Syntax and Semantics for Business Rules

Xiaofan Liu[1,2], Natasha Alechina[1], and Brian Logan[1]

[1] School of Computer Science, University of Nottingham, Nottingham, NG8 1BB, UK
[2] School of Computer and Communication, Hunan University, Hunan, 410081, P.R. China
{lxx,nza,bsl}@cs.nott.ac.uk

Abstract. Business processes are increasingly being expressed in declarative terms, as *business rules* which express obligations concerning conduct, action, practice or procedure regarding a particular business activity or sphere of operation. However, the leading standard for business rules, Semantics for Business Vocabulary and Business Rules (SBVR), leaves certain decisions concerning the precise syntax and semantics of business rules unspecified, for example, the scope and nesting of modalities and first order quantifiers, and the precise semantics of alethic and deontic modalities. In this paper, we propose a precise syntax and semantics for SBVR and present some complexity results for business rules with the proposed syntax and semantics.

1 Introduction

A *business rule* is a statement that defines or constrains some aspect of a business. Business rules express complex constraints on the business states and how the state of a business or business process should evolve. They talk about necessity and possibility, obligation and permission, and impose constraints on temporal sequences of states of affairs. As such they lie outside the scope of standard ontology languages.

Business processes are increasingly being specified in terms of such rules. In addition to capturing the 'knowledge of a company', the codification of business processes in the form of rules is key to many agile software development strategies. Rule-based approaches offer significant advantages to the application developer: their focus on the declarative representation of small, relatively independent, knowledge units makes it easier for developers and even end users to rapidly develop applications. In addition, the clear separation of application logic (represented as rules) from the bulk of the application code facilitates rapid modification in response to changing organizational goals (e.g., changes in the business process).

Business rules can be expressed either informally in English or formally in terms of predicates and logical connectives. Clearly, the formal language of business rules requires a precise formal semantics, so that the users of the language can understand what the rules mean and the conditions under which rules are violated.

The leading standard for business rules is *Semantics for Business Vocabulary and Business Rules* (SBVR) [1]. SBVR defines a very rich logical language for expressing business rules which includes full first order logic, and both alethic (necessity) and deontic (obligation) modalities. However, SBVR leaves some decisions concerning the precise syntax and semantics of business rules unspecified, while other parts of the

R. Setchi et al. (Eds.): KES 2010, Part IV, LNAI 6279, pp. 41–50, 2010.

specification appear ambiguous or contradictory. For example, SBVR states that the formal language of business rules corresponds to a fragment of full first order modal logic. However while the alphabet of this language is given and some general guidance regarding syntax (e.g., *usually*, each rule contains at most one modality, which is its main connective [1, p.101]), there is no definition of what constitutes a well-formed formula. Similarly, SBVR does not define a precise semantics for business rules. For example, at various points SBVR says that the alethic modality $\Box$ means 'true in all states' (which corresponds to an S5 modality), that it could be S4 [1, p.102] and at other points that the exact modal logic is left open [1, p.101]. SBVR also leaves open the precise semantics of the interaction of first order quantifiers and modalities, while at the same time assuming some logical equalities, such as Barcan formulas, that permute universal quantifiers and alethic modalities. The semantics of deontic modalities is also not completely specified [1, p.103], but some equalities of deontic logic are assumed, e.g., $(p \supset \mathbf{O}q) \equiv \mathbf{O}(p \supset q)$ [1, p.103], which says that some sentence p implies that some other sentence q is obligatory, iff it is obligatory that p implies q.

How these questions are resolved has a significant impact on the expressive power of the business rules formalism defined in SBVR and on the meaning of the rules themselves, and hence whether they accurately model the business processes they purport to formalise.

In this paper, we propose a precise syntax and semantics for modalities in business rules and their interaction with first order quantifiers. We show that the implication problem for business rules is undecidable, and characterise complexity of checking whether a state of the business process and its history violates a business rule.

The structure of the paper is as follows: in the next section, the main concepts used to define logical foundations of business rules in SBVR are briefly summarized. In the following sections we discuss syntax and semantics for static and dynamic business rules. In the last section, we propose a restricted syntax of static and dynamic business rules and analyse the compexity of reasoning with business rules.

2 Preliminaries

In this section, the main notions defined in SBVR and used later in the paper are briefly summarized.

A *ground fact* is a "proposition taken to be true by the business". Some facts state that a certain property holds for a specific individual or that some relation holds between individuals . An example would be "Employee 123 works for the sales department". Other facts declare the existence of an individual like "There is a company that has the company name 'Microsoft" [1, p.87-88]. Further, a *fact type* is a kind of ground fact such as "Employee works for Department" [1, p.86]. A ground fact "Employee 123 works for the sales department" is based on or instantiates the fact type "Employee works for Department". Below, we will simplify the account in SBVR and use a single relation symbol for fact types such as $WorksFor(x, y)$, without the additional variable sort information used in SBVR $WorksFor(Employee : x, Department : y)$; this does not make any formal difference but makes formulas more readable.

A *model* or *knowledge base*, as defined in SBVR, is a structure adopted to describe the "universe of discourse" or "business domain" indicating those aspects of the

business that are of interest. A model consists of a sequence of *states* over time. A state, in the sense used here, contains ground facts at a particular point of time [1, p.87].

Business rules are constraints which are used to define conditions on business states. *Static* constraints apply to individual states. An example given in [1, p.86] is

> *Each* employee was born on *at most one* date.

Business states which contain facts of the type "Employee born on Date" stating that some individual was born on two or more different dates are ruled out by this constraint.

Dynamic constraints impose restrictions on transitions between business states. An example in [1, p.86] is

> A person's marital status may change from single to married, but not from divorced to single.

The facts "a person's marital status is single", "a person's marital status is married", and "a person's marital status is divorced" are all based on the same fact type and transitions between them are 'controlled' by the dynamic constraint above, which is also constructed from the same fact type.

We consider the syntax and semantics of static business rules in the next section, and dynamic rules in the subsequent section.

3 Static Business Rules

The language for expressing static business rules given in SBVR contains atomic formulas, boolean connectives, first order quantifiers, alethic modalities $\Box$ and $\Diamond$, and deontic modalities **O**, **F** and **P**. A definition of a formula in this language is not given in SBVR. We state here a standard definition of a formula of a full first order language with alethic and deontic modalities:

$$\begin{aligned}\phi \ ::= \ & P^n(t_1, \ldots, t_n) \mid t_1 = t_2 \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \rightarrow \phi \mid \\ & \exists x\phi \mid \forall x\phi \mid \Box\phi \mid \Diamond\phi \mid \mathbf{O}\phi \mid \mathbf{P}\phi \mid \mathbf{F}\phi\end{aligned}$$

where P^n is an n-ary predicate symbol and $t_1, \ldots, t_n$ are terms (variables or constants), $\neg$, $\wedge$, $\vee$ and $\rightarrow$ are classical boolean connectives (not, and, or, implies), $\exists$ and $\forall$ are existential and universal quantifiers. This part of the definition corresponds to full first order logic. $\Box$ (referred to as necessity, or box modality) and $\Diamond$ (referred to as a possibility, or diamond modality) are called alethic modalities. Intuitively, $\Box\phi$ means that ϕ is true in all states and $\Diamond\phi$ means that ϕ is true in at least one state. $\Diamond\phi$ is definable as $\neg\Box\neg\phi$. Deontic modalities include **O** (it is obligatory that), **P** (it is permitted that) and **F** (it is forbidden that). $\mathbf{P}\phi$ is definable as $\neg\mathbf{F}\phi$ and $\mathbf{F}\phi$ is definable as $\mathbf{O}\neg\phi$.

In addition to the ordinary existential quantifiers $\exists x$, SBVR also adopts counting quantifiers $\exists^n$, $\exists^{0..n}(n \geqslant 1)$, $\exists^{n..}$, and $\exists^{n..m}(n \geqslant 1, m \geqslant 2)$ which mean "there are exactly n", "there are at most n", "there are at least n" and "there are at least n and at most m" respectively. These counting quantifiers can be defined in terms of the standard quantifiers in a standard way.

To avoid the paradoxes of standard deontic logic (SDL) and use the same semantics as for the static alethic rules, we propose to define deontic modalities using alethic ones and a sentential constant V for 'undesirable state-of affairs'.[1] Intuitively, $\mathbf{O}\phi$ will be interpreted as saying that all states of the knowledge base satisfying $\neg\phi$ are 'forbidden', or satisfy V:

$$\mathbf{O}\phi =_{def} \Box(\neg\phi \rightarrow V)$$
$$\mathbf{F}\phi =_{def} \mathbf{O}\neg\phi = \Box(\phi \rightarrow V)$$
$$\mathbf{P}\phi =_{def} \neg\mathbf{F}\phi = \Diamond(\phi \wedge \neg V)$$

A similar idea is suggested in SBVR [1, p.104-106] but it involves a predicate 'forbidden' applied to objectified facts.

3.1 Semantics

In giving a semantics for business facts and rules, we need to determine (a) whether constants denote the same object in different worlds or not, and (b) whether different possible worlds have the same domain of quantification or not, and if not, how the domain can change. Different answers to these questions result in different formulas being valid. For example, the converse Barcan formula $\Box\forall x\phi \rightarrow \forall x\Box\phi$ and Barcan formula $\forall x\Box\phi \rightarrow \Box\forall x\phi$ are only valid at the same time if the domain of quantification is the same in all worlds.

We assumpe that constants (such as 'John Smith') should denote the same object in different states of the knowledge base (this is referred to as Rigid Designation in first order modal logic) and that the set of objects which actually exist (and should be in the domain of quantification) changes from one state of the business to the other: new individuals appear and old ones are deleted from the database. In particular, the denotation of 'John Smith' may not 'exist' in some of the states. However, the second assumption seems to clash with the rigid designation assumption. An approach which accommodates both assumptions is to have an 'active domain' of quantification in each state and a general domain which is used for giving meaning to terms and predicates.

Formally, we interpret the language of first order modal logic in a possible worlds structure where each world intuitively corresponds to a state at a given point in time. Each world has its own domain of quantification which intuitively corresponds to the objects actively existing in that state. The union of all of the world domains corresponds to the 'universal domain' of all possible objects. In accordance with the intuition that 'necessary' means 'true in all possible worlds' [1, p.100], we define $\Box\phi$ to be true if ϕ is true in all possible worlds. This makes $\Box$ an S5 modality (see, for example, [3]).

Definition 1. *A model for the language of static business rules is a tuple* $\mathcal{M} = \langle W, D, \{D_w : w \in W\}, v\rangle$ *where*

[1] The sentential constant V was first proposed in [2]. The key idea is that "it is obligatory that p" can be taken to mean "if not-p, then V" where V is some bad state-of-affairs. For example, "It is obligatory that students go to bed on time" can be expressed as " If students don't go to bed on time, they won't feel energetic the next day", which is a bad state-of-affairs.

W is a non-empty set of possible worlds;
D is the non-empty universal domain of the model (the union of all world domains);
D_w is the domain of quantification in each $w \in W$. Objects in D_w are said to exist in w and for all w, $D_w \in D$.
v is the valuation function which interprets constants and predicate symbols in each $w \in W$, namely for a constant c, $v(w, c) \in D$ and for an n-ary predicate symbol P, $v(w, P) \subseteq D^n$.

The models satisfy the following condition (Rigid Designation):

$$\forall w_i \forall w_j (w_i \in W \wedge w_j \in W) \Rightarrow v(w_i, c) = v(w_j, c))$$

An inductive truth definition relative to a possible world w and a variable assignment a is given below. $\mathcal{M}, w, a \Vdash \phi$ means 'w satisfies ϕ with variable assignment a' or 'ϕ is true in the world w with variable assignment a'.

$\mathcal{M}, w, a \models P^n(t_1, \ldots, t_n)$ iff $(v_a(w, t_1), \ldots, v_a(w, t_n)) \in v(w, P^n)$ (where $v_a(t_i) = v(w, c)$ if t_i is a constant c and $v_a(t_i) = a(x)$ if t_i is a variable x)
$\mathcal{M}, w, a \models t_1 = t_2$ iff $v_a(w, t_1) = v_a(w, v_2)$;
$\mathcal{M}, w, a \models \neg\phi$ iff $\mathcal{M}, w, a \not\models \phi$;
$\mathcal{M}, w, a \models \phi \wedge \psi$ iff $\mathcal{M}, w, a \models \phi$ and $\mathcal{M}, w, a \models \psi$;
other propositional connectives are defined as usual;
$\mathcal{M}, w, a \models \forall x \phi(x)$ iff for each $d \in D_w$, $\mathcal{M}, w, a[d/x] \models \phi(x)$, where $a[d/x]$ is an assignment obtained from a by assigning d to x;
$\mathcal{M}, w, a \models \exists x \phi(x)$ iff there is a $d \in D_w$ such that $\mathcal{M}, w, a[d/x] \models \phi(x)$;
$\mathcal{M}, w, a \models \Box\phi$ iff for each $w' \in W$, $\mathcal{M}, w', a \models \phi$;
$\mathcal{M}, w, a \models \Diamond\phi$ iff there is a $w' \in W$ such that $\mathcal{M}, w', a \models \phi$.

The definitions of truth in a model and logical entailment are standard. A formula ϕ is true in a model, $\mathcal{M} \Vdash \phi$, if $\mathcal{M}, w, a \models \phi$ for all w and a. A set of sentences Γ entails a sentence ϕ, $\Gamma \models \phi$, if ϕ is true in all models where all the sentences of Γ are true.

Note that in the models defined above, both the Barcan formula $\forall x \Box \phi(x) \rightarrow \Box \forall x \phi(x)$ and its converse $\Box \forall x \phi(x) \rightarrow \forall x \Box \phi(x)$ are not valid. A simple counterexample to $\forall x \Box P(x) \rightarrow \Box \forall x P(x)$ is a model $\mathcal{M}$ where $W = \{w1, w2\}$, $D = \{d_1, d_2\}$, $D_{w_1} = \{d_1\}$, $D_{w_2} = \{d_1, d_2\}$, $v(w_1, P) = v(w_2, P) = \{d_1\}$. Then $\mathcal{M}, w_1 \models \forall x \Box P(x)$ but $\mathcal{M}, w_1 \not\models \Box \forall x P(x)$. A counterexample to the converse is $\mathcal{M}$ with the same W and D, but where $D_{w_1} = \{d_1, d_2\}$, $D_{w_2} = \{d_1\}$, $v(w_1, P) = \{d_1, d_2\}$ and $v(w_2, P) = \{d_1\}$. So, if our proposal for the semantics of alethic modalities and quantifiers is accepted, the equivalence $\forall x \Box \phi(x) \equiv \Box \forall \phi(x)$ suggested in SBVR must be given up.

Also, the deontic equivalence suggested in SBVR, $(p \rightarrow \mathbf{O}q) \equiv \mathbf{O}(p \rightarrow q)$ is not valid if $\mathbf{O}q$ is interpreted as $\Box(\neg q \rightarrow V)$.

As noted above, the truth definition for $\Box$ given above makes it an S5 modality. This runs contrary to the suggestion in SBVR that if a choice should be made, S4 [1, p.102] is preferred. However, this is an inevitable consequence of defining $\Box$ as 'true in all possible worlds'.

We now give some examples of static business rules from SBVR *Annex E: EU-Rent Example* and their translations into the first order modal logic defined above.

Example 1. It is necessary that each rental has exactly one requested car group: $\Box\forall x(Rental(x) \rightarrow \exists^1 y\, CarGroupOf(x, y))$

Example 2. It is permitted that a rental is open only if an estimated rental charge is provisionally charged to a credit card of the renter that is responsible for the rental: $\Diamond\exists x\exists y(Rental(x) \wedge Open(x) \wedge DriverOf(y, x) \wedge ChargedFor(x, y) \wedge \neg V)$ $\Box\forall x\forall y(Rental(x) \wedge Open(x) \wedge DriverOf(y, x) \wedge \neg ChargedFor(x, y) \rightarrow V)$

Example 3. It is obligatory that each driver of a rental is qualified: $\Box\forall x\forall y(Rental(x) \wedge DriverOf(y, x) \wedge \neg Qualified(y) \rightarrow V)$

4 Dynamic Business Rules

According to the SBVR specification, dynamic business rules or constraints restrict possible transitions between business states, and there are two types of dynamic constraints: those which simply compare one state to the next and those which may compare states separated by a given period. Dynamic business rules essentially place constraints on temporal sequences of states. For example, they assert that there is no sequence where a state where *MaritalStatus*(*john*, *single*) is true is immediately followed by a state where *MaritalStatus*(*john*, *divorced*) is true. Similarly, a rule that says that the price charged for the rental should be calculated using the lowest tariff which was applicable during the term of the rental says that for all sequences $w_1, \ldots, w_n$, if the rental started in w_1 and was charged in w_n, then the tariff charged in w_n should be the lowest value of the tariff in any of $w_1, \ldots, w_n$. SBVR discusses some possibilities for formal treatment of the semantics of dynamic business rules, including the use of temporal logic, but ultimately defers decisions on their semantics to a later version of the standard [1, p.120].

Our approach to formalizing dynamic rules is based on first-order timed linear time temporal logic (first-order timed LTL) using a similar syntax to [4]. Linear time temporal logic is well suited for stating constraints on discrete linear sequences. Its language contains temporal operators X ($X\phi$ means that ϕ is true in the next state) and U ($U(\phi, \psi)$ means that in some future state, ψ holds, and until then ϕ holds). Similarly to for example [5,6], we introduce timed constraints for the Until operators, of the form $U_{\sim n}$, where $\sim \in \{<, \leq, =, \geq, >\}$ and n is a natural number. The meaning of $U_{\leq n}(\phi, \psi)$ is that ψ will become true within n time units from now, and all states until then satisfy ϕ. Having timed constraints makes it a lot easier to express properties such as 'within 30 days' (which can be expressed in standard LTL using X operators but in a rather awkward way).

4.1 Syntax

In addition to the language of static business rules introduced in the previous section, for dynamic business rules we add temporal operators.

As we did for the static business rules, we distinguish alethic and deontic dynamic business rules; the former say what should always be or might be the case after executing a particular change, the latter specify those conditions leading to a 'bad state-of-affairs'. As before, we do not use the deontic modalities explicitly, but use the proposition V (forbidden) instead.

The definition of a formula is as for the static business rules, but there is an additional clause for formulas with temporal modalities: if ϕ and ψ are formulas, then $X\phi$ and $U_{\sim n}(\phi, \psi)$ are formulas.

4.2 Semantics

Definition 2. *A model for static and dynamic business rules $\mathcal{M}$ is a tuple $\langle W, D, \{D_w : w \in W\}, v, R\rangle$ where $\langle W, D, \{D_w : w \in W\}, v\rangle$ is as in Definition 1 and $R \subseteq W \times W \times \mathbb{N}$ is a temporal transition relation (where $\mathbb{N}$ is the duration of the transition). We assume that W is linearly ordered by $\exists d R(w, w', d)$ (that is, each world has a unique successor and $\mathcal{M}$ corresponds to a linear flow of time), that there is an initial state (moment of time) $w_0 \in W$, and the flow of time is unbounded to the right (for every w, there exists a w' and a d such that $R(w, w', d)$).*

To make some expressions more intuitive, we will sometimes denote a transition $(w_1, w_2, d) \in R$ by $w_1 \overset{d}{\rightarrow} w_2$.

$\pi = w_1 \overset{d_1}{\rightarrow} w_2 \overset{d_2}{\rightarrow} w_3 \ldots$ is called a path in $\mathcal{M}$ starting in w_1. Each model contains a single maximal path π_0 starting in w_0, and all other paths are suffixes of π_0. We define a function $t : W \rightarrow \mathbb{N}$ which assigns timestamps to the possible worlds in $\mathcal{M}$: $t(w_0) = 0$, and $t(w_n) = d_0 + \ldots + d_{n-1}$, where $w_0 \overset{d_0}{\rightarrow} w_1 \ldots \overset{d_{n-1}}{\rightarrow} w_n$.

All clauses of the truth definition are the same as in Definition 1, with the following additions:

$\mathcal{M}, w, a \models X\phi$ iff there exists w' and d such that $w \overset{d}{\rightarrow} w'$ and $\mathcal{M}, w', a \Vdash \phi$

$\mathcal{M}, w, a \models U_{\sim n}(\phi, \psi)$ iff there exists w_k with $k \geq 2$ such that for some $w_2, \ldots$ and $d_1, \ldots$, $w \overset{d_1}{\rightarrow} w_2 \overset{d_2}{\rightarrow} \ldots \overset{d_{k-1}}{\rightarrow} w_k$ and $\mathcal{M}, w_k, a \models \psi$, $t(w_k) - t(w) \sim n$ and for all $w_i \in \{w = w_1, \ldots, w_{k-1}\}$, $\mathcal{M}, w_i, a \models \phi$.

The definitions of truth in a model and logical entailment are standard.

As usual, other operators can be defined in terms of $U_{\sim n}(\phi, \psi)$. Namely, the plain (without a subscript) $U(\phi, \psi) =_{df} U_{\geq 0}(\phi, \psi)$, $F_{\sim n}\phi = U_{\sim n}(\top, \phi)$ (ϕ holds at some time in the future satisfying $\sim n$), and $G_{\sim n}\phi = \neg F_{\sim n}\neg\phi$ (suppose w_i is the current time point, then ϕ holds in all future points w_j, where $t(w_j) - t(w_i) \sim n$).

Below are some examples of dynamic business rules:

Example 4. It is necessary that the marital status of a person does not change from married to single: $\Box\forall x(\mathit{MaritalStatus}(x, \mathit{married}) \rightarrow \neg X\, \mathit{MaritalStatus}(x, \mathit{single}))$

Example 5. It is obligatory that the tariff charged for the rental is not greater than the lowest tariff applicable during the term of the rental: $\Box\forall x\forall y_1(\mathit{Rental}(x) \wedge \mathit{FCharged}(x, y_1) \wedge \neg U(\neg\exists y_2(y_2 < y_1 \wedge \mathit{Tariff}(x, y_2)), \mathit{Charged}(x, y_1)) \rightarrow V)$

Example 6. It is possible that the notification of a problem during a rental occurs after the return of the rental: $\Diamond\exists x\exists y(\mathit{Rental}(x) \wedge \mathit{Returned}(x) \wedge F\ \mathit{Problem}(x, y))$

Example 7. It is obligatory that each invoice which was issued on Date_1 is paid on Date_2 where $\text{Date}_2 \leq \text{Date}_1 + 30$ Days: $\Box\forall x(\mathit{Invoice}(x) \wedge \neg F_{\leq 30}\mathit{Paid}(x) \rightarrow V)$

5 Complexity of Reasoning

Turning to the complexity of reasoning with business rules, we observe that it does not seem to make sense to state the results for full first-order modal temporal logic. Clearly, business rules correspond to a proper fragment of it. For example, all business rules we have seen so far contain a single occurrence of the $\Box$ modality. From empirical analysis of the examples given in SBVR and other collections of business rules, we conclude that it is reasonable to constrain the logical syntax of static business rules to the following general form:

Static necessity constraints $\Box\forall\bar{x}(\phi \rightarrow \psi)$
Static possibility constraints $\Diamond\exists\bar{x}(\phi \wedge \psi)$

where $\bar{x}$ are all the free variables of $\phi \rightarrow \psi$, ϕ is a conjunction of atomic formulas, and ψ is an existentially quantified formula in disjunctive normal form (where existential quantifiers may be counting quantifiers).

Note that some of the constraints above, namely the second sentence in Example 2 and Example 3 were stated in a slightly different form, for readability. However, they can be easily rewritten to conform to the pattern above:

Example 2' $\Box\forall x\forall y(Rental(x) \wedge Open(x) \wedge DriverOf(y, x) \rightarrow ChargedFor(x, y) \vee V)$

Example 3' $\Box\forall x\forall y(Rental(x) \wedge Driver(y, x) \rightarrow Qualified(y) \vee V)$

For the dynamic business rules, we suggest the form

Dynamic necessity constraints $\Box\forall\bar{x}(\phi \rightarrow \psi)$
Dynamic possibility constraints $\Diamond\exists\bar{x}(\phi \wedge \psi)$

where ϕ and ψ are as before, but in addition are allowed to contain $\Box$-free formulas starting with a temporal modality. The latter may be a somewhat overgenerous definition, but on the other hand we do not want to preclude a possibility that the antecedent or the consequent of a rule may contain nested temporal modalities.

Before analyzing the complexity of reasoning with business rules, we translate them into the language of first order logic, for the convenience of proving the corresponding results. In the translation, each predicate acquires an extra argument place corresponding to a state where the formula is evaluated. The translation ST^w is analogous to the Standard Translation from modal logic to classical logic defined by van Benthem [7] (w is the parameter corresponding to the free state variable). Since we use the universal modality to interpret $\Box$, we do not need to introduce an explicit accessibility relation in the translation. However, we need to restrict quantification to the domains of the possible worlds. We do this by introducing a binary predicate $D(x, w)$ which corresponds to 'x is in the quantification domain of w'. We also need predicates R and R^+ corresponding to temporal relations (where R^+ is a strict transitive closure of $\exists d(R(x, y, d))$ and a functional symbol t corresponding to timestamps.

Formally, the translation ST^w is as follows:

$ST^w(P(x_1, \ldots, x_n)) = P(x_1, \ldots, x_n, w)$
ST^w commutes with the booleans

$$
\begin{aligned}
&ST^w(\forall x\phi) = \forall x(D(x,w) \rightarrow ST^w(\phi))\\
&ST^w(\exists x\phi) = \exists x(D(x,w) \wedge ST^w(\phi))\\
&ST^w(\Box\phi) = \forall w\, ST^w(\phi)\\
&ST^w(X\phi) = \exists w' \exists d(R(w,w',d) \wedge ST^{w'}(\phi))\\
&ST^w(U_{\sim n}(\phi,\psi)) = \exists w'(R^+(w,w') \wedge ST^{w'}(\psi) \wedge t(w') - t(w) \sim n \wedge \forall w''(w'' \neq w' \wedge\\
&\quad R^+(w,w'') \wedge R^+(w'',w') \rightarrow ST^{w''}(\phi)))
\end{aligned}
$$

Proposition 1. *ST^w is a truth preserving translation, that is $\mathcal{M}, w, a \Vdash \phi$ iff $M', a \Vdash ST^w(\phi)$ where $\mathcal{M} = \langle W, D, \{D_w : w \in W\}, v, R\rangle$ and $\mathcal{M}' = \langle W \cup D, D^2, R, R^+, v', t^1, w_0\rangle$ is a structure corresponding to $\mathcal{M}$, where the domain is $W \cup D$, a binary relation D^2 corresponds to $\{D_w : w \in W\}$, R^+ is a strict transitive closure of $\exists d R(w_1, w_2, d)$, v' is a set of relations given by v but with an extra worlds parameter, t^1 is a function $W \cup D \rightarrow \mathbb{N}$ where for $w \in W$, $t(w)$ is the distance from w_0 and for $e \in D$ $t(e) = 0$, and w_0 is the initial world.*

Proof. It is immediate from the Definition 2 that the translation is just a reformulation of the truth conditions for the formulas of modal temporal logic. It is easy to prove by induction on the complexity of the formula that the translation is truth preserving. □

The first problem we would like to consider is the implication problem for business rules. By the implication problem for business rules we mean the question whether a finite set of business rules Γ entails a business rule ϕ. Clearly, it is very important to be able to check whether a set of rules is consistent (entails a trivial business rule $\perp$). It may also be useful to check whether a set of rules is redundant (some rule is actually entailed by other rules). Both questions can be answered if we have an algorithm to solve the implication problem. Unfortunately, even if we restricted the syntax of business rules as proposed above, the problem is undecidable.

Proposition 2. *The implication problem for business rules is undecidable.*

Proof. The translation of business rules into first-order logic includes all formulas in the $\forall^*\exists^*$ prenex class which is undecidable [8]. □

We next turn to another natural question: are any constraints violated by a run of the system? Note that this problem only has practical meaning for necessity constraints, and even then only for a subset of them (basically the ones which translate into a purely universal first order sentence; we will refer to them as purely universal business rules). This is not a classical model-checking problem: whether a model $\mathcal{M}$ satisfies a sentence (business rule) ϕ. Rather, we are talking about *run-time* model-checking: whether a finite run of a process violates an invariant property. We view the current state of a business and its history as an initial prefix of a real (infinite) model. To see why it does not make sense to check for violations of possibility constraints, consider the following example of a simple possibility constraint $\Diamond\exists x(Employee(x) \wedge PartTime(x))$. It says that a state where someone is a part-time employee is possible — in principle; it does not say that such a state should have actually existed in the past history of the business, so if it does not hold in the current state or any of the preceding states, we still should not consider it violated. Similarly, a necessity constraint which says that each invoice should be paid within 30 days is not violated — yet — if there exists an unpaid invoice,

but it has just been issued. However, this constraint can be reformulated as 'All invoices issued 30 or more days ago should have been paid', which is a universal statement and can be meaningfully checked for the finite history and the present state of the business.

Formally, we define as a 'state and history of the business' $\mathcal{S}$ a finite prefix of a model $\mathcal{M}$. It corresponds to a finite linearly ordered sequence of states with the initial state w_0 (when the records began) and the current state w_n. We define the problem of checking whether a state and history of the business satisfies a universal business rule ϕ as the problem of checking whether $\mathcal{S}, w_0 \Vdash \phi$.

Proposition 3. *The problem of checking whether a state and history of the business $\mathcal{S}$ satisfies a purely universal business rule ϕ is solvable in time polynomial in the size of $\mathcal{S}$.*

Proof. Using the translation ST^{w_0} and Proposition 1, we can reduce this problem to the question whether a finite first order structure $\mathcal{S}'$ satisfies a first-order sentence $ST^{w_0}(\phi)$. It is well known, see for example [9], that this problem can be solved in time polynomial in the size of $\mathcal{S}'$. Note that the size of $\mathcal{S}$ is the same as the size of $\mathcal{S}'$ apart from the explicit representation of R^+ and t. However R^+ and t can be computed in polynomial time and only result in a polynomial increase in the size of $\mathcal{S}'$. □

6 Conclusions

In this paper, we made a proposal for the precise formal syntax of business rules and their logical semantics. We have also shown that the implication problem for business rules is undecidable, and a natural problem of checking for violation of universal constraints can be solved in time polynomial in the size of the business state and history. In future work we plan to investigate the design and implementation of efficient algorithms for solving the latter problem.

References

1. OMG: Semantics of business vocabulary and business rules (SBVR) (2008)
2. Anderson, A.R.: Some nasty problems in the formal logic of ethics. Noûs 1, 345–360 (1967)
3. Bull, R., Segerberg, K.: Basic modal logic. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic. Extensions of Classical Logic, vol. II, pp. 1–81. Kluwer, Dordrecht (2001)
4. Kare, J., Kristoffersen, C.P., Andersen, H.R.: Checking temporal business rules. In: Proceedings of the First International REA Workshop (2004)
5. Alur, R., Courcoubetis, C., Dill, D.L.: Model-checking in dense real-time. Inf. Comput. 104(1), 2–34 (1993)
6. Laroussinie, F., Markey, N., Schnoebelen, P.: Efficient timed model checking for discrete-time systems. Theor. Comput. Sci. 353(1-3), 249–271 (2006)
7. van Benthem, J.: Modal logic and classical logic. Bibliopolis (1983)
8. Börger, E., Grädel, E., Gurevich, Y.: The Classical Decision Problem. Springer, Heidelberg (1997)
9. Ebbinghaus, H.-D., Flum, J.: Finite model theory. Springer, Heidelberg (1995)