A methodology for plan revision under norm and outcome compliance

Simone Scannapieco$^{\bullet *\dagger}$, Guido Governatori$^{\bullet *}$,
Francesco Olivieri$^{\bullet *\dagger}$, and Matteo Cristani†

†Department of Computer Science, University of Verona, Italy
$^{\bullet}$NICTA, Queensland Research Laboratory, Australia*
*Institute for Integrated and Intelligent Systems, Griffith University, Australia

Abstract. Scholars understand an agent as a system acting in an environment. Such an environment is usually governed by norms, and the agent has to obey to such norms when pursuing her objectives. We report a non-monotonic modal logic able to describe the environment, the norms, and the agent's capabilities as well as her mental attitudes (e.g., desires, intentions). First, we show how such a logic is expressive enough to determine when the agent is compliant with respect to norms and objectives by extending it with a formal characterisation of the concepts of norm and outcome compliance. Then, in the case the agent violates some norms or does not achieve all her objectives, we propose a preliminary analysis of methodologies to revise the theory and restore compliance.

Keywords: Norm compliance, goal compliance, Defeasible Logic, business process, logic-based revision

1 Introduction

An agent is a system which operates in the environment where she is embedded driven by a set of objectives. Typically, an agent is equipped with a library of plans and, based on her mental attitudes, she deliberates on the course of action to adopt to achieve her objectives [1].

We depart from the standard architecture proposed in [2,3,4] where an agent selects a plan from her plan library and instead we assume that the agent generates alternative plans to reach a particular set of objectives in the form of a *business process* (workflows), derived from the declarative specifications of the agent's knowledge base. This is in line with the classical definition of business process as the set of all the possible ways in which it can be executed, i.e., the set of its execution traces [5]. In turn, the notion of trace is compatible with the classical AI definition of plan as a sequence of actions that are triggered by pre-conditions and that generate some effects (post-conditions) [6].

The constraints imposed by the environment play an important role in choosing the most suitable plan for a given circumstance. In fact, the environment is usually governed

[$\star$] NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy, the Australian Research Council through the ICT Centre of Excellence program and the Queensland Government.

by a set of rules describing the “right way” of behaving, which are expressed in various forms (e.g., guidelines, norms, laws, regulations).

In this perspective, it is possible to consider scenarios where an agent is not able to achieve all her objectives without violating the norms. That is to say, there is no possible way for the process to be executed without either violating the norms, or achieving some objectives. In other words, the whole business process is not compliant [7,8].

Consider the following example. Robbie the Robot is downtown to deliver some mail for its boss, Carmen. It is just done with the task and thinking to get a (well-deserved) ice-cold oil-beer, when Carmen calls him: the meeting she was supposed to attend at 3pm has been anticipated and will start in one hour. Therefore, she tells him to bring some food at her office since she will not be able to have lunch at her favourite restaurant for lack of time. Moreover, Robbie cannot forget to pick up a suit for Carmen, which is necessary for the presentation she will hold during the meeting.

Robbie’s internal system elaborates four plans. Plan A is to go to the dry-cleaner to get Carmen’s favourite suit and then to a take-away, but to get all these things done in time, it needs to drive beyond the speed limit, thus violating the traffic laws. Plan A_1 is not to stop at the dry-cleaner, while plan A_2 is not to stop to get the food (not fulfilling one objective in both cases). Instead, Plan B involves going to Carmen’s place to get another suit and then stop at a fast-food close by (Carmen’s less favourite food). In this way, there is no time for the ice-cold oil-beer. Thus, whatever plan Robbie decides to adopt, either a violation of the norms occurs, or some of the objectives are not fulfilled.

The main aim of this paper is to start an analytical study on how to restore compliance of a business process with respect to objectives and obligations. More specifically, we search for a methodology to revise the knowledge base of the agent to obtain a new plan which complies with the governing norms and meets her objectives.

To integrate the previous example, suppose Robbie has recently been equipped with an (on-line) unit which can suggest alternative plans. Strangely enough, the unit informs Robbie that there is a grocery store close to Carmen’s home. Just by downloading from the server the task-programmes *buy_groceries* and *make_a_sandwich*, Robbie may indeed buy all the necessary ingredients to prepare an healthy sandwich (which Carmen prefers to a fast-food burger), get an oil-beer six-pack and return to Carmen’s office in time without driving fast. With great delight, Robbie opts for this plan, happy and proud of its new software.

To succeed in our endeavour, we start from the logic presented in [9] which is a modal variant of Defeasible Logic (DL) [10]. The logic is inspired by the BDI (Belief-Desire-Intention) architecture [2,3,4] and defines a framework to model rational agents, based on three kind of knowledge:

(i) the internal constraints guiding the agent (i.e., her mental attitudes);
(ii) the external constraints she has to obey to (i.e., norms);
(iii) her vision of the world (i.e., her beliefs) along with the actions she is able to perform in the form of *rules*; each action has its pre-conditions (inputs) and post-conditions (effects).

Equipped with this knowledge, the agent is able to choose different courses of action to reach the objectives she decides to achieve.

The paper is organised as follows. In Section 2, we first illustrate the logical formalism, and we end by discussing the notion of compliance with respect to norms and objectives (Subsection 2.3). Section 3 gives an overview of existing formalisms and suggests a methodology to address business process revision using the logical framework proposed. We summarise and comment some future works in Section 4.

2 Logic

Despite the traditional treatment of internal constraints in the BDI architecture, the logic proposed in [9] considers all mental attitudes (desires, goals, intentions, and social intentions) as facets of the same concept of *outcome*, which is regarded as something the agent would like to achieve. As a result, mental attitudes were modelled by using a single type of rule, called *outcome rule*, whose nature is very similar to that of *reparative chains* modelling the mechanism of norm violation/reparation [11]. Given the rule $r : \mathit{\Gamma} \Rightarrow o_1 \otimes o_2 \otimes \cdots \otimes o_n$, if the context described by $\mathit{\Gamma}$ holds, then the obligation in force is o_1; in case o_1 is violated then the new obligation in force is o_2, and so on. Similarly, we use the same rationale to model an agent which ranks alternative outcomes by a *preference ordering*. Using only one type of rule to derive all agent's mental attitudes reflects a natural way to express alternatives to more preferred outcomes. The agent then deploys her knowledge of the environment to filter out the actual objectives she will commit to from all her preferences. For instance, consider the following setting. Alice, during her holidays, plans to pay a visit to her friend John, who lives close to her parents. A possible plan for Alice is described by the following modal outcome rule, where the notation $\Rightarrow_{\mathrm{U}}$ denotes the fact that rule s is used to derive oUtcomes.

$$s : \mathit{holiday} \Rightarrow_{\mathrm{U}} \mathit{visit\_John} \odot \mathit{visit\_parents} \odot \mathit{stay\_home}$$

The intended meaning is *"I shall come over to John's place to visit him, but if it is not possible (for instance, if he is not home), I am going to visit my parents. If this is not possible as well, I shall take some rest at home"*.

Four different types of mental attitudes were identified: desires, goals, intentions, and social intentions, whose meaning is the following.

Desires as acceptable outcomes. Consider an agent equipped with the outcome rules

$$r : a_1, \ldots, a_n \Rightarrow_{\mathrm{U}} b_1 \odot \cdots \odot b_m \qquad\qquad s : a'_1, \ldots, a'_n \Rightarrow_{\mathrm{U}} b'_1 \odot \cdots \odot b'_k$$

and that the situation described by $a_1, \ldots, a_n$ and $a'_1, \ldots, a'_n$ are mutually compatible but b_1 and b'_1 are not, namely $b_1 = \neg b'_1$. In this case $b_1, \ldots, b_m, b'_1, \ldots, b'_k$ are anyway all *acceptable outcomes*, including the incompatible outcomes b_1 and b'_1. *Desires* are expected or acceptable outcomes, independently of whether they are compatible with other expected or acceptable outcomes.

Goals as preferred outcomes. For rule r alone the preferred outcome is b_1, and for rule s alone it is b'_1. But if both rules are applicable, then the agent would not be rational if she considers both b_1 and $\neg b_1$ as her preferred outcomes. Hence, the agent has to decide if she prefers a state where b_1 holds to one where b'_1 (i.e., $\neg b_1$) holds, or the other way around. If the agent has no way to decide which is the most suitable option for her, then neither the chain of r nor that of s can produce preferred outcomes. Suppose that the agent opts for b'_1; this can be done if she establishes that the second rule overrids the first one, i.e., $s > r$. Accordingly, the preferred outcome is b'_1 for the chain of outcomes defined by s, and b_2 is the preferred outcome of r. b_2 is the second best alternative according to rule r: in fact b_1 has been discarded as an acceptable outcome given that s prevails over r.

Two degrees of commitment: intentions and social intentions. We now clarify which are the outcomes for an agent to commit to. Naturally, if the agent values some outcomes more than others, she should strive for the best, i.e., for the most preferred outcomes.

We first consider the case where only rule r applies. Here, the agent should commit to the outcome she values the most, i.e., b_1. But what if the agent *believes* that b_1 cannot be achieved in the environment, or she knows that $\neg b_1$ holds? Committing to b_1 would result in a waste of agent's resources; rationally, she should target the next best outcome, in this case b_2. Accordingly, the agent derives b_2 as her *intention*. Suppose, now, that b_2 is *forbidden*, and the agent is social (an agent is social if the agent would not knowingly commit to anything that is forbidden [12]). Once again, in this situation the agent has to lower her expectation and settle for b_3, which is considered her *social intention*.

To complete the analysis, consider the situation where both rules r and s apply and the agent prefers s to r. As we have seen before, $\neg b_1$ (b'_1) and b_2 are the preferred outcomes since the agent stated $s > r$. Assume that, this time, the agent knows she cannot achieve $\neg b_1$ (or equivalently, b_1 holds). If the agent is rational, she cannot commit to $\neg b_1$. Thus, the best option for her is to commit to b'_2 and b_1 (both regarded as intentions and social intentions), where she is guaranteed to be successful. In this scenario, the best course of action for the agent is where she commits herself to some outcomes that are not her preferred ones, or even that she would consider not acceptable based only on her preferences, but such that they influence her decision process given that they represent relevant external factors (either her beliefs or the norms that apply to her).

2.1 Language

Let PROP be a set of propositional atoms, $\mathrm{MOD} = \{\mathrm{B}, \mathrm{O}, \mathrm{D}, \mathrm{G}, \mathrm{I}, \mathrm{SI}\}$ the set of modal operators, whose reading is B for *belief*, O for *obligation*, D for *desire*, G for *goal*, I for *intention* and SI for *social intention*. Let Lab be a set of arbitrary labels. The set $\mathrm{Lit} = \mathrm{PROP} \cup \{\neg p | p \in \mathrm{PROP}\}$ denotes the set of *literals*. The *complementary* of a literal q is denoted by $\sim q$; if q is a positive literal p, then $\sim q$ is $\neg p$, and if q is a negative literal $\neg p$ then $\sim q$ is p. The set of *modal literals* is $\mathrm{ModLit} = \{\Box l, \neg \Box l | l \in \mathrm{Lit}, \Box \in \{\mathrm{O}, \mathrm{D}, \mathrm{G}, \mathrm{I}, \mathrm{SI}\}\}$. We assume that modal operator "$\Box$" for belief B is the empty modal operator. Accordingly, a modal literal Bl is equivalent to literal l; the complementary of B$\sim l$ and $\neg$Bl is l. We define a *defeasible theory D* as a structure $(F, R, >)$, where

(i) $F \subseteq \mathrm{Lit} \cup \mathrm{ModLit}$ is a set of *facts* or indisputable statements;
(ii) R contains three sets of *rules*: for beliefs, obligations, and outcomes;
(iii) $> \subseteq R \times R$ is a *superiority relation* to determine the relative strength of conflicting rules.

Belief rules are used to relate the factual knowledge of an agent (her vision of the environment), and defines the relationships between states of the world. As such, provability for beliefs does not generate modal literals. *Obligation rules* determine when and which obligations are in force. The conclusions generated by obligation rules are modalised with obligation. Finally, *outcome rules* establish the possible outcomes of an agent depending on the particular context. Apart from obligation rules, outcome rules are used to derive conclusions for all modes denoting possible types of outcomes: desires, goals, intentions, and social intentions.

Following ideas given in [11], rules can gain more expressiveness when a *preference operator* $\odot$ is used: an expression like $a \odot b$ means that if a is possible, then a is the first choice and b is the second one; if $\neg a$ holds, then the first choice is not attainable and b is the actual choice. This operator is used to build chains of preferences, called $\odot$*-expressions*. The formation rules for $\odot$-expressions are: (i) every literal is an $\odot$-expression; (ii) if A is an $\odot$-expression and b is a literal then $A \odot b$ is an $\odot$-expression. In this paper, we exploit the classical definition of *defeasible rule* in DL [10]. A defeasible rule is an expression $r : A(r) \Rightarrow_{\Box} C(r)$, where

(i) $r \in \mathrm{Lab}$ is the name of the rule;
(ii) $A(r) = \{a_1, \dots, a_n\}$ with $a_i \in \mathrm{Lit} \cup \mathrm{ModLit}$ is the set of the premises (or the *antecedent*) of the rule;
(iii) $\Box \in \{\mathrm{B}, \mathrm{O}, \mathrm{U}\}$ represents the *mode* of the rule (from now on, we omit the subscript B in rules for beliefs, i.e., $\Rightarrow$ is used as a shortcut for $\Rightarrow_{\mathrm{B}}$);
(iv) $C(r)$ is the *consequent* (or *head*) of the rule, which is a single literal if $\Box = \mathrm{B}$, or an $\odot$-expression otherwise. Notice that modal literals can occur only in the antecedent of rules: the reason is that the rules are used to derive modal conclusions and we do not conceptually need to iterate modalities. The motivation of a single literal as a consequent for belief rules is dictated by the intended reading of the belief rules, where these rules are used to describe the environment.

We use the following abbreviations on sets of rules: $R^{\Box}$ ($R^{\Box}[q]$) denotes all rules of mode $\Box$ (with consequent q), and $R[q] = \bigcup_{\Box \in \{\mathrm{B},\mathrm{O},\mathrm{U}\}} R^{\Box}[q]$. $R[q,i]$ denotes the set of rules whose head is $\odot_{j=1}^{n} c_j$ and $c_i = q$, with $1 \leq i \leq n$.

Notice that labelling the rules of DL produces nothing more but a simple treatment of the modalities, thus two interaction strategies between modal operators are analysed.

Rule conversions and conflict-detection/resolution. It is sometimes meaningful to use rules for a modality $\Box$ as they were for another modality $\blacksquare$, i.e., to convert one type of conclusions into a different one. Formally, given rule $r : a_1, \dots, a_n \Rightarrow_{\Box} b$ and the situation where $\blacksquare a_1, \dots, \blacksquare a_n$ hold, the asymmetric binary relation $\mathrm{Convert}(\Box, \blacksquare)$ permits to derive $\blacksquare b$. For instance, if statement $A =$ "buy a car" implies statement $B =$ "spend money" and we have the intention to buy a car, then we may conclude that we also have the intention to spend money (if $A \Rightarrow B$ and $\mathrm{I}A$, then $\mathrm{I}B$). In our framework, we consider $\mathrm{Convert}(\mathrm{B}, \Box)$ with $\Box \in \mathrm{MOD} \setminus \{\mathrm{B}\}$. Accordingly, we enrich the notation with $R^{\mathrm{B},\Box}$, denoting the set of belief rules that can be used for a conversion to mode $\Box$. The antecedent of all such rules is not empty, and does not contain any modal literal.

Moreover, it is crucial to identify criteria for detecting and solving conflicts between different modalities. Formally, we define an asymmetric binary relation $\mathrm{Conflict} \subseteq \mathrm{MOD} \times \mathrm{MOD}$ such that $\mathrm{Conflict}(\Box, \blacksquare)$ means 'modes $\Box$ and $\blacksquare$ are in conflict and mode $\Box$ prevails over $\blacksquare$'. In our framework, we consider $\mathrm{Conflict} = \{(\mathrm{B}, \mathrm{I}), (\mathrm{B}, \mathrm{SI})\}$ defining the realistic attitude of the agents [13], and Conflict $= \{(\mathrm{O}, \mathrm{SI})\}$ defining the social attitude of the agents [12]. For instance, a social agent cannot have the social intention to smoke in a public place where it is forbidden to smoke.

There are two applications of the *superiority relation*. The first considers rules of the same mode. The latter compares two rules of different modes, with complementary literals and the two modes are related by the Convert relation.

2.2 Inferential Mechanism

A *proof* P of *length* n is a finite sequence $P(1), \ldots, P(n)$ of *tagged literals* of the type $+\partial_{\Box} q$ and $-\partial_{\Box} q$, where $\Box \in$ MOD. As a conventional notation, $P(1..i)$ denotes the initial part of the sequence P of length i. Given a defeasible theory D, $+\partial_{\Box} q$ means that q is defeasibly provable in D with the mode $\Box$, and $-\partial_{\Box} q$ that it has been proved in D that q is not defeasibly provable in D with the mode $\Box$. From now on, the term *refuted* is a synonym of *not provable* and we use $D \vdash \pm\partial_{\Box} l$ iff there is a proof P in D such that $P(n) = \pm\partial_{\Box} l$ for an index n.

To characterise the notions of provability for all modalities, it is essential to define when a rule is *applicable* or *discarded*. To this end, the preliminary notion of when a rule is *body-applicable*/*body-discarded* is needed, stating that each literal in the body of the rule must be proved/refuted with the suitable mode.

Definition 1. *Let P be a proof and $\Box \in \{\mathrm{O}, \mathrm{D}, \mathrm{G}, \mathrm{I}, \mathrm{SI}\}$. A rule $r \in R$ is* body-applicable *(at step $n+1$) iff for all $a_i \in A(r)$:*

1. *if $a_i = \Box l$ then $+\partial_{\Box} l \in P(1..n)$,*
2. *if $a_i = \neg\Box l$ then $-\partial_{\Box} l \in P(1..n)$,*
3. *if $a_i = l \in \mathrm{Lit}$ then $+\partial l \in P(1..n)$.*

A rule $r \in R$ is body-discarded *(at step $n+1$) iff there is $a_i \in A(r)$ such that*

1. *$a_i = \Box l$ and $-\partial_{\Box} l \in P(1..n)$, or*
2. *$a_i = \neg\Box l$ and $+\partial_{\Box} l \in P(1..n)$, or*
3. *$a_i = l \in \mathrm{Lit}$ and $-\partial l \in P(1..n)$.*

As already stated, belief rules allow us to derive literals with different modes. The applicability mechanism must take into account this constraint.

Definition 2. *Let P be a proof. A rule $r \in R$ is 1.* Conv-applicable*, 2.* Conv-discarded *(at step $n+1$) for $\Box$, $\Box \in \mathrm{MOD} \setminus \{\mathrm{B}\}$, iff*

1. *$r \in R^{\mathrm{B}}$, $A(r) \neq \emptyset$ and for all $a \in A(r)$, $+\partial_{\Box} a \in P(1..n)$;*
2. *$r \notin R^{\mathrm{B}}$ or $A(r) = \emptyset$ or $\exists a \in A(r)$, $-\partial_{\Box} a \in P(1..n)$.*

As an example, consider theory $D = (\{a, b, \mathrm{O}c\}, \{r_1 : a \Rightarrow_{\mathrm{O}} b, r_2 : b, c \Rightarrow d\}, \emptyset)$. Rule r_1 is applicable, while r_2 is not since c is not proved as a belief. Instead, r_2 is Conv-applicable for O, since Oc is a fact and r_1 proves Ob.

The notion of applicability gives guidelines on how to consider the next element in a given chain. Since a rule for belief cannot generate reparative chains but only single literals, we can conclude that the applicability condition for belief collapses into body-applicability. The same happens to desires, where we also consider the Convert relation. For obligations, each element before the current one must be a violated obligation. A literal is a candidate to be a goal only if none of the previous elements in the chain have been proved as a goal. For intentions, the elements of the chain must pass the wishful thinking filter, while social intentions are also constrained not to violate any norm.

Definition 3. *Given a proof P, $r \in R[q,i]$ is* applicable *(at index i and step $n+1$) for*

1. B *iff $r \in R^{\mathrm{B}}$ and is body-applicable.*
2. O *iff either: (2.1) (2.1.1) $r \in R^{\mathrm{O}}$ and is body-applicable,*
   (2.1.2) $\forall c_k \in C(r), k < i, +\partial_{\mathrm{O}} c_k \in P(1..n)$ and $-\partial c_k \in P(1..n)$, or
   (2.2) r is Conv-applicable.
3. D *iff either: (3.1) $r \in R^{\mathrm{U}}$ and is body-applicable, or*
   (3.2) Conv-applicable.
4. $\square \in \{\mathrm{G}, \mathrm{I}, \mathrm{SI}\}$ *iff either: (4.1) (4.1.1) $r \in R^{\mathrm{U}}$ and is body-applicable, and*
   (4.1.2) $\forall c_k \in C(r), k < i, +\partial_{\blacksquare} {\sim} c_k \in P(1..n)$ for some $\blacksquare$ such that $\mathrm{Conflict}(\blacksquare, \square)$ *and $-\partial_{\square} c_k \in P(1..n)$, or*
   (4.2) r is Conv-applicable.

   For G *there are no conflicts; for* I *we have* $\mathrm{Conflict}(\mathrm{B}, \mathrm{I})$*, and for* SI *we have* $\mathrm{Conflict}(\mathrm{B}, \mathrm{SI})$ *and* $\mathrm{Conflict}(\mathrm{O}, \mathrm{SI})$.

Conditions to establish that a rule is discarded correspond to the constructive failure to prove that the same rule is applicable, and follow the principle of *strong negation*. The strong negation principle is closely related to the function that simplifies a formula by moving all negations to an inner most position in the resulting formula, and replaces the positive tags with the respective negative tags, and the other way around [14].

We can now describe the proof conditions for the various modal operators; we start with those for desires:

$+\partial_{\mathrm{D}}$: If $P(n+1) = +\partial_{\mathrm{D}} q$ then
(1) $\mathrm{D}q \in \mathrm{F}$ or
(2) (2.1) $\neg \mathrm{D}q \notin \mathrm{F}$ and
 (2.2) $\exists r \in R[q,i]$: r is applicable for D and
 (2.3) $\forall s \in R[{\sim} q, j]$ either (2.3.1) s is discarded for D, or (2.3.2) $s \not> r$.

We say that a *desire* is each element in a chain of an outcome rule for which there is no stronger argument for the opposite desire. The proof conditions for $+\partial_{\square}$, with $\square \in \mathrm{MOD} \setminus \{\mathrm{D}\}$ are as follows:

$+\partial_{\square}$: If $P(n+1) = +\partial_{\square} q$ then
(1) $\square q \in \mathrm{F}$ or
(2) (2.1) $\neg \blacksquare q \notin \mathrm{F}$ for $\blacksquare = \square$ or $\mathrm{Convert}(\blacksquare, \square)$ and
 (2.2) $\exists r \in R[q,i]$: r is applicable for $\square$ and
 (2.3) $\forall s \in R^{\blacksquare}[{\sim} q, j]$ either
  (2.3.1) s is discarded for $\blacksquare$, or
  (2.3.2) $\exists t \in R^{\diamond}[q,k]$: t is applicable for $\diamond$ and either
   (2.3.2.1) $t > s$ if $\blacksquare = \diamond$, $\mathrm{Convert}(\blacksquare, \diamond)$, or $\mathrm{Convert}(\diamond, \blacksquare)$; or
   (2.3.2.2) $\mathrm{Conflict}(\diamond, \blacksquare)$.

To show that a literal q is defeasibly provable with modality $\Box$ we have two choices: (1) modal literal $\Box q$ is a fact; or (2) we need to argue using the defeasible part of D. In this case, we require that a complementary literal (of the same modality, or of a conflictual modality) does not appear in the set of facts (2.1), and that there must be an applicable rule for q for mode $\Box$ (2.2). Moreover, each possible attack brought by a rule s for $\sim q$ has to be either discarded (2.3.1), or successfully counterattacked by another stronger rule t for q (2.3.2). We recall that the superiority relation combines rules of the same mode, rules with different modes that produce complementary conclusion of the same mode through conversion (both considered in clause (2.3.2.1)), and conflictual modalities (clause 2.3.2.2). Obviously, if $\Box = \mathrm{B}$, then the proof conditions reduce to those of classical DL [10].

Again, the negative counterparts ($-\partial_{\mathrm{D}}$ and $-\partial_{\Box}, \Box \in \mathrm{MOD} \setminus \{\mathrm{D}\}$) are derived by strong negation applied to conditions for $+\partial_{\mathrm{D}}$ and $+\partial_{\Box}$, respectively.

Example 1. Let us consider the theory $D = (\{\neg b_1, \mathrm{O}\neg b_2, \mathrm{SI} b_4\}, \{r : \Rightarrow_{\mathrm{U}} b_1 \odot b_2 \odot b_3 \odot b_4\}, \emptyset)$. Then r is trivially applicable for D and $+\partial_{\mathrm{D}} b_i$ holds, for $1 \leq i \leq 4$. Moreover, we have $+\partial_{\mathrm{G}} b_1$ and r is discarded for G after b_1. Since $+\partial \neg b_1$, $-\partial_{\mathrm{I}} b_1$ holds (as well as $-\partial_{\mathrm{SI}} b_1$); the rule is applicable for I and b_2, and we are able to prove $+\partial_{\mathrm{I}} b_2$, thus the rule becomes discarded for I after b_2. Given that $\mathrm{O}\neg b_2$ is a fact, r is discarded for SI and b_2 and $-\partial_{\mathrm{SI}} b_2$ is proved, which in turn makes the rule applicable for SI at b_3, proving $+\partial_{\mathrm{SI}} b_3$. As we have argued before, this would make the rule discarded for b_4. Nevertheless, b_4 is still provable with mode SI (in this case because it is a fact, but in other theories there could be more rules with b_4 in their head).

In [9], authors showed the coherency and consistency of the logical apparatus.

2.3 Norm and Outcome Compliance

Our logic is able to model in a natural way the concepts of being compliant with respect to norms and outcomes. Consider the obligation rule $r : \Gamma \Rightarrow_{\mathrm{O}} o_1 \odot o_2 \odot o_3$ in a theory where $\mathrm{O} o_1$ and $\mathrm{O} o_2$ are the case. To be compliant with r, the agent has either to prove $\mathrm{B} o_1$, or to compensate by deriving $\mathrm{B} o_2$.

To formalise the concept of compliance, we first introduce a new literal $\perp$ whose interpretation is a not compliant situation, and we provide proof conditions to (defeasibly) derive it. We exploit the modal derivations of $\perp$ to formally characterise norm compliant ($-\partial_{\mathrm{O}} \perp$) and outcome compliant ($-\partial_{\Box} \perp, \Box \in \{\mathrm{G}, \mathrm{I}, \mathrm{SI}\}$) situations.

$-\partial_{\mathrm{O}} \perp$: If $P(n+1) = -\partial_{\mathrm{O}} \perp$ then
(1) $\forall r \in R^{\mathrm{O}} \cup R^{\mathrm{B},\mathrm{O}}$ either r is discarded or either
  (2.1) $\forall c_i \in C(r)$, $-\partial_{\mathrm{O}} c_i \in P(1..n)$, or
  (2.2) $\exists c_i \in C(r)$ such that $+\partial_{\mathrm{O}} c_i \in P(1..n)$ and $+\partial c_i \in P(1..n)$.

To be norm compliant, all applicable rules producing an obligation are such that either all elements in the consequent are not actually active obligations (condition (2.1)), or one element c_i is an obligation in force and is fulfilled (condition (2.2)). The situation is slightly different when addressing outcome compliance.

$-\partial_{\Box}\bot$: If $P(n+1) = -\partial_{\Box}\bot$ ($\Box \in \{\mathrm{G}, \mathrm{I}, \mathrm{SI}\}$) then
(1) $\forall r \in R^{\mathrm{U}} \cup R^{\mathrm{B},\Box}$, Conflict($\blacksquare, \Box$), either r is discarded or
(2) (2.1) $\exists c_i \in C(r)$ such that $+\partial_{\Box} c_i \in P(1..n)$, $\forall c_j \in C(r), j < i, -\partial_{\Box} c_j \in P(1..n)$, and
    (2.2) $\exists c_k, k \geq i$ such that
        (2.2.1′) (2.1.1.1′) if $k = i$ then $+\partial c_k \in P(1..n)$, or
                (2.1.1.2′) if $k \neq i$ then $+\partial c_k$ and $+\partial_{\mathrm{D}} c_k \in P(1..n)$.
        (2.2.1″) $+\partial c_k \in P(1..n)$ and $-\partial_{\Box} {\sim} c_k \in P(1..n)$.

First, the agent chooses her level of commitment, that is the mode $\Box$ among G, I, or SI to comply with (notice that in some cases this process is not particularly meaningful, e.g., desires). Then, we select the first element proved with modality $\Box$ in the consequent of any applicable rule (element c_i in the proof condition (2.1)). We propose two variants of outcome compliance corresponding to sub-conditions (2.2.1′) and (2.2.1″).

In the first case, we are compliant iff either c_i is proved as a belief (being the first element in the chain proved with modality $\Box$), or if there exists a following element c_k which has been proved as a desire as well as a belief. In the latter, we are outcome compliant with respect to r if an element c_k following c_i has been proved as a belief, and its opposite has not been chosen as an outcome to achieve. It may be the case that, semantically but not syntactically, if $\neg\Box{\sim}c_k$ is the case then $+\partial_{\mathrm{D}} c_k$, but this is left to further analysis.

Again, the counterparts $+\partial_{\mathrm{O}}\bot$ and $+\partial_{\Box}\bot, \Box \in \{\mathrm{G}, \mathrm{I}, \mathrm{SI}\}$ are derived by strong negation applied to conditions for $-\partial_{\mathrm{O}}\bot$ and $-\partial_{\Box}\bot$, respectively.

Example 2. We now formalise the "Robbie the Robot" example to show how compliance definitions work within our theoretical framework. For the sake of simplicity, we take the stance that whenever the unit suggests Robbie a plan, all the actions to perform are either derivable from the theory, or considered as additional facts.

$$
\begin{array}{lll}
\mathrm{F} = \{ & \multicolumn{2}{l}{\mathit{Robbie\_downtown}, \mathit{mail\_delivered}, \mathit{early\_meeting}, \mathit{lunch\_time}\},} \\
R = \{ & \multicolumn{2}{l}{r_1 : \mathit{lunch\_time} \Rightarrow_{\mathrm{U}} \mathit{restaurant} \odot \mathit{take\_away} \odot \mathit{sandwich} \odot \mathit{fast\_food},} \\
 & \multicolumn{2}{l}{r_2 : \mathit{Robbie\_downtown}, \mathit{mail\_delivered} \Rightarrow_{\mathrm{U}} \mathit{oil\text{-}beer},} \\
 & \multicolumn{2}{l}{r_3 : \mathit{Robbie\_downtown}, \mathit{dry\_cleaner}, \mathit{take\_away} \Rightarrow \mathit{speed},} \\
 & r_4 : \Rightarrow_{\mathrm{O}} \neg \mathit{speed}, & r_5 : \mathit{early\_meeting} \Rightarrow_{\mathrm{U}} \mathit{best\_suit} \odot \mathit{2nd\_suit}, \\
 & r_6 : \mathit{take\_away} \Rightarrow \neg \mathit{fast\_food}, & r_7 : \mathit{fast\_food} \Rightarrow \neg \mathit{take\_away}, \\
 & r_8 : \mathit{best\_suit} \Rightarrow \mathit{dry\_cleaner}, & r_9 : \mathit{best\_suit} \Rightarrow \neg \mathit{2nd\_suit}, \\
 & r_{10} : \mathit{2nd\_suit} \Rightarrow \mathit{go\_home}, & r_{11} : \mathit{go\_home}, \mathit{fast\_food} \Rightarrow \neg \mathit{oil\text{-}beer}, \\
 & r_{12} : \mathit{early\_meeting} \Rightarrow \neg \mathit{restaurant}\}. &
\end{array}
$$

If Robbie decides to go to the take away and to give Carmen her best suit, then it must stop to the dry cleaner (r_8); this course of action (r_3) violates r_4. If Robbie decides to pick up the Carmen's second favourite suit from home (r_{10}) and then stop to a fast food, it will not get an oil-beer (r_{11}). In the first scenario, we are not compliant with respect to the norm stated by r_4, while in the second scenario we are not outcome-compliant since Robbie would not be able to derive as belief the conclusion of r_2.

3 Revision under Compliance

Norm and outcome compliance give rise to a non-trivial question: what to be done when a business process is not norm compliant or outcome compliant, or even both? To the best of our knowledge, no *effective approach* addresses the issue of how to revise non-compliant business processes.

We take on this challenge by taking inspiration from *business process revision*, which has received great attention in recent years given its crucial influence, for example, on organisation practices. The aim of the present section is twofold:

(i) to give a critical overview of the state of the art on business process revision and justify the proposed approach taking compliance into consideration;
(ii) to devise a methodology based on the logic proposed in Section 2 to afford revision under compliance.

Roughly speaking, all the efforts spent in the area of business process revision subscribe to two general approaches.

The first approach relies on modelling notations and languages which define the structural aspect of business processes and are extended with other formalisms to represent the behavioural aspects. As an example, BPMN enriched with semantic annotations is able to describe the effects implied by the execution of a particular task [7]. On the same grounds, several translations from modelling notations into other formalisms have been proposed, for example semantic nets [15] and business process graphs [16].

The second approach is instead based on pure logic formalisms, where revising a business process means revising the theory describing the business process itself. The underlying logical theory formally represents at the same time the structural and the behavioural aspects of the business process [17,18,19].

These two approaches capture different (and both interesting) meanings of compliance. The first aims at revising a business process at an higher level, in terms of removal, addition, swapping, and substitutions of tasks in the business process. On the other hand, the second one abstracts from the concepts of task and conditions that trigger (or are caused by) a task: they are all denoted by literals in the same theory and the main focus is on how they interact with each other to obtain other literals.

The most representative example of the first approach, given in [20] and then developed in [21], relies on the emerging trend of designing and thinking about business processes as related collections of *reusable modules* (or *fragments*). Reusable modules denote sets of standardised actions to be performed to achieve some fixed objectives (giving outcome compliance) that can be used with slight or no modifications also in other business processes. Modules are further augmented with built-in statements ensuring that the module is norm compliant according to the statements specified in the module.

The reusable modules approach is theoretically applicable both when a norm uncompliant process is given, or when it must be built from scratch and we have to ensure norm and outcome compliance at design time. For the first case, the algorithm for norm compliance checking proposed in [7] finds out the *exact* point in a business process where a violation of an obligation occurs. Thus, we can substitute the un-compliant part of the business process with a module that reaches the same objectives and compensates the previous violation(s). In the second case, we build the process starting from a given repository of modules, based on objectives to achieve and norms to comply with.

For many aspects, the reusable module approach recall the SOC paradigm that "promotes the idea of assembling application components into a network of services that can be loosely coupled to create flexible, dynamic business processes and agile applications" [22]. As such, the main advantage of this approach is the possibility of exploiting well defined techniques and methodologies developed through years of investigation in the field. Above all, *Web Service technology* is nowadays the most promising means for a widespread deployment of SOC-based architectures in on-the-net business process software development [23].

However, this approach is not free from drawbacks, especially in case of compliance recovery of a given business process. For example, the addition or substitution of a fragment in a business process may lead to a lack of resources, jeopardising the entire execution of the process. In fact, reusable fragments are intrinsically bound to a *localised* concept of compliance. Every fragment introduces potentially new effects in the business process and, consequently, other obligations or outcomes may be triggered. Thus, once a fragment has been added or replaced, there is no guarantee that the business process is *globally* compliant (with respect to norms and outcomes). Notice that the issue can be avoided in some cases when, for instance, the effects attached to executed modules are independent, or the execution of modules is mutually exclusive (when modules belong to different branches of XOR paths). Nevertheless, these cases represent a very limited part of the whole behavioural sphere of organisational practices.

Apart from the difficulties brought by the first family of approaches in managing compliance, a fully logic approach (like the one herein proposed) results much more appealing due to several factors:

(i) the approach of revision based on module reuse is quite hypothetical at this moment, since no formal investigation has been yet carried out that looks at the possible ways to provide a correct and complete solution;

(ii) a logic approach always comes with an inferential mechanism to reason about the underlying framework and to prove the computational properties like consistency and coherency;

(iii) it is particularly simple to map many different logic approaches existing in the current literature of the field into *graph models* and this is a promise for simplicity in the case we are dealing with too;

(iv) a roadmap that can be considered useful aims at building real applications, and one possible target is a system that modifies a business process based upon preferences specified by the user. In this scenario, it is meaningful to provide a logic approach that extends execution environments (for example *Deimos* [24] and *SPINdle* [25]) where the schema is useful for deploying real systems.

3.1 Revision via Proof Tags Analysis

Rule-based nonmonotonic formalisms – like the one developed in this work – have been used in many types of reasoning given their predisposition to capture aspects from many domains of interest. As an example, in [26] authors address the problem of revising non-modal defeasible theories in the area of legal reasoning through an exhaustive analysis of proof tags. The major contribution of that work is the identification of three relevant cases, named *canonical*, where a revision operator could act by only changing the relative strength between pairs of rules.

First case. The revision operator acts on a defeasibly proved literal p and makes it not provable anymore, i.e., from $+\partial p$ to $-\partial p$;

Second case. The revision operator acts on a defeasibly proved literal p and makes its opposite defeasibly proved, i.e., from $+\partial p$ to $+\partial{\sim}p$;

Third case. The revision operator acts on a not defeasibly proved literal p and makes it defeasibly proved, i.e., from $-\partial p$ to $+\partial p$.

Additional proof tags other than those for classical defeasible proof ($\pm\partial$) are used to better identify all relevant situations. The new proof tags do not modify the expressive power of the logic itself, but they identify specific properties of literals within the theory, e.g., their reachability or being always derived.

There are three major reasons for using and extending the approach of proof tag analysis also in the framework at hand.

First, canonical cases clearly refer to the basic issue we address. For example, by definition of $+\partial_{\mathrm{O}}\bot$, there exists at least one applicable rule r for obligations such that each element c_i defeasibly proved as an obligation in the chain is not derived as a belief, i.e., if $+\partial_{\mathrm{O}}c_i$ holds, then $-\partial c_i$ holds as well. As this chain represents the consequent of a rule, there are two strategies to recover norm compliance with respect to r:

(i) By making one element c_i defeasibly provable as a belief, or
(ii) Acting on r, either by making the rule body-discarded in the sense of Definition 1, or by blocking the derivation of the first element in the chain as an obligation.

Case (i) represents a modal variant of the third canonical case, that is from $-\partial_{\Box}p$ to $+\partial_{\Box}p$, while case (ii) represent a modal variant of the first canonical case, that is from $+\partial_{\Box}p$ to $-\partial_{\Box}p$. Notice that in most cases, the first approach reflects a more intuitive way to address the problem. Indeed, it seems more reasonable to find a way to compensate an obligation (or the failure to achieve an outcome) rather than to make discarded the entire chain of obligations or outcomes.

Second, it is quite simple to align the additional proof tags introduced in [26] with a modal defeasible theory. As an example, the tagged literal $+\Sigma p$ means that there exists a reasoning chain supporting literal p. A more fine-grained definition of support can be given in our framework, that is, when a literal p is supported in a modal defeasible theory with modality $\Box$ ($+\Sigma_{\Box}p$). An analogous reasoning can be made for an unsupported literal p ($-\Sigma_{\Box}p$).

$+\Sigma_{\Box}$: If $P(n+1) = +\Sigma_{\Box}q$ then
(1) $\Box q \in F$ or
(2) $\exists r \in R^{\Box}[q,i]$ such that $\forall a \in A(r)$, $+\Sigma_{\mathrm{B}}a \in P(1..n)$, and
$\forall \blacksquare b \in A(r)$, $+\Sigma_{\blacksquare}b \in P(1..n)$, or
(3) $\exists r \in R^{\mathrm{B}}[q]$ such that $\forall a \in A(r)$, $+\Sigma_{\Box}a \in P(1..n)$

Third, the alignment of proof tags in a modal defeasible theory allows us to formally represent the conditions under which a revision operator could return a positive solution of the problem. To give an overall idea, we report in Figure 1 a conceptual tree representing the possible cases a revision operator could deal with in our framework.

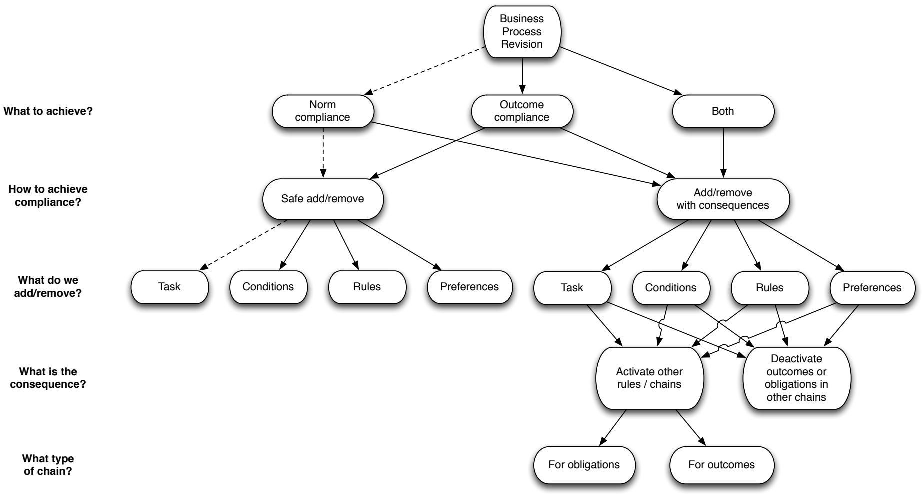

Fig. 1: Possible problems for a business process revision operator.

The schema is organised in levels, and every level sets up the parameters that the revision operator must obey to.

The first level represents the type of compliance to be fulfilled; in the first two cases, we take the stance that a norm un-compliant business process is already outcome compliant, and the other way around.

The second level defines the possible actions the revision operator can perform in order to meet compliance. In our case, removal or addition of elements in the theory are the only feasible actions. At first glance, the combination of addition and removal could represent a potential candidate to define the operation of *swapping*: for example, to change the order between tasks t_1 and t_2, we just remove t_1 (resp. t_2) and then add it after t_2 (resp. before t_1).

The third level defines the elements (literals for tasks, literals representing conditions, rules, or preference relations between rules in the theory) the revision operator can manipulate based on the type of actions permitted. We define an addition or a removal *safe* if the operation does not make applicable any previously discarded rule in the theory, and does not deactivate obligations or outcomes previously in force. Otherwise, we define the operation *unsafe* or *with consequences*. In this case, we discriminate between the two types of consequences (fourth level), and in the first case, we specify which type of chain is activated (fifth level).

Intuitively, every path from the root to a leaf denotes a particular type of problem to be solved. For example, the dotted path in Figure 1 represents the problem of finding conditions under which we can revise a business process to recover norm compliance, remaining outcome compliant, only through safe addition or removal of tasks.

The schema aims at providing a combinatorial exploitation of the cases for a revision operator for business processes with compliance preservation. This analysis can be valued as exhaustive provided that we do not consider neither redundancy nor unfeasibility. These further aspects will be considered in the next step of this research.

Here follow some intuitive ideas, based on [27], on how to address the first (and second) canonical case by using rule addition/removal and preference manipulation.

Rule removal and exception addition. As the name suggests, the operation of *rule removal* contracts a literal p by simply removing rules in the theory directly deriving it. One of the proposed variants acts only on defeasible rules, and, unlike other variants, it has to take care of preserving minimality (in the sense of number of erased rules). The intuition here is to identify all rules that are essential to prove the literal at hand. This corresponds to remove only the applicable rules that are not inferiorly defeated, i.e., rules such that there is no stronger applicable rule for the opposite. Several variants of *exception addition* were discussed as well, and one of them consider only defeasible rules. In this case, the revision operator introduces a new conflicting defeasible rule for the opposite conclusion to block the conclusion we want to contract.

Preference addition/removal. This kind of contraction acts on the relative strength of rules, by making two complementary arguments of the same strength (and then both conclusions cannot be derived), or by making the opposite argument stronger than the argument for the literal to contract. When we deal with changing a theory by only modifying the superiority relation, we must switch perspective with respect to the operations of rule removal/addition and exception addition. The operations cannot anymore focus only on active/inactive chains for the literal we want to change, but they must take into account the entire theory. Moreover, one strong problematic that affects the contraction operation by modifying the superiority relation only is on the hardness of defining when a literal is absolutely unrefutable. Roughly speaking, a literal is absolutely unrefutable if it is true in every interpretation (and then we cannot contract it). We found examples on the structure of a theory where a non factual literal is absolutely unrefutable, and we proved that the problem of deciding if a literal is absolutely unrefutable is NP-hard. The proof is not reported in the present work for space reasons.

4 Conclusions and Further Work

This paper introduced a logical framework to represent objective-driven agents which deliberates her course of action based on constraints imposed by the environment. The logic deems the notions of desires, goals and intentions (and social intentions) as different nuances of the (more general) notion of outcome. The framework naturally describes and integrates the notion of being compliant with the notions of norms and outcomes. This allowed us to describe situations where the agent is able (or not) to reach all her objectives while not violating the norms. In addition, we proposed a preliminary analysis on how to devise a methodology to restore from un-compliant situations, covering the case of outcomes or norms. The analysis carried out in the previous sections points out several directions of research. We report the most valuable, referring to the case analysis reported in Figure 1.

Does a solution exist for each case? As already pointed out, one prior challenge to address is to study conditions under which the revision operator returns a positive answer to the case at hand. In this sense, proof tag analysis seems a good candidate for defining properties that describe the status of every literal in the theory. Moreover, this approach is compatible with a fully logical representation of business processes.

If a solution exists, what is the complexity of the problem instance? Once having established that at least a solution exists, another issue is to find algorithmic means that compute a particular (or all) solution(s). This is crucial for a better decision-making process with a view to automatic deployment. In fact, algorithms based on the logical system herein proposed could extend well-known execution environments, and that is a promise for the revision process to become a proper extension of the process for compliance checking, and could be fully integrated in the business process life-cycle.

Which is the minimum set of actions to obtain a solution? The trivial case is that a solution does not exist because the set of actions performed is too restrained. In this case, an interesting problem is to determine which action (or actions) the revision operator should be allowed to use in addition to solve the problem. In other terms, we define the revision operator as an entity with several *degrees of freedom* and we determine which is the minimum degree of freedom of the operator to solve each case.

In particular, the second problem deserves particular attention. Indeed, by its own nature a rule in our framework may have multiple antecedents, and a chain for obligations and outcomes with more than one element as a consequent. In the view of revision, a crucial aspect is defining the best point where to act during the process, be it in the antecedent or the consequent. This question naturally resembles the meaning of *minimal change* as a key tenet for the revision to be rational [28]. However, several criteria of minimality can be chosen, such as the degree of change in the extension of the theory, or based on the number of actions to be performed by the revision operator. In this respect, a method that exploits the concept of *literal dependency* could be useful. A literal l depends on another literal m if m appears in every reasoning chain supporting l. We are working on a complete formalisation of dependency property, and an algorithmic implementation to find out the dependency set of a literal in a theory is currently under development. Intuitively, if the revision operator decides to refute a (previously proved) literal l to be recover compliance, then in principle refuting a literal m on which l depends (if any) minimises the changes in the extension.

References

1. Wooldridge, M., Jennings, N.R.: Agent theories, architectures, and languages: A survey. In Wooldridge, M., Jennings, N.R., eds.: ECAI Workshop on Agent Theories, Architectures, and Languages. Volume 890 of LNCS., Springer (1995)
2. Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. Artificial Intelligence **42**(2-3) (1990)
3. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In Allen, J.F., Fikes, R., Sandewall, E., eds.: KR, Kaufmann, M. (1991)
4. Rao, A.S., Georgeff, M.P.: Decision procedures for bdi logics. Journal of Logic and Computation **8**(3) (1998)
5. van der Aalst, W.M.P.: The application of Petri Nets to workflow management. Journal of Circuits, Systems, and Computers **8**(1) (1998)
6. Ghallab, M., Nau, D., Traverso, P.: Automated planning - theory and practice. Elsevier (2004)

7. Governatori, G., Sadiq, S.: The journey to business process compliance. In: Handbook of Research on BPM. IGI Global (2008)
8. Governatori, G., Olivieri, F., Scannapieco, S., Cristani, M.: Designing for compliance: Norms and goals. In Olken, F., Palmirani, M., Sottara, D., eds.: RuleML America. Volume 7018 of Lecture Notes in Computer Science., Springer (2011)
9. Governatori, G., Olivieri, F., Rotolo, A., Scannapieco, S., Cristani, M.: Picking up the best goal: An analytical study in defeasible logic. In Paschke, A., Morgenstern, L., Stefaneas, P., eds.: RuleML. Volume 8035 of LNCS., Springer (2013)
10. Antoniou, G., Billington, D., Governatori, G., Maher, M.J.: Representation results for defeasible logic. ACM Transactions on Computational Logic **2**(2) (2001)
11. Governatori, G., Rotolo, A.: Logic of violations: A gentzen system for reasoning with contrary-to-duty obligations. Australasian Journal of Logic **4** (2006)
12. Governatori, G., Rotolo, A.: BIO logical agents: Norms, beliefs, intentions in defeasible logic. Journal of Autonomous Agents and Multi-Agent Systems **17**(1) (2008)
13. Broersen, J., Dastani, M., Hulstijn, J., van der Torre, L.: Goal generation in the BOID architecture. Cognitive Science Quarterly **2**(3-4) (2002)
14. Antoniou, G., Billington, D., Governatori, G., Maher, M.J., Rock, A.: A family of defeasible reasoning logics and its implementation. In Horn, W., ed.: ECAI, IOS Press (2000)
15. Ghose, A., Koliadis, G.: Auditing business process compliance. In Krämer, B., Lin, K.J., Narasimhan, P., eds.: ICSOC. Volume 4749 of LNCS., Springer (2007) 169–180
16. Dijkman, R., Dumas, M., van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Inf. Syst. **36**(2) (2011)
17. Governatori, G., Rotolo, A.: Norm compliance in business process modeling. [29] 194–209
18. van der Aalst, W., Pesic, M., Schonenberg, H.: Declarative workflows: Balancing between flexibility and support. Computer Science - R&D **23**(2) (2009)
19. Rotolo, A.: Rule-based agents, compliance, and intention reconsideration in defeasible logic. In Bassiliades, N., Governatori, G., Paschke, A., eds.: RuleML Europe. Volume 6826 of Lecture Notes in Computer Science., Springer (2011)
20. Schumm, D., Leymann, F., Ma, Z., Scheibler, T., Strauch, S. In: Integrating Compliance into Business Processes Process Fragments as Reusable Compliance Controls. Universitätsverlag Göttingen (2010)
21. Schumm, D., Türetken, O., Kokash, N., Elgammal, A., Leymann, F., van den Heuvel, W.J.: Business process compliance through reusable units of compliant processes. In Daniel, F., Facca, F., eds.: ICWE Workshops. Volume 6385 of LNCS., Springer (2010)
22. Leymann, F.: Combining web services and the grid: Towards adaptive enterprise applications. In Castro, J., Teniente, E., eds.: CAiSE Workshops (2), FEUP Edições, Porto (2005)
23. Weerawarana, S., Curbera, F., Leymann, F., Storey, T., Ferguson, D.: Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More. Prentice Hall PTR (2005)
24. Rock, A.: Deimos: Query answering defeasible logic system (2000)
25. Lam, H.P., Governatori, G.: The making of SPINdle. In: Rule Representation, Interchange and Reasoning on the Web. Number 5858, Springer (2009)
26. Governatori, G., Olivieri, F., Scannapieco, S., Cristani, M.: Superiority based revision of defeasible theories. [29] 104–118
27. Governatori, G., Olivieri, F., Rotolo, A., Scannapieco, S.: Legal contractions: A logical analysis. In Ashley, K.D., van Engers, T.M., eds.: ICAIL, ACM (2013)
28. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. J. Symb. Log. **50**(2) (1985)
29. Dean, M., Hall, J., Rotolo, A., Tabet, S., eds.: Semantic Web Rules - International Symposium, RuleML 2010, Washington, DC, USA. In Dean, M., Hall, J., Rotolo, A., Tabet, S., eds.: RuleML. Volume 6403 of LNCS., Springer (2010)