wedge \pi_j, \ \ \forall j = \overrightarrow{1 \dots l} \tag{3.31c}$$

$$\mathfrak{I}_{u_j} \models \bigwedge_{i=1}^{m} \theta_i \wedge \bigwedge_{i=1}^{k} \eta_i \wedge \rho_j, \ \ \forall j = \overrightarrow{1 \dots n} \tag{3.31d}$$

Proposition 3.3.2. *Given a FODAL regulation Σ, if a Σ-canonical pointed model $\mathfrak{M}^\Sigma = \langle \mathfrak{F}^\Sigma, \mathfrak{I}^\Sigma, w_\Box \rangle$ exists, then Σ is true in this model.*

Proof. This result follows directly from the construction of a Σ-canonical pointed model $\mathfrak{M}^\Sigma$ since the constructed frame is a **FODAL** frame and the definition of $\mathfrak{I}^\Sigma$ coincides with the result (3.28a - 3.28d) obtained in the previous section. Thus, if it is possible to define $\mathfrak{I}^\Sigma$ such that it meets the conditions (3.31a - 3.31d), then $\mathfrak{I}^\Sigma$ satisfies Σ at the world $w_\Box$, hence Σ is true in the pointed model $\mathfrak{M}^\Sigma$. □

Theorem 3.3.3. *A FODAL regulation Σ is satisfiable if and only if there exists a Σ-canonical pointed model $\mathfrak{M}^\Sigma = \langle \mathfrak{F}^\Sigma, \mathfrak{I}^\Sigma, w_\Box \rangle$.*

Proof. The reverse direction follows from Proposition 3.3.2, therefore it remains to prove the following statement: if $\mathfrak{M}^\Sigma$ can not be constructed, then Σ is not satisfiable in any other model.

Assume that the Σ-canonical pointed model $\mathfrak{M}^\Sigma$ can not be constructed, which means that it is not possible to define such a FODAL interpretation $\mathfrak{I}^\Sigma$ that meets conditions (3.31a - 3.31d). So, either of the following holds:

(a) $\forall$ FOL interpretations $\mathfrak{I}_{w_\Box}$, $\mathfrak{I}_{w_\Box} \not\models \bigwedge_{i=1}^{k} \eta_i$

(b) $\forall$ FOL interpretations $\mathfrak{I}_{w_{\boldsymbol{O}}}$, $\mathfrak{I}_{w_{\boldsymbol{O}}} \not\models \bigwedge_{i=1}^{m} \theta_i \wedge \bigwedge_{i=1}^{k} \eta_i$

(c) $\exists j \in [1, l]$ such that $\forall$ FOL interpretations $\mathfrak{I}_{v_j}$, $\mathfrak{I}_{v_j} \not\models \bigwedge_{i=1}^{k} \eta_i \wedge \pi_j$

(d) $\exists j \in [1, n]$ such that $\forall$ FOL interpretations $\mathfrak{I}_{u_j}$, $\mathfrak{I}_{u_j} \not\models \bigwedge_{i=1}^{m} \theta_i \wedge \bigwedge_{i=1}^{k} \eta_i \wedge \rho_i$

Now assume that condition 3.31a cannot be met, hence (a) holds, and suppose that Σ is satisfiable, so there exists a model $\mathfrak{M}' = \langle \mathfrak{F}', \mathfrak{I}' \rangle$ and a possible world w' of this model such that

$$\mathfrak{M}', w' \models \bigwedge_{i=1}^{k} \Box\eta_i \wedge \bigwedge_{i=1}^{l} \Diamond\pi_i \wedge \bigwedge_{i=1}^{m} \boldsymbol{O}\theta_i \wedge \bigwedge_{i=1}^{n} \boldsymbol{P}\rho_i$$

Therefore, by the definition of conjunction, we have that $\mathfrak{M}', w' \models \bigwedge_{i=1}^{k} \Box\eta_i$. This means that, by definition of necessity, $\forall v$ such that $w' R_\Box v$, $\mathfrak{M}', v \models \bigwedge_{i=1}^{k} \eta_i$ and thus $\mathfrak{I}'(v) \models^{FOL} \bigwedge_{i=1}^{k} \eta_i$. However, this contradicts with the statement (a) and hence Σ is unsatisfiable. The same results can be obtained for the rest of the cases (b)-(d) in a similar way, which completes the proof. □

3.4 Reduction to monomodal logic $\boldsymbol{QK}$

As a matter of fact, the FODAL logic inherits the property of undecidability from both its component logics: standard predicate modal logics $\boldsymbol{QS}4$ and $\boldsymbol{QKD}$ are undecidable [24].

However, decidability results have been obtained for several well-studied fragments of quantified modal logics [39]. This section defines a *truth-preserving translation* of atomic modal sentences of the FODAL logic into standard predicate modal logic $\boldsymbol{QK}$, which allows to determine decidable fragments of the FODAL logic.

Definition 38 (Monomodal simulating pointed frame)**.**
Given a **FODAL** frame $\mathfrak{F} = \langle \mathcal{W}, R_{\boldsymbol{O}}, R_{\Box}, \mathfrak{D} \rangle$ and a possible world $w_0 \in \mathcal{W}$, a *monomodal simulating pointed frame* $\mathfrak{F}^s_{w_0}$ is defined as a tuple $\langle \mathcal{W}^s, R^s, \mathfrak{D}^s, w_0 \rangle$, such that:

- $\mathcal{W}^s$ includes w_0 and all its deontic and alethic successors:
  $\mathcal{W}^s = \{w_0\} \cup \{v \mid (w_0, v) \in R_{\boldsymbol{O}}\} \cup \{v \mid (w_0, v) \in R_{\Box}\} = |\text{since } R_{\boldsymbol{O}} \subseteq R_{\Box} \text{ and } R_{\Box} \text{ is reflexive}| = \{v \mid (w_0, v) \in R_{\Box}\}$.
- $R^s = \{(w_0, v) \mid (w_0, v) \in R_{\Box}\}$, and $\Box^s$, $\Diamond^s$ are modal operators associated with R^s.
- $\mathfrak{D}^s$ is a domain function on $\mathcal{W}^s$ such that $\mathfrak{D}^s(v) = \mathfrak{D}(v) \cup \{\pi^{\mathfrak{D}^s}\}, \forall v \in \mathcal{W}^s$, where $\pi^{\mathfrak{D}^s} \notin \mathfrak{D}$ is a new service domain symbol.

Since the definition of R^s does not preserve specific properties of $R_{\boldsymbol{O}}$ and $R_{\Box}$, the resulting frame $\mathfrak{F}^s_{w_0}$ does not belong either to serial or transitive or reflexive class of frames and therefore can be classified as a **K**-frame.

We now define the translation schema for FODAL regulations.

Definition 39 (Monomodal translation)**.**
Given a FODAL regulation Σ expressed as a conjunction of FODAL atomic modal sentences 3.30, a *monomodal translation of regulation* $MTR(\Sigma_{\wedge})$ is defined inductively as follows:

$$
\begin{aligned}
MTR(\phi) &= \phi, \text{ where } \phi \text{ is an objective FODAL formula,}\\
MTR(\phi_1 \wedge \phi_2) &= MTR(\phi_1) \wedge MTR(\phi_2), \text{ where } \phi_1 \text{ and } \phi_2 \text{ are FODAL atomic modal sentences,}\\
MTR(\Box\psi) &= \Box^s MTR(\psi),\\
MTR(\boldsymbol{O}\psi) &= \Box^s(\neg\Pi \rightarrow MTR(\psi)),\\
MTR(\Diamond\psi) &= \Diamond^s(MTR(\psi) \wedge \Pi),\\
MTR(\boldsymbol{P}\psi) &= \Diamond^s(MTR(\psi) \wedge \neg\Pi),
\end{aligned}
$$

where ψ is a objective FODAL formula and Π is a 0-place predicate symbol, i.e. propositional letter, encapsulating the nature of the original modality of the rules of possibility and permission.

Definition 40 (Simulated pointed model)**.**
Given a **FODAL** model $\mathfrak{M} = \langle \mathfrak{F}, \mathfrak{I} \rangle$ and a possible world $w_0 \in \mathcal{W}$, a *simulated pointed model* $\mathfrak{M}^s_{w_0}$ is defined as a tuple $\langle \mathfrak{F}^{\mathfrak{s}}_{\mathfrak{w}_\mathfrak{o}}, \mathfrak{I}^{\mathfrak{s}} \rangle$ such that:

- $\mathfrak{F}^s_{w_0} = \langle \mathcal{W}^s, R^s, \mathfrak{D}^s, w_0 \rangle$ is a monomodal simulating pointed frame for $\mathfrak{F} = \langle \mathcal{W}, R_{\boldsymbol{O}}, R_{\Box}, \mathfrak{D} \rangle$ and a possible world $w_0 \in \mathcal{W}$,
- $\mathfrak{I}^s$ is a first-order interpretation on the frame $\mathfrak{F}^s_{w_0}$ such that:
  - For each $v \in \mathcal{W}^s$ and for every n-place predicate P, $\mathfrak{I}^s(P, v) = \mathfrak{I}(P, v)$,
  - For each $v \in \mathcal{W}^s$ such that $(w_0, v) \in R_{\boldsymbol{O}}$, $\mathfrak{I}^s(\Pi, v) = \varnothing$,
  - For each $v \in \mathcal{W}^s$ such that $(w_0, v) \notin R_{\boldsymbol{O}}$, $\mathfrak{I}^s(\Pi, v) = \{\pi^{\mathfrak{D}^s}\}$.

We now state formally that the translation given above is truth-preserving with respect to varying domain semantics.

Theorem 3.4.1. *For any FODAL regulation Σ, any **FODAL** model $\mathfrak{M}$ and any possible world w_0 of a model, we have that*

$$\mathfrak{M}, w_0 \vDash \Sigma \quad \textit{if and only if} \quad \mathfrak{M}^s_{w_0}, w_0 \vDash MTR(\Sigma), \tag{3.32}$$

where $\mathfrak{M}^s_{w_0}$ is a simulated pointed model *for $\mathfrak{M}$ and w_0.*

Proof. Since a FODAL regulation Σ may be expressed as a conjunction of FODAL atomic modal sentences 3.30, we can use Definition 24 of a truth in a model to expand the formula and then prove the proposition by induction on the structure of $\Sigma_\wedge$.
Therefore it is necessary to prove the statement 3.32 for the 4 basic cases corresponding to different modalities:

(a) $\mathfrak{M}, w_0 \vDash \Box\phi$ iff $\mathfrak{M}^s_{w_0}, w_0 \vDash \Box^s\phi$.
Assume that $\mathfrak{M}, w_0 \vDash \Box\phi$ holds.
By the definition of the modal operator $\Box$, it is the case if and only if $\forall v, w_0 R_\Box v : \mathfrak{M}, v \vDash \phi$.
Then, by the definition of interpretation in $\mathfrak{M}$, we have that $\forall v, w_0 R_\Box v : \mathfrak{I}(v) \vDash^{FOL} \phi$.
Now, as $R^s = \{(w_0, v) \mid (w_0, v) \in R_\Box\}$, it follows that $\mathfrak{M}, w_0 \vDash \Box\phi$ if and only if $\forall v, w_0 R^s_\Box v : \mathfrak{I}(v) \vDash^{FOL} \phi$.
Since the first-order interpretation $\mathfrak{I}^s$ disagrees with $\mathfrak{I}$ only on the newly introduced predicate symbol Π and ϕ does not contain Π as a subformula, then $\mathfrak{I}(v) \vDash^{FOL} \phi$ if and only if $\mathfrak{I}^s(v) \vDash^{FOL} \phi$.
Therefore we have that $\mathfrak{M}, w_0 \vDash \Box\phi$ iff $\forall v, w_0 R^s v : \mathfrak{I}^s(v) \vDash^{FOL} \phi$.
Hence, by definition of the modal operator $\Box^s$: $\mathfrak{M}, w_0 \vDash \Box\phi$ iff $\mathfrak{M}^s_{w_0}, w_0 \vDash \Box^s\phi$.

(b) $\mathfrak{M}, w_0 \models \boldsymbol{O}\phi$ iff $\mathfrak{M}^s_{w_0}, w_0 \models \Box^s(\neg\Pi \rightarrow \phi)$.
Assume that $\mathfrak{M}^s_{w_0}, w_0 \models \Box^s(\neg\Pi \rightarrow \phi)$ holds.
By the definition of $\Box^s$ and interpretation in $\mathfrak{M}^s$, it is the case if and only if
$\forall v, w_0 R^s v : (\ \mathfrak{I}^s(v) \models^{FOL} \Pi$ or $\mathfrak{I}^s(v) \models^{FOL} \phi\ )$.
However, $\mathfrak{I}^s(v) \models^{FOL} \Pi$ iff $(w_0, v) \notin R_{\boldsymbol{O}}$, by the definition of $\mathfrak{I}^s$.
Then, simplifying the set expressions for (w_0, v) and repeating the arguments on $\mathfrak{I}^s$ from the case (a), we have that $\mathfrak{M}^s_{w_0}, w_0 \models \Box^s(\neg\Pi \rightarrow \phi)$ iff $\forall v, (w_0, v) \in R_{\boldsymbol{O}} :\ \mathfrak{I}(v) \models^{FOL} \phi$.
Therefore $\mathfrak{M}^s_{w_0}, w_0 \models \Box^s(\neg\Pi \rightarrow \phi)$ iff $\mathfrak{M}, w_0 \models \boldsymbol{O}\phi$.

(c) $\mathfrak{M}, w_0 \models \Diamond\phi$ iff $\mathfrak{M}^s_{w_0}, w_0 \models \Diamond^s(\Pi \wedge \phi)$.
Assume that $\mathfrak{M}^s_{w_0}, w_0 \models \Diamond^s(\Pi \wedge \phi)$ holds. Then, similar to cases (a)–(b), the following sequence of statements proves the proposition for this particular case:
$\mathfrak{M}^s_{w_0}, w_0 \models \Diamond^s(\Pi \wedge \phi)$ iff $\exists v, w_0 R^s v :\ \mathfrak{I}^s, v \models^{FOL} \phi \wedge \Pi$.
$\mathfrak{I}^s, v \models^{FOL} \Pi$ iff $(w_0, v) \notin R_{\boldsymbol{O}}$.
$\mathfrak{M}^s_{w_0}, w_0 \models \Diamond^s(\Pi \wedge \phi)$ iff $\exists v, (w_0, v) \in R_{\Box} \setminus R_{\boldsymbol{O}} :\ \mathfrak{I}(v) \models^{FOL} \phi$.
$\mathfrak{M}^s_{w_0}, w_0 \models \Diamond^s(\Pi \wedge \phi)$ iff $\mathfrak{M}, w_0 \models \Diamond\phi$.

(d) $\mathfrak{M}, w_0 \models \boldsymbol{P}\phi$ iff $\mathfrak{M}^s_{w_0}, w_0 \models \Diamond^s(\neg\Pi \wedge \phi)$
Assume that $\mathfrak{M}^s_{w_0}, w_0 \models \Diamond^s(\neg\Pi \wedge \phi)$ holds. Then, similar to cases (a)–(c), the following sequence of statements proves the proposition for this particular case:
$\mathfrak{M}^s_{w_0}, w_0 \models \Diamond^s(\neg\Pi \wedge \phi)$ iff $\exists v, w_0 R^s v :\ \mathfrak{I}^s, v \models^{FOL} \phi \wedge \neg\Pi$.
$\mathfrak{I}^s, v \models^{FOL} \neg\Pi$ iff $(w_0, v) \in R_{\boldsymbol{O}}$.
$\mathfrak{M}^s_{w_0}, w_0 \models \Diamond^s(\neg\Pi \wedge \phi)$ iff $\exists v, (w_0, v) \in R_{\Box} \cap R_{\boldsymbol{O}} :\ \mathfrak{I}(v) \models^{FOL} \phi$.
$\mathfrak{M}^s_{w_0}, w_0 \models \Diamond^s(\neg\Pi \wedge \phi)$ iff $\mathfrak{M}, w_0 \models \boldsymbol{P}\phi$.

Therefore we proved that the translation MTR defined for FODAL regulations is truth-preserving and, thus, enables the transfer of decidability results from well-studied fragments of predicate modal logics [39] to FODAL. In particular, the following fragments of FODAL logic are decidable:

- the set of atomic modal sentences with at most two variables (denoted by $\mathcal{MAS}^2$),
- the set of *monadic* atomic modal sentences, all predicate symbols in which are at most unary(denoted by $\mathcal{MAS}^{mon}$),
- the set of atomic modal sentences, the modal operators in which are applied to subformulas from the *guarded fragment* of first-order logic [4] (denoted by $\mathcal{MAS}^{\mathcal{GF}}$).

□

CHAPTER 4

Implementation of automated reasoning support tool

In order to demonstrate the practical value of the results obtained in Section 3.3, in particular, a certain case of reduction of consistency of SBVR rules set to satisfiability in description logic, we developed a special tool which provides automated support for some reasoning tasks along with translation into an OWL2 ontology.

4.1 General description of the tool

Since the tool was developed in the context of the ONTORULE FP7 project [26], the set of SBVR rules is expressed as an ORM2 diagram or an equivalent CogNIAM schema [38, annex L]. This schema is then translated into the ORM2 Formal Syntax without any loss of information [15], which is then passed as an input to the developed tool. The CogNIAM schema to ORM2 Formal Syntax translation has been provided by a project partner and described in the public deliverable [9].

The implementation of the tool is written in Java and includes a parser for ORM2 Formal Syntax, a set of Java classes representing the ORM2 knowledge database, a translator into an OWL2 ontology and a modal reasoning engine using HermiT or FaCT++ as an underlying reasoner. The parser was developed using the JavaCC parser generator.

The tool provides the following functionality:

- Checking the consistency of a given set of rules
- Translating a given ORM2 schema into OWL2 ontology

The workflow diagram of the tool is depicted on Figure 4.1.

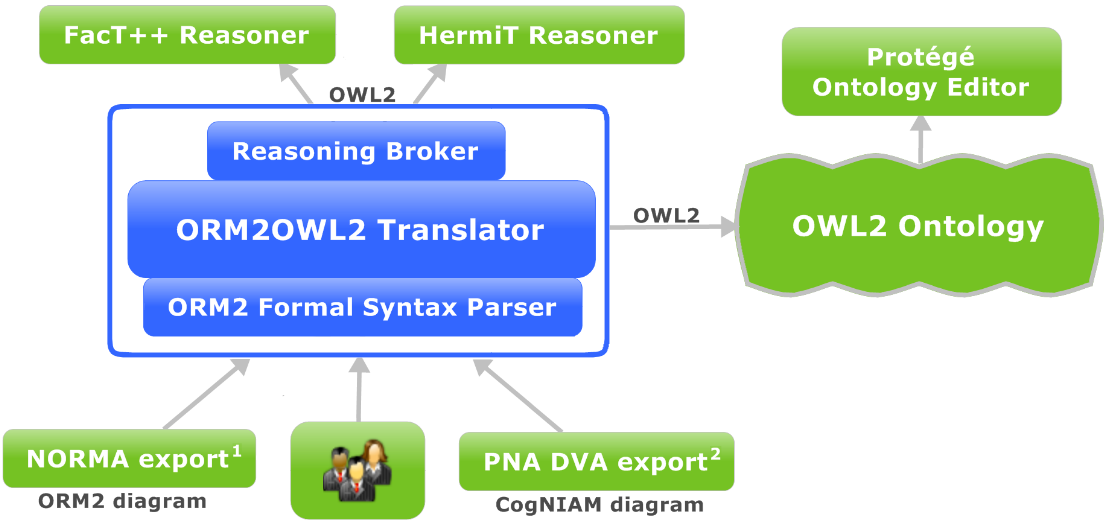


Figure 4.1: Workflow Diagram

[1] *Neumont ORM Architect for Visual Studio* [1]
[2] *PNA Group Discovery and Validation Assistant* [2], [9]

4.2 Input specifications

4.2.1 ORM2 Formal Syntax

In this section we recapitulate the definition of the ORM2 Formal Syntax which was developed in [15] with a humble contribution of the author of the thesis.

First, we define a signature Ω consisting of:

- a set $\mathcal{E}$ of *entity type* symbols;
- a set $\mathcal{V}$ of *value type* symbols;
- a set of *object type* symbols as $\mathcal{O} = \mathcal{E} \cup \mathcal{V}$;
- a set $\mathcal{R}$ of *relation* symbols;
- a set $\mathcal{A}$ of (attribute) *role* symbols;
- a set $\mathcal{D}$ of *domain* symbols, and

- a set Λ of pairwise disjoint sets of values;
- an extension function $\Lambda_{(\cdot)} : \mathcal{D} \to \Lambda$ associating each domain symbol $D \in \mathcal{D}$ to an extension Λ_D;
- a binary relation $\varrho \subseteq \mathcal{R} \times \mathcal{A}$ linking role symbols to relation symbols. We take the pair $R.a$ as the atomic elements of the syntax, and we call it *localized role*. Given a relation symbol R, $\varrho_R = \{R.a | R.a \in \varrho\}$ is the set of localized roles with respect to R; $arity(R) = |\varrho_R|$ is the arity of the relation R;
- for each relation symbol R, a bijection $\tau_R \colon \varrho_R \to [1..|\varrho_R|]$ mapping each element in ϱ_R to an element in the finite succession of natural numbers $[1..|\varrho_R|]$. We also define $\tau = \bigcup_{R \in \mathcal{R}} \tau_R$. The mapping τ_R guarantees a correspondence between role components and argument positions in a relation, so that we can freely choose between an 'attribute-based' and a 'positional-based' representation.

Then an ORM2 *conceptual schema Σ* over Ω consists of the following relations:

- $\mathsf{TYPE} \subseteq \varrho \times \mathcal{O}$ – relation representing *role connections*.
- $\mathsf{FREQ} \subseteq \wp(\varrho) \times (\mathbb{N} \times (\mathbb{N} \cup \{\infty\}))$ – relation defining *frequency constraints*. The introduced generalized version of the FREQ construct covers the following: (i) *external frequency* (i.e., occurrence frequency constraints that apply to single roles from different relations), and (ii) *internal frequency* (i.e., occurrence frequency constraints that apply to single roles from a single relation) constraints, as well as, their respective *compound* versions, where two or more roles are involved. Finally, the introduced syntax naturally supports the expression of *frequency ranges* (for a detailed introduction of the different constraints, see [21, p.272]).
- $\mathsf{MAND} \subseteq \wp(\varrho) \times \mathcal{O}$ – relation representing *mandatory role constraints*.
- $\mathsf{R\text{-}SET_H} \subseteq (\wp(\varrho) \times \wp(\varrho)) \times (\mu \colon \varrho \to \varrho)$, where μ is a partial bijection such that, for any $\langle A, B, \mu \rangle \in \mathsf{R\text{-}SET_H}$, $A = \{R.a | \mu(R.a) \in B\}$ and $\mathsf{H} = \{\mathsf{Sub}, \mathsf{Exc}\}$.
  $\mathsf{R\text{-}SET_{Sub}}$ represents *subset constraints* while $\mathsf{R\text{-}SET_{Exc}}$ represents *exclusion constraints* respectively.
- $\mathsf{O\text{-}SET_H} \subseteq 2^{\mathcal{O}} \times \mathcal{O}$, where $\mathsf{H} = \{\mathsf{Isa}, \mathsf{Tot}, \mathsf{Ex}\}$, – binary relation representing the *subtyping hierarchy* on object types.
- $\mathsf{O\text{-}CARD} \subseteq \mathcal{O} \times (\mathbb{N} \times (\mathbb{N} \cup \{\infty\}))$ – *partial* function defining the *cardinality* of the object type extensions (i.e. each population of A includes a number of instances that is constrained to be in a given range).

- R-CARD $\subseteq \mathcal{R} \times (\mathbb{N} \times (\mathbb{N} \cup \{\infty\}))$ – *partial* function defining the *cardinality* of the relations extensions (i.e. each population of R includes a number of instances that is constrained to be in a given range).

- OBJ $\subseteq \mathcal{R} \times \mathcal{O}$ – binary relation defining *objectifications*.

- $\mathsf{RING_J} \subseteq \wp(\varrho \times \varrho)$, where $\mathsf{J} = \{\mathsf{Irr}, \mathsf{Asym}, \mathsf{Trans}, \mathsf{Intr}, \mathsf{Antisym}, \mathsf{Acyclic}, \mathsf{Sym}, \mathsf{Ref}\}$, – relation defining *ring constraints*.

- V-VAL: $\mathcal{V} \rightarrow \wp(\Lambda_D)$, for some $\Lambda_D \in \Lambda$, – relation representing *object type value constraints*.

The glossary depicting the exact correspondence between key terms and symbols of an ORM2 diagram and their formal syntactical representation is given in Appendix A.

In order to provide means for expressing the modalities of the constraints expressed in ORM2 Formal Syntax, we introduce special superscripts $(\cdot)^{\Box}$, $(\cdot)^{\Diamond}$, $(\cdot)^{\boldsymbol{O}}$, and $(\cdot)^{\boldsymbol{P}}$, which are applicable to all the relations in Ω (e.g. $\mathsf{OBJ}^{\Box}$) and correspond to modal operators of necessity, possibility, obligation and permission respectively. If no superscript is provided, then a necessity superscript $(\cdot)^{\Box}$ is considered implicitly.

4.2.2 Input format

We now define a machine-readable format of ORM2 Formal Syntax, i.e. the plain-text representation of all the elements of the signature Ω and the ORM2 conceptual schema Σ.

Formal Syntax	Input Format
A set $\mathcal{E}$ of *entity type* symbols	`ENTITYTYPES: {EntType [,EntType]*}` *Ex.*: `ENTITYTYPES:{SeatBeltTest,BusinessObject}`
A set $\mathcal{V}$ of *value type* symbols	`VALUETYPES: {ValType [,ValType]*}` *Ex.*: `VALUETYPES:{MethodName,Quality,Time}`
Entity type or *value type* symbol	`UppercaseWord`, can contain letters, numbers and the following symbols: '_' and '-' *NB: entity and value types should have unique names.*
A set $\mathcal{R}$ of *relation* symbols	`RELATIONS: {Rel [,Rel]*}` *Ex.*: `RELATIONS:{HasCost,HasQuality}`
Relation symbol	`anycaseWord`, can contain letters, numbers and the following symbols: '_' and '-' *NB: relations should have unique names (see below).*

A set $\mathcal{A}$ of (attribute) *role* symbols	`ATTRIBUTES: {attr [,attr]*}`
A set $\mathcal{D}$ of *domain* symbols	`DOMAIN: {DomainElem [,DomainElem]*}`
Role or *domain* symbol	`anycaseWord`, can contain letters, numbers and the following symbols: '_' and '-' *NB: sets of role names, domain symbols, relation names, entity and value type names should be disjoint.*
A binary relation $\varrho \subseteq \mathcal{R} \times \mathcal{A}$	`LOC-ROLES: {Rel.attr [,Rel.attr]*}` *Ex.*: `LOC-ROLES:{TestedBy.function,TestedBy.method}`
A bijection $\tau_R \colon \varrho_R \to [1..\lvert\varrho_R\rvert]$	•`LOC-ROLES-INDEX(Rel.attr,i)` •`LOC-ROLES-INDEX: {(Rel.attr,i) [,(Rel.attr,i)]*}` *Ex.*: `LOC-ROLES-INDEX(HasQuality.quality,2)`
$\mathsf{TYPE} \subseteq \varrho \times \mathcal{O}$	`TYPE(Rel.attr,ObjType)` *Ex.*: `TYPE(HasCost.method,Method)`
$\mathsf{FREQ} \subseteq \wp(\varrho) \times (\mathbb{N} \times (\mathbb{N} \cup \{\infty\}))$	`FREQ({Rel.attr [,Rel.attr]*},(i,j))`, where `i`≥ 0 and `j`> 0 or `j=inf` or `j=INF`, corresponding to `j`$= +\infty$ *Ex.*: `FREQ({HasQuality.method},(1,1))`
$\mathsf{MAND} \subseteq \wp(\varrho) \times \mathcal{O}$	`MAND({Rel.attr [,Rel.attr]*},ObjType)` *Ex.*: `MAND({IsIdentifByMethodName.method},Method)`
$\mathsf{R\text{-}SET_H} \subseteq (\wp(\varrho) \times \wp(\varrho)) \times (\mu \colon \varrho \to \varrho)$	`R-SET`*h*`({Rel.attr [,Rel.attr]*}, {Rel.attr [,Rel.attr]*}, {(Rel.attr,Rel.attr) [,(Rel.attr,Rel.attr)]*})`, where *h*=`[sub|exc]` *Ex.*: `R-SETsub({isIn.course,isIn.subj}, {activIn.course,activIn.subj}, {(isIn.course,activIn.course), (isIn.subj,activIn.subj)})`
$\mathsf{O\text{-}SET_H} \subseteq 2^{\mathcal{O}} \times \mathcal{O}$, where $\mathsf{H} = \{\mathsf{Isa}, \mathsf{Tot}, \mathsf{Ex}\}$	`O-SET`*h*`({ObjType [,ObjType]*},ObjType)`, where *h*=`[isa|tot|ex]` *Ex.*: `O-SETtot({PhysMethod,VirtMethod},Method)`
$\mathsf{O\text{-}CARD} \subseteq \mathcal{O} \times (\mathbb{N} \times (\mathbb{N} \cup \{\infty\}))$	`O-CARD(ObjType)=(i,j)`, where `i`≥ 0 and `j`> 0 or `j=inf` or `j=INF`, corresponding to `j`$= +\infty$ *Ex.*: `O-CARD(ActiveCourse)=(1,50)`

<table>
<tr><td>R-CARD $\subseteq \mathcal{R} \times (\mathbb{N} \times (\mathbb{N} \cup \{\infty\}))$</td><td>R-CARD(Rel)=(i,j), where i$\geq$ 0 and j$>$ 0 or j=inf or j=INF, corresponding to j$= +\infty$
<u>Ex.</u>: R-CARD(HasQuality)=(1,INF)</td></tr>
<tr><td>OBJ $\subseteq \mathcal{R} \times \mathcal{O}$</td><td>OBJ(Rel,ObjType)
<u>Ex.</u>: OBJ(hasPrerequisite,O-Prerequisite)</td></tr>
<tr><td>RING$_J \subseteq \wp(\varrho \times \varrho)$, where J $=$ {Irr, Asym, Trans, Intr, Antisym, Acyclic, Sym, Ref}</td><td>RINGj(Rel.attr,Rel.attr),
where h=[irr|asym|trans|intr|antisym|acyclic|sym|ref]
<u>Ex.</u>: RINGirr(hasPrereq.subject,hasPrereq.object)</td></tr>
<tr><td>V-VAL: $\mathcal{V} \rightarrow \wp(\Lambda_D)$ for some $\Lambda_D \in \Lambda$</td><td>• V-VAL(valType)={(value..value)}
• V-VAL(valType)={(..value]}
• V-VAL(valType)={(value..)}
• V-VAL(valType)={value..value}
• V-VAL(valType)={..value}
• V-VAL(valType)={value..}
• V-VAL(valType)={value [,value]*}
• V-VAL(valType)={<xsd:datatype>}
where
• value can be either an integer or float(e.g. 1.3) number or a string constant escaped by '''' or '',
• range constraints are defined only for numbers,
• range constraints can either include ('[', ']') or exclude ('(', ')') bounds,
• unlimited range (e.g. (value..)) cannot include unlimited bound,
• when not explicitly stated bounds are included,
• xsd:datatype is any built-in XML Schema Datatype, e.g "xsd:decimal"

<u>Ex.</u>: V-VAL(Course-Code)={[101..399]}
<u>Ex.</u>: V-VAL(Student-Nr)={<xs:decimal>}</td></tr>
<tr><td>$(\cdot)^{\Box}$, $(\cdot)^{\Diamond}$, $(\cdot)^{\boldsymbol{O}}$, and $(\cdot)^{\boldsymbol{P}}$</td><td>\BOX{·}, \DIA{·}, \OB{·} and \PM{·} respectively.</td></tr>
</table>

4.2.3 Naming convention

According to the specification of ORM2 [21] it is not possible to have several relations with the same name, therefore it is obligatory that all distinct relations of the input schema have unique case sensitive names. Moreover, if a relation does not have a label (which is quite often the case for ORM2 diagrams), it is the duty of a tool which converts to ORM2 Formal Syntax to assign such a label.

Each role within its relation should have a unique label, i.e. it's forbidden that a relation has two attributes with the same name. In case of a relation having the same labels of its roles (e.g., `friendOf / friendOf`), new labels have to be assigned. However, it is allowed that two different relations share the labels for their attributes (e.g. `hasCar.owner` and `hasHouse.owner`).

In the case of dealing with reference schemas (e.g., `.name`, `.Nr`) the person or tool that produces the ORM2 input file should identify those reference predicates (relations) by either surrogate (e.g., `P2`, `P3`) or extended names (e.g., `PersonHasName`, `IsIdentifByCarName`), as recommended by [21]. We suggest several policies to be used in order to provide unique names for reference predicates:

- Substitute the dot '`.`' with a dash and a typical "`has`" and add the name of the relevant concept in front of the dash, e.g. `Person-hasNumber`.
- Take a typical key-phrase "`IsIdentifiedBy`" and add the name of the relevant range value type, e.g. `IsIdentifiedByPersonName`.

It is obligatory that the sets of relation names, value type names, entity type names, attribute names, domain symbols are mutually disjoint and define unique names for each the aforementioned element.

4.2.4 Obligatory input elements

Although the tool is capable of filling the ORM2 signature Ω on-the-fly, while parsing ORM2 conceptual schema Σ constraints, it is strongly recommended that the input file contains an explicit full specification of the ORM2 signature.
Moreover, the following elements of the signature Ω are considered to be obligatory:

- the bijection `LOC-ROLES-INDEX` defining the indices of roles in a relation. This is necessary in order to define the direction and the arity of the relation, which is especially crucial in cases of so-called "looping" relations, where relation connects several objects of the same object type (e.g. `childOf`).

- the set `VALUETYPES` of *value type* symbols. This requirement is imposed by the fact that in some cases it is not possible to determine whether the parameter of the constraint is a value type or an entity type (e.g. in case of a typing constraint). Moreover, for implementation reasons, it is necessary to link all the value types to built-in OWL2 datatypes, which can be done in one of the following ways:

- if for some value type there is a `V-VAL` range or enumeration constraint, then the base OWL2 datatype is inferred by the type of the constants used in this constraint.

- the base datatype for some value type can be defined explicitly, using a value constraint of the form `V-VAL(valType)={<xsd:datatype>}`.

- if for some value type there exists no value constraint, then its base datatype is taken to be *xsd:string*.

4.3 Logical foundations of implementation

The algorithm of the developed automated reasoning support tool relies on two fundamental results.

Firstly, it implements the procedure defined in [15] to translate a set of constraints from ORM2 Formal Syntax to $\mathcal{ALCQI}$ description logic, which is in fact a fragment of OWL2. The mapping procedure from ORM2 Formal Syntax to $\mathcal{DLR}$ is given in Appendix B and the encoding of the resulting $\mathcal{DLR}$ knowledge base into $\mathcal{ALCQI}/$(OWL2) is done in accordance to [10]. Since $\mathcal{ALCQI}$ is less expressible than $\mathcal{DLR}$, in particular, it doesn't support n-ary relations, the following information about the ORM2 conceptual schema might be lost:

- no frequency constraints on multiple roles

- no generalized subset (R-SET) constraints

  - *supported:* simple case of stand-alone roles
  - *supported:* special case of contiguous full-set of roles

- no ring constraints *(NB: drawback of mapping n-ary relations via reification)*

Secondly, in order to check the consistency of a set of business rules expressed in $\mathcal{ALCQI}$-definable fragment of ORM2, we utilize the modularity of the approach defined in Section 3.3 and adapt the result of satisfiability reduction for the case of general description logic DL. The satisfiability relation for ORM2 is then provided by the semantic-preserved translation from ORM2 Formal Syntax to $\mathcal{ALCQI}$ [15].

Theorem 4.3.1.
A FODAL regulation $\Sigma = \{\Box\eta_1, ..., \Box\eta_k, \Diamond\pi_1, ..., \Diamond\pi_l, \boldsymbol{O}\theta_1, ..., \boldsymbol{O}\theta_m, \boldsymbol{P}\rho_1, ..., \boldsymbol{P}\rho_n\}$*, expressed in* DL*-definable fragment of* ORM2*, is internally consistent if and only if each of the following description logic formulae* $\mathcal{N}^{DL}, \mathcal{O}^{DL}, \mathcal{Q}_j^{DL}, \mathcal{P}_j^{DL}$ *is independently satisfiable in* DL*:*

$$\begin{aligned}
\mathcal{N}^{DL} &= \prod_{i=1}^{k} \eta_i \\
\mathcal{O}^{DL} &= \prod_{i=1}^{m} \theta_i \sqcap \prod_{i=1}^{k} \eta_i \\
\mathcal{Q}_j^{DL} &= \pi_j \sqcap \prod_{i=1}^{k} \eta_i, \ \ \forall j = \overrightarrow{1 \dots l} \\
\mathcal{P}_j^{DL} &= \rho_j \sqcap \prod_{i=1}^{m} \theta_i \sqcap \prod_{i=1}^{k} \eta_i, \ \ \forall j = \overrightarrow{1 \dots n}
\end{aligned} \tag{4.1}$$

Thus, we can reduce the consistency of a given set of constraints to $\mathcal{ALCQI}$ satisfiability, which in turn can be interpreted as unsatisfiable concepts' check in resulting OWL2 ontology. Indeed, whenever a formula in $\mathcal{ALCQI}$ is unsatisfiable, it means that the concept definition expressed by this formula contains a contradiction which prevents the concept from having a model, i.e. the concept is forced to *not* have any instances, hence is unsatisfiable.

Therefore, in order to check the consistency of an ORM2 schema Σ expressed in ORM2 Formal Syntax extended with four basic modal operators, we should perform the following steps:

1. Translate each constraint $r \in \Sigma$, omitting its preceding modal operator, into $\mathcal{ALCQI}$ formula ϕ_r^{ALCQI} using the mapping procedure from ORM2 Formal Syntax to $\mathcal{DLR}$ given in Appendix B and the following translation into $\mathcal{ALCQI}$ according to [10].
   N.B.: In the case when a constraint cannot be expressed in $\mathcal{ALCQI}$, a corresponding error message should be generated.

2. Process the resulting formulae according to Section 3.3 obtaining the set of objective formulae $\mathcal{N}^{DL}, \mathcal{O}^{DL}, \mathcal{Q}_j^{DL}, \mathcal{P}_j^{DL}$ from Theorem 4.3.1, each of those is then mapped into a corresponding OWL2 ontology $\mathfrak{O}_i$.

3. For each ontology $\mathfrak{O}_i$ perform the unsatisfiable classes check using one of the supported OWL2 reasoners (e.g. FaCT++ or HermiT). Then Σ is internally consistent if and only if none of $\mathfrak{O}_i$ contains unsatisfiable classes.

Taking advantage of Definition 4.1 it is also becomes possible to define whether the inconsistency is alethic, deontic or of a cross type. Furthermore, in certain cases when inconsistencies are caused by rules of possibility or permission it would be possible to define the source of inconsistency.

4.4 Examples

This section demonstrates the functionality of the implemented tool by means of examples of its usage for different tasks. The graphical user interface of a tool is introduced on Figure 4.2.

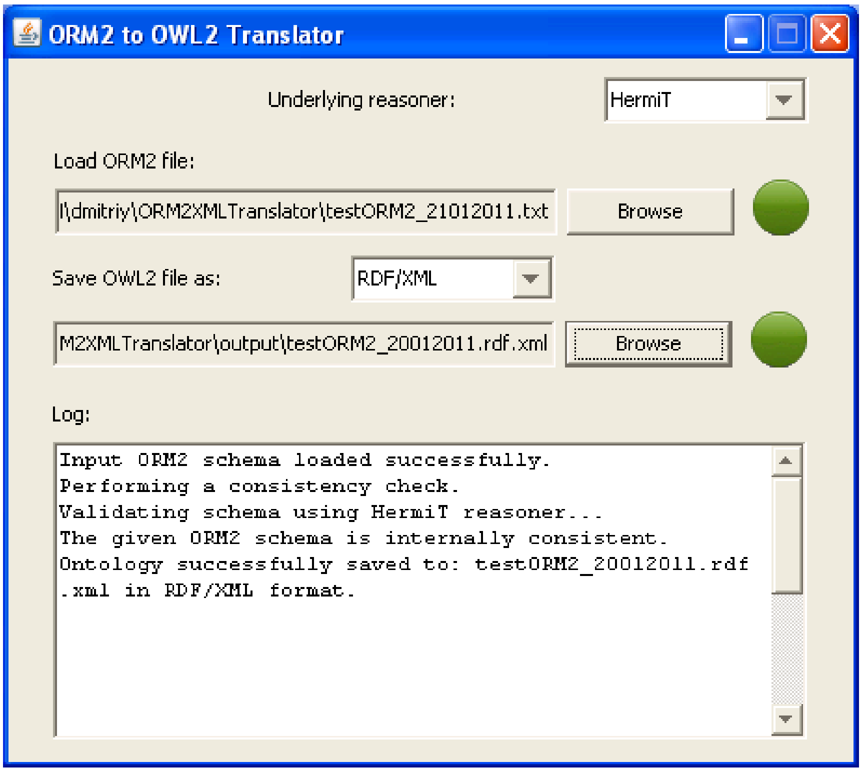


Figure 4.2: The Graphical User Interface

4.4.1 Checking the consistency of a given set of rules

The functionality of the consistency check will be demonstrated on the classical example of an inconsistent ORM2 schema from [21, p.295] (see Figure 4.3).

Below is an $\mathcal{ALCQI}$-expressible fragment of this schema in ORM2 Formal Syntax without any modal operators, therefore necessities are implicitly understood:

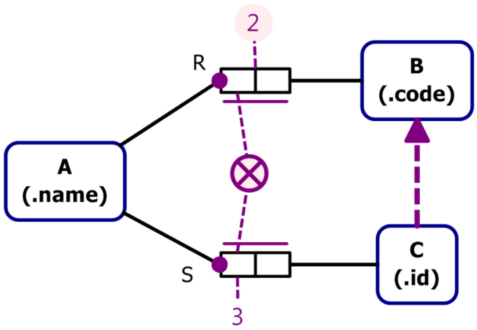


Figure 4.3: Inconsistent ORM2 Schema

```
ENTITYTYPES: {A,B,C}
RELATIONS: {R,S}
TYPE(R.a,A)
TYPE(R.b,B)
TYPE(S.a,A)
TYPE(S.c,C)
LOC-ROLES-INDEX: {(R.a,1),(R.b,2),(S.a,1),(S.c,2)}
MAND({R.a},A)
MAND({S.a},A)
FREQ({S.a},(1,3))
O-SETisa({C},B)
R-SETexc({R.a},{S.a},{(R.a,S.a)})
```

This schema is internally inconsistent since the mandatory roles on A imply an equality constraint between these role, therefore if A is populated, then the exclusion set constraint cannot be satisfied. The same result is indeed demonstrated by the implemented tool on Figure 4.4.

Let us now express the exclusion constraint as an obligation, so that still conflicts with the mandatory constraints, but in a deontic manner:

```
\OB{ R-SETexc({R.a},{S.a},{(R.a,S.a)}) }
```

We then get the following message as a result of a consistency check:

```
The given ORM2 schema is internally inconsistent w.r.t. obligations since its
OWL2 translation contains unsatisfiable concepts.
The following concepts are unsatisfiable:
  obj_S
  obj_R
  A
```

Let us now introduce the modal operators in the following way:

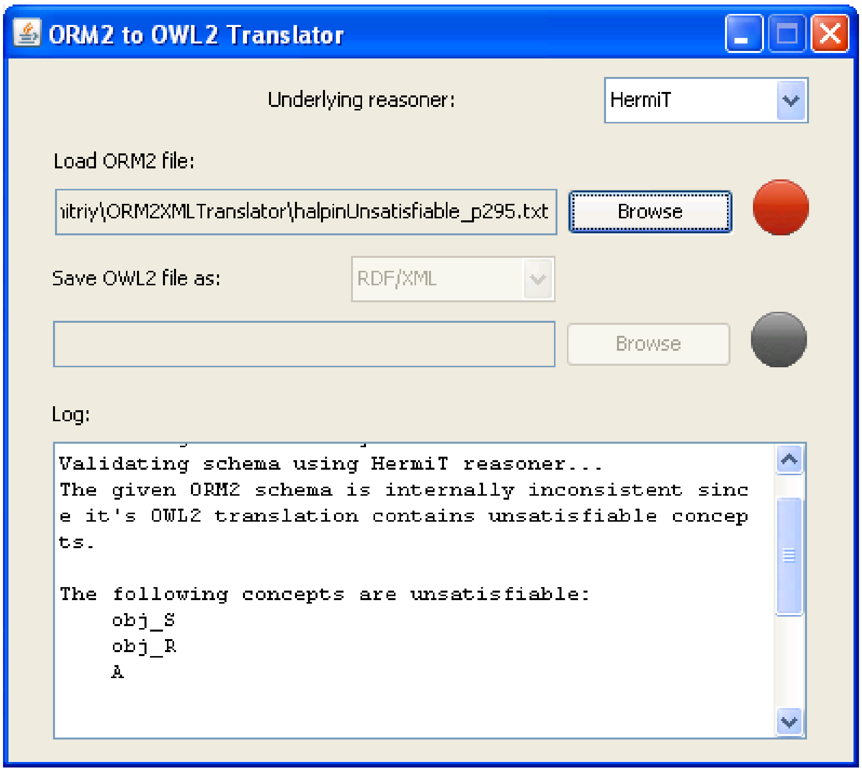


Figure 4.4: Sample output

```
\OB{ MAND({R.a},A) }
\OB{ MAND({S.a},A)
FREQ({S.a},(1,3))
O-SETisa({C},B)
\DIA{ R-SETexc({R.a},{S.a},{(R.a,S.a)}) }
```

We then get the following message as a result of a consistency check:

```
The given ORM2 schema is internally consistent.
```

This is definitely the case since none of the constraints contradict with each other.

4.4.2 Translating a given ORM2 schema into OWL2 ontology

As a part of implementation the translation from a given ORM2 schema into OWL2 ontology was implemented. However, this translation does not support modal operators in their diversity and therefore ignores all rules except necessities.

A sample ORM2 schema is presented on Figure 4.5. We present below only a fragment of

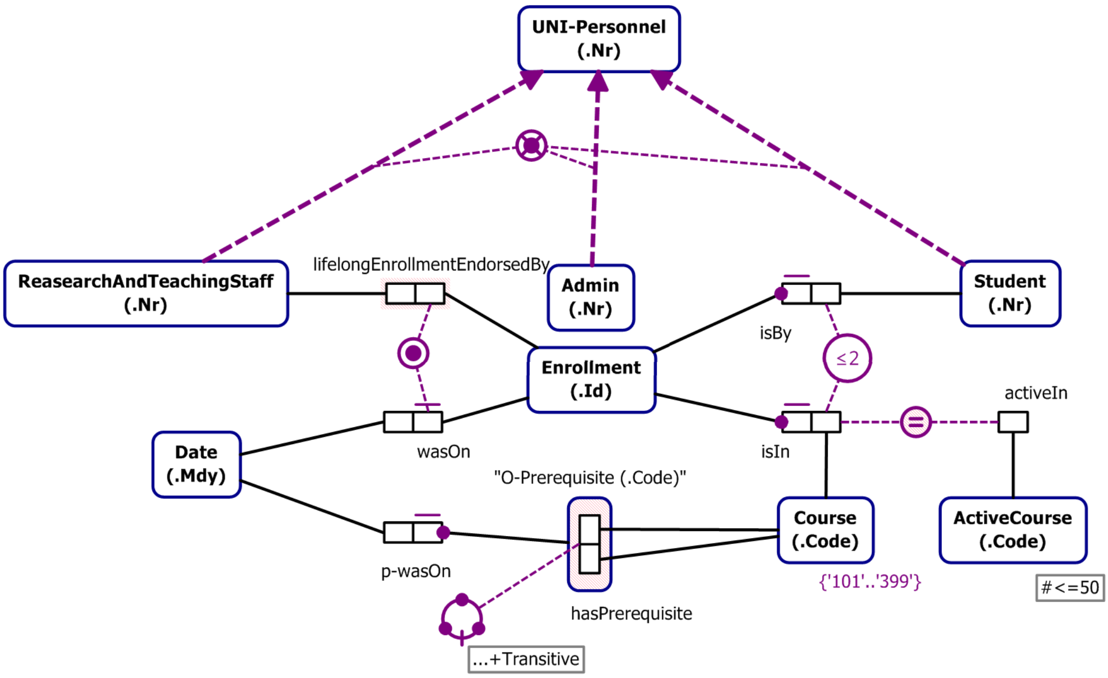


Figure 4.5: Sample ORM2 input schema

an input file in ORM2 Formal Syntax with some key constraints. The OWL/XML fragment of the OWL2 output obtained from this particular example is introduced in Appendix C.

```
OBJ(hasPrerequisite,O-Prerequisite)
MAND({lifelongEnrollmentEdorsedBy.object,wasOn.enrollment},Enrollment)
O-SETtot({ReseachOrTeachingStaff,Student,Admin},UNI-Personnel)
R-SETsub({isIn.course},{activeIn.activecourse},
         {(isIn.course,activeIn.activecourse)})
RINGtrans(hasPrerequisite.subject,hasPrerequisite.object)
O-CARD(ActiveCourse)=(1,50)
R-CARD(activeIn)=(1,INF)
V-VAL(Admin-Nr)={<xsd:decimal>}
V-VAL(Course-Code)={[101..399]}
```

CHAPTER 5

Conclusion and future work

The research conducted in this thesis was dedicated to the problem of logical formalization of the Semantics of Business Vocabulary and Rules standard (SBVR) and providing the business expert with automated solutions with reasoning support based on such logical formalization.

Firstly, we investigated in detail the formal semantics of SBVR defined by the standard itself and pointed out several shortcomings of the adopted approach. The most significant drawbacks include, but are not limited to, disregarding the possible interaction between different types of rules, incompatibility with the formal semantics of classical deontic logics as well as ambiguity of the model-theoretic semantics. These facts of imperfection of the existing formalization justified the demands for a well-defined, noncontradictory formal semantics for business vocabularies and rules.

As a result of our research, motivated by the drawbacks of the existing methods, we defined a first-order deontic-alethic logic (FODAL) along with its syntax, semantics and complete and sound axiomatization. We also defined a certain class **FODAL** of bimodal frames and proved that such frames correctly capture the desired semantics of alethic and deontic rules as well as their interactions.

Another fundamental result obtained in this thesis is that we demonstrated that restricting the domain of interest to business rules expressed as atomic modal sentences allows to reduce the problem of satisfiability in FODAL to satisfiability in first-order logic and, further restricting rule formulations, to satisfiability in description logic. Also in the attempt to establish a relationship with a standard logical formalism, we defined a truth-preserving translation from atomic modal sentences of bimodal FODAL into quantified monomodal logic $\boldsymbol{QK}$, that can be used to facilitate the transfer of decidability results from well-studied fragments of predicate modal logics to FODAL.

The implementation part of the thesis has resulted in a special tool which provides an automated support for consistency checks of the conceptual model along with its translation to OWL2 ontology. A new plain-text input format was also developed for a newly defined Formal Syntax of ORM2 schemas – a graphical notation for SBVR.

The main functionality of the developed tool results in support for consistency checks of a set of $\mathcal{ALCQI}$-expressible deontic and alethic business rules. Another important task which is supported by the tool is translation of the aforementioned fragment of an ORM2 schema into an OWL2 ontology. However, this translation, as opposed to consistency check, does not support any modalities except necessity due to lack of notions representing deontic rules in OWL2 (that however exist in some extensions of OWL2, e.g. SWRL). This tool was developed in the context of the ONTORULE FP7 project [3] and became a part of the public deliverable [26].

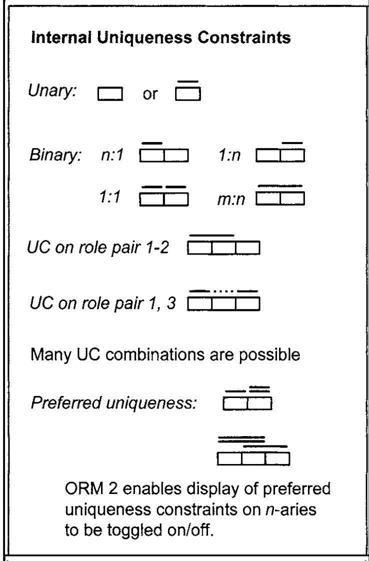
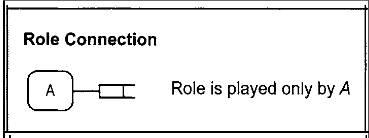
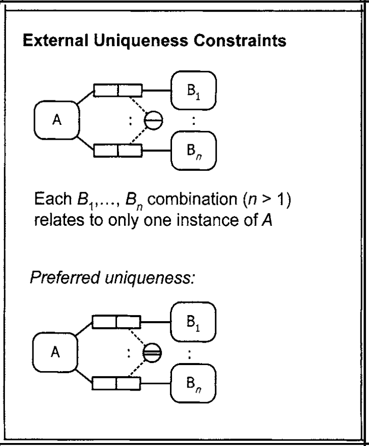
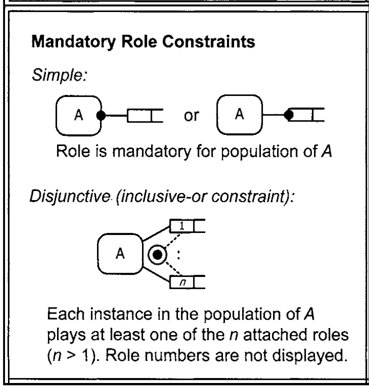
The future research in the field of logical formalization of SBVR aims to study the problem of entailment with respect to possible interaction of alethic and deontic modalities. Another possible future course of work may include defining an approach to translate an ORM2 schema with its alethic and deontic rules to SWRL or some other extension of OWL2.

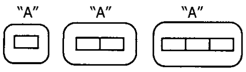
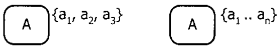
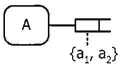
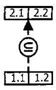
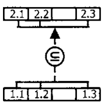
There is also an ongoing work on binding the implemented approach together with NORMA (an open-source ORM2 tool) which could lead into more extensive integration with other modeling tools and fulfil the demand for automated solutions with reasoning support.

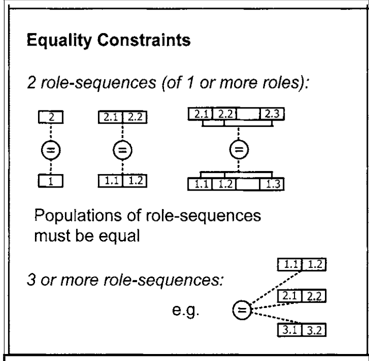
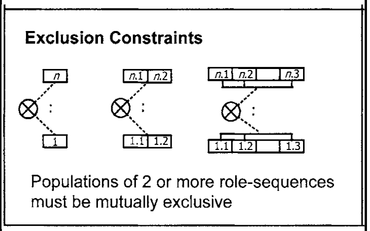
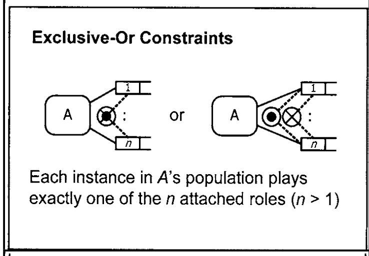
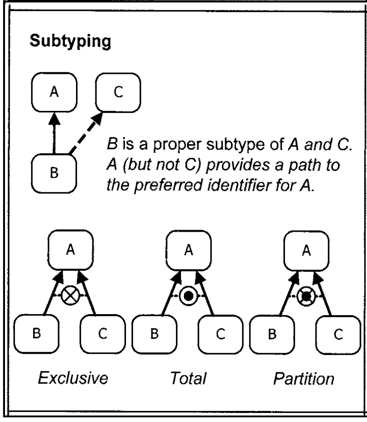
APPENDIX A

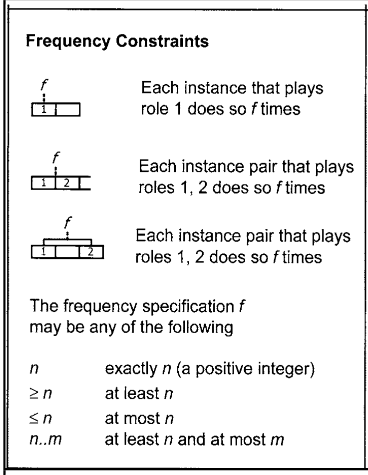
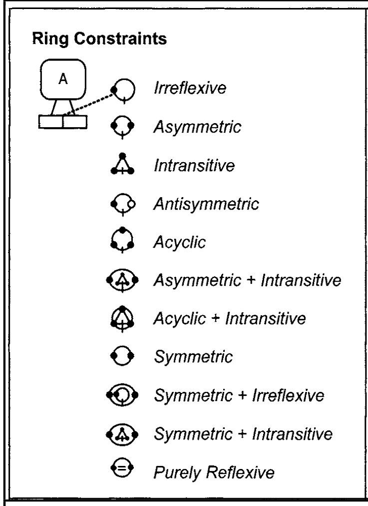
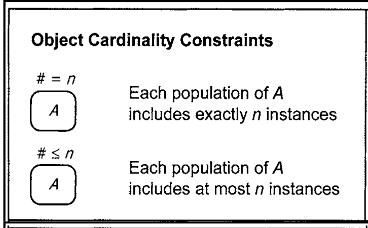
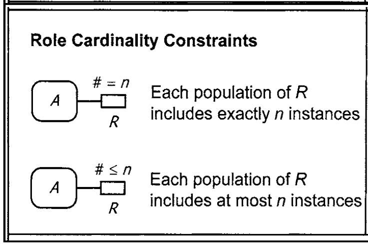
Glossary of the ORM2 Formal Syntax

The glossary depicts the exact correspondence between key terms and symbols of an ORM2 diagram (on the left) and their formal syntactical representation (on the right) as given in [15].

ORM2	
	$FREQ(\{R.a_1,\ldots,R.a_n\},(0,1))$
	$TYPE(R.a,A)$
	$FREQ(\{R.a,S.b\},(0,1))$ $FREQ(\{R^1.a^1_1,\ldots,R^1.a^1_n,\ldots,R^k.a^k_1,\ldots,R^k.a^k_m\},(0,1))$
	$MAND(\{R.a\},A)$ $MAND(\{R^1.a^1_1,\ldots,R^1.a^1_n,\ldots,R^k.a^k_1,\ldots,R^k.a^k_m\},A)$

ORM2	
Objectification  etc. Fact type is objectified as object type *A*. ORM 2 allows any fact type to be objectified.	$\mathrm{OBJ}(R,A)$
Object Value Constraints *Enumeration* *Range*  *Semibounded discrete range* { a.. } { ..a } *Bounded continuous range* $\{[a_1 .. a_2]\}$ *includes both end values* $\{(a_1 .. a_2)\}$ *excludes both end values* $\{[a_1 .. a_2)\}$ *includes first value* $\{(a_1 .. a_2]\}$ *includes last value* Combinations are allowed.	$\mathrm{V\text{-}VAL}(A)=\{v^D{}_1,\ldots,v^D{}_n\}$
Role Value Constraints  Same patterns as above	DERIVED [TYPE + V-VAL + MAND + UNIQ]
Subset Constraints *Simple:*  Each object that plays role 1 also plays role 2 *Contiguous Role-pair:*  Each object pair that plays the role sequence 1.1, 1.2 also plays the role sequence 2.1, 2.2 *Other cases:*  Each object tuple that plays the role sequence 1.1, 1.2, 1.3 also plays the role sequence 2.1, 2.2, 2.3 ORM 2 also displays subset constraints over join paths	$\mathrm{R\text{-}SET}_{\mathrm{Sub}}(\{R.a\},\{S.b\},\mu)$ $\mathrm{R\text{-}SET}_{\mathrm{Sub}}(\{R^1.a^1{}_1,\ldots,R^1.a^1{}_n,\ldots,R^k.a^k{}_1,\ldots,R^k.a^k{}_m\},$ $\{S^1.b^1{}_1,\ldots,S^1.b^1{}_n,\ldots,S^k.b^k{}_1,\ldots,S^k.b^k{}_m\},\mu)$

ORM2	
	DERIVED [$\text{R-SET}_{\text{Sub}}$]
	$\text{R-SET}_{\text{Exc}}(\{R.a\},\{S.b\},\mu)$ $\text{R-SET}_{\text{Exc}}(\{R^1.a^1{}_1,\ldots,R^1.a^1{}_n,\ldots,R^k.a^k{}_1,\ldots,R^k.a^k{}_m\},$ $\{S^1.b^1{}_1,\ldots,S^1.b^1{}_n,\ldots,S^k.b^k{}_1,\ldots,S^k.b^k{}_m\},\mu)$
	DERIVED [MAND + $\text{R-SET}_{\text{Exc}}$]
	$\text{O-SET}_{\text{Isa}}(\{O_1,\ldots,O_n\},A)$ $\text{O-SET}_{\text{Tot}}(\{O_1,\ldots,O_n\},A)$ $\text{O-SET}_{\text{Ex}}(\{O_1,\ldots,O_n\},A)$

ORM2	
 **Frequency Constraints** Each instance that plays role 1 does so *f* times Each instance pair that plays roles 1, 2 does so *f* times Each instance pair that plays roles 1, 2 does so *f* times The frequency specification *f* may be any of the following *n* exactly *n* (a positive integer) ≥ *n* at least *n* ≤ *n* at most *n* *n..m* at least *n* and at most *m*	FREQ($\{R.a\}$,(min,max)) FREQ($\{R^1.a^1{}_1,\ldots,R^1.a^1{}_n\}$,(min,max)) [+ 'External Frequency Constraint': FREQ($\{R^1.a^1{}_1,\ldots,R^1.a^1{}_n,\ldots,R^k.a^k{}_1,\ldots,R^k.a^k{}_m\}$,(min,max))]
 **Ring Constraints** *Irreflexive* *Asymmetric* *Intransitive* *Antisymmetric* *Acyclic* *Asymmetric + Intransitive* *Acyclic + Intransitive* *Symmetric* *Symmetric + Irreflexive* *Symmetric + Intransitive* *Purely Reflexive*	RING$_J$($R.a,R.b$) where J={Irr,Asym,Trans,Intr,Antisym,Acyclic,Sym,Ref}
 **Object Cardinality Constraints** Each population of *A* includes exactly *n* instances Each population of *A* includes at most *n* instances	O-CARD(A)=(min,max)
 **Role Cardinality Constraints** Each population of *R* includes exactly *n* instances Each population of *R* includes at most *n* instances	R-CARD(R)=(min,max)

APPENDIX B

Mapping ORM2 Formal Syntax into $\mathcal{DLR}$

In this appendix we present the mapping of the ORM2 Formal Syntax into into $\mathcal{DLR}$ description logic, which was developed in [15] with a humble contribution of the author of the thesis.

$\mathcal{DLR}$ concepts and relations (of arity between 2 and n_{max}) are built according to the following syntax[1]:

$$
\begin{array}{lllllllllll}
R & ::= & \top_n & \mid & P & \mid & (\$i/n : C) & \mid & \neg R & \mid & R_1 \sqcap R_2 \\
C & ::= & \top_1 & \mid & A & \mid & \neg C & \mid & C_1 \sqcap C_2 & \mid & \\
 & & \exists[\$i]R & \mid & (\leq k[\$i]R) & & & & & &
\end{array}
$$

where P and A denote *atomic relations* and *atomic concepts* respectively, R and C denote arbitrary *relations* and *concepts*, i denotes components of relations (i.e. an integer between 1 and $n_m ax$), k denotes nonnegative integer, $\top_1$ denotes the top concept, $\top_n$, for $n = 2, \ldots, n_{max}$, denotes the top relation of arity n.

[1]Concepts and relations in $\mathcal{DLR}$ must be *well-typed*.

- Background domain axioms:

$$E \sqsubseteq \Delta \tag{B.1}$$

$$V \sqsubseteq \Lambda_D, \text{ where } \Lambda_D = \{v_1^D, \ldots, v_n^D\}, \text{ for some } D \tag{B.2}$$

$$\top \sqsubseteq \Delta \cup \Lambda_{D_1} \sqcup \cdots \sqcup \Lambda_{D_n} \tag{B.3}$$

$$\Delta \sqsubseteq \neg(\Lambda_{D_1} \sqcup \cdots \sqcup \Lambda_{D_n}) \tag{B.4}$$

$$\Lambda_{D_1} \sqsubseteq \Lambda_{D_2} \tag{B.5}$$

$$\vdots$$

$$\Lambda_{D_{n-1}} \sqsubseteq \Lambda_{D_n}$$

- $\mathsf{TYPE}(R.a, O)$

$$\exists[\$a]R \sqsubseteq O$$

- $\mathsf{FREQ}(R.a, (\mathsf{min}, \mathsf{max}))$

$$\exists[\$a]R \quad \sqsubseteq \quad \geq \mathsf{min}[\$a]R \sqcap \leq \mathsf{max}[\$a]R$$

- $\mathsf{MAND}(R_1, \ldots, R_k, O)$

$$\begin{aligned} O \quad \sqsubseteq \quad & \exists[\$a_1^1]R_1 \sqcup \cdots \sqcup \exists[\$a_n^1]R_1 \sqcup \cdots \sqcup \\ & \exists[\$a_1^k]R_k \sqcup \cdots \sqcup \exists[\$a_n^k]R_k \end{aligned}$$

- $\mathsf{R\text{-}SET}_{\mathsf{H}}(A, B)$
  where $A = \{R.a_1, \ldots, R.a_n\}$, $B = \{S.b_1, \ldots, S.b_n\}$, and $n = |\sigma_A| = |\sigma_B|$:

  $\mathsf{R\text{-}SET}_{\mathsf{Sub}}(A, B)$

$$A \sqsubseteq B$$

  $\mathsf{R\text{-}SET}_{\mathsf{Exc}}(A, B)$

$$A \sqsubseteq \neg B$$

- $\mathsf{O\text{-}SET}_{\mathsf{Isa}}(P_1, \ldots, P_n, Q)$

$$P_1 \sqcup \cdots \sqcup P_n \sqsubseteq Q$$

- $\mathsf{O\text{-}SET}_{\mathsf{Tot}}(P_1, \ldots, P_n, Q)$

$$Q \equiv P_1 \sqcup \cdots \sqcup P_n$$

- O-SET$_{\mathsf{Ex}}(P_1, \ldots, P_n, Q)$

$$\begin{aligned} P_1 \sqcup \cdots \sqcup P_n &\sqsubseteq Q \\ P_1 &\sqsubseteq \neg P_2 \\ &\vdots \\ P_{n-1} &\sqsubseteq \neg P_n \end{aligned}$$

- V-VAL$(V, \{v_1^D, \ldots, v_n^D\})$

$$V \equiv \{v_1^D, \ldots, v_n^D\}$$

APPENDIX C

The OWL/XML fragment of a sample OWL2 output

Below is the OWL/XML fragment of the OWL2 output of the translation tool for the ORM2 schema depicted on Figure 4.5.

```
<?xml version="1.0"?>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
     ontologyIRI="http://www.ontorule.com/ontologies/testORM2_20012011.owl">
    <Annotation>
        <AnnotationProperty IRI="#TranslationNote_3"/>
        <Literal
            datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Missed one
            of R-SETj constraints: unable to translate! One of involved
            relations is &#39;lifelongEnrollmentEdorsedBy&#39;</Literal>
    </Annotation>
    <Declaration>
        <Class IRI="#Course"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Date"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Enrollment"/>
    </Declaration>
    <EquivalentClasses>
        <Class IRI="#UNI-Personnel"/>
        <ObjectUnionOf>
            <Class IRI="#Admin"/>
```

```
            <Class IRI="#ReseachOrTeachingStaff"/>
            <Class IRI="#Student"/>
        </ObjectUnionOf>
    </EquivalentClasses>
    <SubClassOf>
        <Class IRI="#ActiveCourse"/>
        <ObjectSomeValuesFrom>
            <ObjectInverseOf>
                <ObjectProperty IRI="#activecourse"/>
            </ObjectInverseOf>
            <Class IRI="#obj_ActCourse-hasCode"/>
        </ObjectSomeValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Admin"/>
        <ObjectIntersectionOf>
            <ObjectComplementOf>
                <Class IRI="#ReseachOrTeachingStaff"/>
            </ObjectComplementOf>
        </ObjectIntersectionOf>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Enrollment"/>
        <ObjectUnionOf>
            <ObjectSomeValuesFrom>
                <ObjectInverseOf>
                    <ObjectProperty IRI="#enrollment"/>
                </ObjectInverseOf>
                <Class IRI="#obj_wasOn"/>
            </ObjectSomeValuesFrom>
            <ObjectSomeValuesFrom>
                <ObjectInverseOf>
                    <ObjectProperty IRI="#object"/>
                </ObjectInverseOf>
                <Class IRI="#obj_lifelongEnrollmentEdorsedBy"/>
            </ObjectSomeValuesFrom>
        </ObjectUnionOf>
    </SubClassOf>
    <SubClassOf>
        <ObjectUnionOf>
            <Class IRI="#Admin"/>
            <Class IRI="#ReseachOrTeachingStaff"/>
            <Class IRI="#Student"/>
        </ObjectUnionOf>
        <Class IRI="#UNI-Personnel"/>
    </SubClassOf>
```

```
    <SubClassOf>
        <ObjectSomeValuesFrom>
            <ObjectInverseOf>
                <ObjectProperty IRI="#activecourse"/>
            </ObjectInverseOf>
            <Class IRI="#obj_ActCourse-hasCode"/>
        </ObjectSomeValuesFrom>
        <ObjectExactCardinality cardinality="1">
            <ObjectInverseOf>
                <ObjectProperty IRI="#activecourse"/>
            </ObjectInverseOf>
            <Class IRI="#obj_ActCourse-hasCode"/>
        </ObjectExactCardinality>
    </SubClassOf>
    <FunctionalObjectProperty>
        <ObjectProperty IRI="#admin"/>
    </FunctionalObjectProperty>
    <FunctionalObjectProperty>
        <ObjectProperty IRI="#course"/>
    </FunctionalObjectProperty>
    <DatatypeDefinition>
        <Datatype IRI="#ActiveCourse-Code"/>
        <Datatype abbreviatedIRI="xsd:string"/>
    </DatatypeDefinition>
    <DatatypeDefinition>
        <Datatype IRI="#Course-Code"/>
        <DataIntersectionOf>
            <DatatypeRestriction>
                <Datatype abbreviatedIRI="xsd:integer"/>
                <FacetRestriction
                    facet="http://www.w3.org/2001/XMLSchema#minInclusive">
                    <Literal
                        datatypeIRI="http://www.w3.org/2001/XMLSchema#integer">101</Literal>
                </FacetRestriction>
            </DatatypeRestriction>
            <DatatypeRestriction>
                <Datatype abbreviatedIRI="xsd:integer"/>
                <FacetRestriction
                    facet="http://www.w3.org/2001/XMLSchema#maxInclusive">
                    <Literal
                        datatypeIRI="http://www.w3.org/2001/XMLSchema#integer">399</Literal>
                </FacetRestriction>
            </DatatypeRestriction>
        </DataIntersectionOf>
    </DatatypeDefinition>
</Ontology>
```

Bibliography

[1] Neumont ORM Architect (NORMA) for Visual Studio. `http://www.objectrolemodeling.com/AboutORM/ORMTools/NORMA/tabid/87/Default.aspx`.

[2] PNA Group Discovery and Validation Assistant prototype. `http://ontorule.pna-group.nl/`.

[3] The ONTORULE Project (FP7/2009-2011 grant agreement 231875). `www.ontorule-project.eu`.

[4] Hajnal Andréka, Johan Van Benthem, and István Németi. Modal languages and bounded fragments of predicate logic, 1996.

[5] Donald E. Baisley, John Hall, and Donald Chapin. Semantic formulations in SBVR. In *Rule Languages for Interoperability*. W3C, 2005.

[6] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Number 53 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, UK, 2001.

[7] Peter Bollen. The orchestration of fact-orientation and SBVR. In Terry A. Halpin, John Krogstie, Selmin Nurcan, Erik Proper, Rainer Schmidt, Pnina Soffer, and Roland Ukor, editors, *BMMDS/EMMSAD*, volume 29 of *Lecture Notes in Business Information Processing*, pages 302–312. Springer, 2009.

[8] Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, 1997.

[9] John Bulles, Jean-Paul Koster, and Sjir Nijssen. D1.2 Transforms from SBVR to ontologies and rules. Public deliverable, The ONTORULE Project (FP7/2009-2011 grant agreement 231875), 2010.

[10] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. Reasoning in expressive description logics with fixpoints based on automata on infinite trees. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, IJCAI '99, pages 84–89, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[11] Paolo Ceravolo, Cristiano Fugazza, and Marcello Leida. Modeling semantics of business rules. In *Proceedings of the Inaugural IEEE International Conference On Digital Ecosystems and Technologies (IEEE-DEST)*, February 2007.

[12] Donald Chapin. SBVR: What is now possible and why? *Business Rules Journal*, 9(3), 2008.

[13] Melvin Fitting. Modal proof theory. In Frank Wolter Patrick Blackburn, J. F. A. K. van Benthem, editor, *Handbook of Modal Logic*, pages 86–135. Elsevier, 2007.

[14] Melvin Fitting and Richard L. Mendelsohn. *First-order modal logic*. Kluwer Academic Publishers, Norwell, MA, USA, 1999.

[15] Enrico Franconi, Alessandro Mosca, and Dmitry Solomakhin. ORM2: Syntax and semantics. Internal report, KRDB Research Centre for Knowledge and Data, 2011.

[16] Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. Attempto Controlled English for Knowledge Representation. In Cristina Baroglio, Piero A. Bonatti, Jan Małuszyński, Massimo Marchiori, Axel Polleres, and Sebastian Schaffert, editors, *Reasoning Web, Fourth International Summer School 2008*, number 5224 in Lecture Notes in Computer Science, pages 104–124. Springer, 2008.

[17] Stijn Goedertier and Jan Vanthienen. Rule-based business process modeling and execution. In *Proceedings of the IEEE EDOC Workshop on Vocabularies Ontologies and Rules for The Enterprise (VORTE 2005). CTIT Workshop Proceeding Series (ISSN 0929-0672*, pages 67–74, 2005.

[18] Terry Halpin. *A Logical Analysis of Information Systems: Static Aspects of the Data-oriented Perspective*. PhD thesis, Department of Computer Science, University of Queensland, 1989.

[19] Terry Halpin. Object-Role Modeling (ORM/NIAM). In *Handbook on Architectures of Information Systems*, pages 81–102. Springer-Verlag, 1998.

[20] Terry Halpin. ORM2 Graphical Notation. Technical Report ORM2-01, September 2005.

[21] Terry Halpin and Tony Morgan. *Information Modeling and Relational Databases*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition, 2008.

[22] Terry A. Halpin and Jan Pieter Wijbenga. FORML 2. In Ilia Bider, Terry A. Halpin, John Krogstie, Selmin Nurcan, Erik Proper, Rainer Schmidt, and Roland Ukor, editors, *BMMDS/EMMSAD*, volume 50 of *Lecture Notes in Business Information Processing*, pages 247–260. Springer, 2010.

[23] Rami Hodrob and Mustafa Jarrar. Mapping ORM into OWL 2. In *Proceedings of the 1st International Conference on Intelligent Semantic Web-Services and Applications*, ISWSA '10, pages 9:1–9:7, New York, NY, USA, 2010. ACM.

[24] G. E. Hughes and M. J. Cresswell. *A New Introduction To Modal Logic*. Routledge, 1996.

[25] C. Maria Keet. Mapping the Object-Role Modeling language ORM2 into Description Logic language DLRifd. *CoRR*, abs/cs/0702089, 2007.

[26] Roman Korf, Eva Kiss, Adil El Ghali, Hugues Citeau, Amina Chniti, Sylvain Viguie, Diego Berrueta, Jeroen Hoppenbrouwers, Alessandro Mosca, Adeline Nazarenko, and Francois Levy. ONTORULE M24 Show-Case Demonstrator. Public deliverable, The ONTORULE Project (FP7/2009-2011 grant agreement 231875), 2010.

[27] Saul A. Kripke. *Naming and Necessity*. Harvard University Press, 1980.

[28] Agi Kurucz. Combining modal logics. In Frank Wolter Patrick Blackburn, J. F. A. K. van Benthem, editor, *Handbook of Modal Logic*, pages 869–926. Elsevier, 2007.

[29] Paul McNamara. Deontic logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2010 edition, 2010.

[30] John-Jules Meyer and Frank Veltman. Intelligent agents and common sense reasoning. In Frank Wolter Patrick Blackburn, J. F. A. K. van Benthem, editor, *Handbook of Modal Logic*, pages 991–1030. Elsevier, 2007.

[31] Thi Dieu Thu Nguyen, Nhan Le Thanh, and Thi Anh Le Pham. Modeling ORM Schemas in Description Logics. In Geilson Loureiro and Ricky Curran, editors, *Proceedings...*, pages 533–541, São José dos Campos, 2007. ISPE International Conference on Concurrent Engineering, 14.(CE 2007), Instituto Nacional de Pesquisas Espaciais (INPE).

[32] W3C OWL Working Group. *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation, 27 October 2009. Available at `http://www.w3.org/TR/owl2-overview/`.

[33] Johan van Benthem Patrick Blackburn. Modal logic: A semantic perspective. In Frank Wolter Patrick Blackburn, J. F. A. K. van Benthem, editor, *Handbook of Modal Logic*, pages 1–84. Elsevier, 2007.

[34] Ronald G. Ross. *Principles of the Business Rule Approach*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.

[35] Gerhard Schurz. Combinations and completeness transfer for quantified modal logics. *Logic Jnl IGPL*, pages jzp085+, January 2010.

[36] Dmitry Solomakhin, Alessandro Mosca, Enrico Franconi, and Alberto Siena. On the logical foundations of business rules. In *Proceedings of 1st Workshop on Law Compliancy Issues in Organisational Systems and Strategies*, 2010.

[37] The Business Rules Group. Defining business rules. What are they really? Technical report, The Business Rules Group, 2001.

[38] The Object Management Group. Semantics of Business Vocabulary and Business Rules (SBVR). Formal specification, v1.0, 2008.

[39] Frank Wolter and Michael Zakharyaschev. Decidable fragments of first-order modal logics. *J. Symb. Log.*, 66(3):1415–1438, 2001.

[40] Gang Zhao, Yanbin Gao, and Robert Meersman. An ontology-based approach to business modelling. In *Proceedings of the International Conference of Knowledge Engineering and Decision Support (ICKEDS2004)*, pages 213–221, 2004.