A Logic Framework of Normative-based Contract Management

Guido Governatori

NICTA, Australia, guido.governatori@nicta.com.au

Abstract. In this paper an extended Defeasible Logic framework is presented to do the representation and reasoning work for the normative-based contract management. A simple case based on FIDIC is followed as the usage example. This paper is based on the idea that normative concepts and normative rules should play the decisive roles in the normative-based contract management. Those normative concepts and rules are based on the normative literals and operators like action, obligation, permission and violation. The normative reduction is based on the normative concepts, normative connections and normative rules, especially on the superiority relation over the defeasible rules.

1 Introduction

Contract Management is one of the most important aspects in business management because most business activities are carried out in accordance with contracts and various other regulations. Generally speaking, Contract Management covers many disciplines such as Baseline Management, Commitment Management and Transaction Compliance Management. As a result it plays a key role in the business life. Traditionally, Contract Management has been carried out based on the working experience of professionals. More recently computer aided software has been introduced into this area, however such software is still based on professional experience and lacks the core ability of reasoning.

The business world is a huge correlated network composed of many independent companies which form the nodes of the network. Each company in this network can be treated as an autonomous intelligent agent and the whole network as a multi-agent system (MAS). In this MAS, every company has certain rights, responsibilities, permissions and prohibitions. Additionally the company being autonomous can choose to obey the rules, as well as sometimes elect to break the rules and act in breach of obligations. Specifically, the Contract Management could be treated as a multi-agent system as well. In this MAS a typical business process generally consists of a number of interrelated activities jointly conducted by different autonomous business companies. All the activities in the business process are mainly directed, performed, judged and evaluated by corresponding contracts. Furthermore, the business contracts and business processes are restrained by and should conform to the general laws of society and corresponding industrial regulations. Most notably the common laws, industrial regulations and international conventions.

This brings to our attention that normative concepts such as obligation and right, permission and prohibition, as well as abidance and violation are playing a decisive role

Fourth International Workshop on Juris-informatics (JURISIN 2010)
Satoshi Tojo (ed)

in the MAS of Contract Management. Simply put, if everything in the business process follows the way stipulated in the contract without violation, an ideal situation and result would obviously be attained. However in the event of a violation, corresponding remedy or punishment procedures could then follow. Hence, in theory, the possible corresponding results of agents' behavior and business processes should be predictable and the result should fall into a limited or unlimited results set within some certain boundary. At the same time the preconditions of some possible or existing results should be identifiable and henceforth set some kind of precedents as well according to the related normative rules. It means that the logical modeling of institutionalised agencies could be used in the Contract Management.

In this paper we combine the defeasible logic of institutionalised agency [6,5,7] with the defeasible logic of temporalised normative positions [10]. The resulting logic offers a computationally oriented (non-monotonic) formal framework for the representation of normative-based contract management. In particular the logic offers facilities to represent abstract agency (implementing a "brings it about" modal operator), counts-as conditionals, and normative conditionals for obligations, permissions, prohibitions and violations. The temporal extension allows us to reason with deadlines as well as the initiation and termination of normative positions.

The focus of the paper is to demonstrate that the proposed logical framework is capable to represent and to implement real life contract management standards. In particular we propose a logical encoding of one clause of the FIDIC regulation: Clause 67 about the rules of the disputes and the corresponding procedures of solving disputes. The encoding provides an executable specification of the clause, and the resulting set of rules can be executed by an appropriate defeasible logic rule engine [12,9].

2 Logical Framework

The logical framework is based on Defeasible Logic (DL). Defeasible Logic is a rule based non-monotonic formalism. A rule in DL corresponds to a relationships between a set of premises (literals) and a conclusion (literal). In Defeasible Logic we have three types of rules: strict rules, which are rules in the classical sense, $a_1,\dots,a_n \to b$, meaning that b is a definite/indisputable conclusion of $a_1,\dots,a_n$ when these are indisputable; defeasible rules, $a_1,\dots,a_n \Rightarrow b$, meaning that usually b is a conclusion of $a_1,\dots,a_n$ unless there is some stronger evidence against it. Finally we have defeaters. Defeaters are the weakest relationships between a set of premises and a conclusions, a defeater like $a_1,\dots,a_n \leadsto b$, is can be understood as a reason to prevent the derivation of $\neg b$ from $a_1,\dots,a_n$, but not to support the conclusion of b from the same set of premises.

Beside the above classification of rule based on their strength, we have a second classification based on the type of conclusion (mode) we have: in this paper we consider "counts-as" rules (which will also be used as the basic inferential mechanism of the logic), "results-in" rules, and "obligation" rules. Defeaters have a special meaning for obligation rules, instead of creating a new obligation, they terminate an existing one.

To capture the temporal dimension, we introduce (1) temporalised literals, e.g., $a : t$ meaning that a holds at time t where t is a timestamp, and (2) for each rule we specify whether the conclusion is transient, i.e., it holds only for a particular instant, or permanent, i.e., it continues to hold until a terminating event occurs.

Based on the above discussion we can have rules like:

$$a_1 : t_1, \ldots a_n : t_n \Rightarrow_{C}^{per} b : t \tag{1}$$

$$a_1 : t_1, \ldots a_n : t_n \Rightarrow_{E_j}^{tr} b : t \tag{2}$$

$$a_1 : t_1, \ldots a_n : t_n \Rightarrow_{O_j}^{per} b : t \tag{3}$$

$$a_1 : t_1, \ldots a_n : t_n \rightsquigarrow_{O_j}^{per} b : t \tag{4}$$

Here (1), a counts-as rule, states that in the context of the FIDIC contract under analysis if a_i holds at time t_i (for all i, $1 \leq i \leq n$), then we are allowed to assert that b holds at time t, and, since the rule is labelled as persistent, the effect b will hold after t until some event eventually terminates the validity of b. For (2), a results-in rule, the meaning is that if a_i holds at time t_i (for all i, $1 \leq i \leq n$), then we are allowed to assert that agent j does b at time t (or j brings is about that b at time t, $E_j b : t$). However, this is a transient rule, so we cannot assert that j achieved b at a time t', $t' > t$. Rule (3) is an obligation rule that specifies that agent j has the obligation b at time t ($O_j b : t$), when the conditions a_i to a_n are satisfied at the appropriate times, and that the obligation persists, until a discharging event occurs. Finally, (4) terminates the obligation on $\neg b$ (prohibition of b) at time t. For a full presentation of the logics see [7,10,6,8].

An important aspect of the resulting logic is the ability to represent temporal intervals, thus for example, the pair of rules

$$a_1 : t_1, \ldots, a_n : t_n \Rightarrow_{X}^{per} b : t \tag{5}$$

$$a_1 : t_1, \ldots, a_n : t_n \rightsquigarrow_{X}^{tr} \neg b : t' \tag{6}$$

establish, that, given the premises of the rules and $t < t'$, b holds between t and t'. Thus, we use the shorthand, and we write

$$a_1 : t_1, \ldots, a_n : t_n \Rightarrow_{X} b[t, t'] \tag{7}$$

to denote the above two rules. This structure is the structure defining rules for [3].

3 FIDIC Contract Management in DL

FIDIC is the acronym of International Federation of Consulting Engineers (the abbreviation represents the French version of the name). Founded in 1913 FIDIC has now become an international federation of national associations of consulting Engineers including 67 Member Associations from all parts of the world. The series of standard contract forms stipulated by FIDIC has become international standards in project management, including civil, mechanical, electricity, medicine engineering projects.[1]

First of all, some general background knowledge needs to be encoded for the FIDIC contract management using this DL framework. To begin with there are four agent literals: *Employer*, *Contractor*, *Engineer* and *Arbitrator*. Several atomic status literals

[1] for a comprehensive survey, please check `http://www.fidic.org` for more details.

are needed to label the milestones of a dispute process: *StartOfProject*, *BeginOfDisp*, *EndOfDisp*, *NotiOfDeciOfEngi*, *BeginOfArbi*,
StartArbitrating, *EndOfArbitration*, *AmicableEndOfDisp*. A set of atomic discrete ordered time point literals are also provided $\{t_1, t_2, t_3, \ldots, t_n, \ldots\}$.

A FIDIC contract comprises two set of clauses: *General Clauses* and *Specific Clauses*. The general clauses are meant to define and restrict basic goals, general relationships and basic behaviour of the agents involved in a project. The specific clauses stipulate the details of goals, relationships and behavior standards of Agents in that certain project. In this paper we focus on the modelling of the general clauses for management of a FIDIC contract.

The *GeneralClauses* of the *FIDICContract* as far as we are concerned can be summarised in the following statements. From the start of a project or contract, the *Employer* has the obligation to provide the funds, the working area and other necessary facilities as well as undertake other necessary work as specified in the *FIDICContract* to the *Engineer* and the $Contractor$. The $Contractor$ has the obligation to do the work and finish the project as specified in the *FIDICContract*. The *Engineer* has the obligation to supervise the work of the $Contractor$ and the $Employer$ as specified in the *FIDICContract*. Together with the obligations, all the Agents have the right or the privilege to get the production or the payment from the counterpart in return for their hard work as specified in the *FIDICContract*. (Note: to save space in the following paper *em* for $Employer$, *co* for $Contractor$, *en* for *Engineer* and *ar* for $Arbitrator$)

$G_{1.1}$: $StartOfProject : t_{start} \Rightarrow_{O_{em}} E_{co}Work(project) : [t_{start}, t_{end}]$
$G_{1.2}$: $StartOfProject : t_{start} \Rightarrow_{O_{em}} E_{en}Supervise(project) : [t_{start}, t_{end}]$
$G_{1.3}$: $StartOfProject : t_{start} \Rightarrow_{O_{co}} E_{em}WorkPay(project, co) : [t_{start}, t_{end}]$
$G_{1.4}$: $StartOfProject : t_{start} \Rightarrow_{O_{en}} E_{em}WorkPay(project, en) : [t_{start}, t_{end}]$

The specific clauses then define the meaning of the terms used in the general clauses. For example, they can specify what it means to work in a project for a contractor ($E_{co}Work(project)$).

In the rest of the paper we concentrate on Clause 67 of FIDIC concerning procedure for dispute and dispute resolution.

3.1 Clause 67 of FIDIC

In the following we report the whole content of the Clause 67 of FIDIC [2].

67.1 Engineers decision If a dispute of any kind whatsoever arises between the Employer and the Contractor in connection with, or arising out of, the contract or the execution of the Works, whether during the execution of the Works or after their completion and whether before or after repudiation or other termination of the Contract, including any dispute as to any opinion, instruction, determination, certificate or valuation of the Engineer, the matter in dispute shall, in the first place, be referred in writing to the Engineer, with a copy to the other party. Such reference shall state that it is made pursuant to this Clause. No later than the eighty-fourth day after the day on which he received such reference the Engineer shall give notice of his decision to the Employer and the Contractor. Such decision shall state that it is made pursuant to this Clause.

Unless the Contract has already been repudiated or terminated, the Contractor shall, in every case, continue to proceed with the Works with all due diligence and the Contractor and the Employer shall give effect forthwith to every such decision of the Engineer unless and until the same shall be revised, as hereinafter provided, in an amicable settlement or an arbitral award.

If either the Employer or the Contractor are dissatisfied with any decision of the Engineer, or if the Engineer fails to give notice of his decision on or before the eighty-fourth day after the day on which he received the reference, then either the Employer or the Contractor may, on or before the seventieth day after the day on which he received notice of such decision, or on or before the seventieth day after the day on which the said period of 84 days expired, as the case may be give notice to the other party, with a copy for information to the Engineer, of his intention to commence arbitration, as hereinafter provided, as to the matter in dispute. Such notice shall establish the entitlement of the party giving the same to commence arbitration, as hereinafter provided, as to such dispute and, subject to Sub-Clause 67.4, no arbitration in respect thereof may be commenced unless such notice is given.

If the Engineer has given notice of his decision as to a matter in dispute to the Employer and the Contractor and no notice of intention to commence arbitration as to such dispute has been given by either the Employer or the Contractor on or before the seventieth day after the day on which the parties received notice as to such decision from the Engineer, the said decision shall become final and binding upon the Employer and the Contractor.

67.2 Amicable settlement Where notice of intention to commence arbitration as to a dispute has been given in accordance with Sub-Clause 67.1, arbitration of such dispute shall not be commenced unless an attempt has first been made by the parties to settle such dispute amicably. Provided that, unless the parties otherwise agree, arbitration may be commenced on or after the fifty-sixth day after the day on which notice of intention to commence arbitration of such dispute was given, whether or not any attempt at amicable settlement thereof has been made.

67.3 Arbitration Any dispute with respect of which:

(a) the decision, if any, of the Engineer has not become final and binding pursuant to Sub-Clause 67.1 and

(b) amicable settlement has not been reached within the period stated in Sub-Clause 67.2

shall be finally settled, unless otherwise specified in the contract, under the rules of conciliation and arbitration of the International Chamber of Commerce by one or more arbitrators appointed under such rules. The said arbitrator/s shall have full power to open up, review and revise any decision, opinion, instruction, determination, certificate or valuation of the Engineer related to the dispute.

Neither party shall be limited in the proceedings before such arbitrator/s to the evidence or arguments put before the Engineer for the purpose of obtaining his said decision pursuant to Sub-Clause 67.1. No such decision shall disqualify the Engineer from being called as a witness and giving evidence before the arbitrator/s on any matter whatsoever relevant to the dispute.

Arbitration may be commenced prior to or after completion of the works, provided that the obligations of the Employer, the Engineer and the Contractor shall not be altered by reason of the arbitration being conducted during the progress of the works.

67.4 Failure to comply with Engineers decision Where neither the Employer nor the Contractor has given notice of intention to commence arbitration of a dispute within the period stated in Sub-Clause 67.1 and the related decision has become final and binding, either party may, if the other party fails to comply with such decision, and without prejudice to any other rights it may have, refer the failure to arbitration in accordance with Sub-Clause 67.3, the provisions of Sub-Clause s 67.1 and 67.2 shall not apply to any such reference.

3.2 Encoding Clause 67 of FIDIC using DL framework framework

Generally this clause introduces a three-stage process for disputes-solving in the FIDIC contract management. Herein we will use the DL framework to represent this knowledge base. For simplicity we constrain all the business processes and rules from the time of t_{start} to t_{end}. In fact FIDIC's power extends even after the t_{end} and that situation can be dealt with in a similar fashion.

Engineers decision In the first stage, the *Employer* or the *Contractor* should send the dispute to the *Engineer* in writing. Then the *Engineer* has at most 12 weeks to consider this referred dispute and draw his decision towards it. In this period both parties of the *Employer* and the *Contractor* should go on to obey the *Engineer* original specifications as defined in some specific clauses of the contract.

Several actions needed to be set here.

- Action A_1: (the *Employer* or the *Contractor*) to refer a dispute to the Engineer.
- Action A_2: (the *Engineer*) to draw a notice of decision.

The rules describing this cases are

$$R_{1.1.1}\text{: } E_{em}A_1 : t_1 \Rightarrow^{tr}_{C} \mathit{BeginOfDisp} : t_1$$
$$R_{1.1.2}\text{: } E_{co}A_1 : t_1 \Rightarrow^{tr}_{C} \mathit{BeginOfDisp} : t_1$$
$$R_{1.2.0}\text{: } \mathit{BeginOfDisp} : t_1 \Rightarrow_{O_{en}} E_{en}A_2 : [t_1, t_1 + 84]$$
$$R_{1.3.1}\text{: } E_{en}A_2 : t_2, t_2 \in [t_1, t_1 + 84] \Rightarrow^{tr}_{C} \mathit{NotiOfDeciOfEngi} : t_2$$
$$R_{1.3.2}\text{: } E_{en}A_2 : t_2, t_2 \in [t_1, t_1 + 84] \Rightarrow^{per}_{O_{em,co}} \neg E_{en}A_2 : t_2$$

Notice that $R_{1.3.2}$ means that if one agent follows her obligation successfully at time point t then she is discharged of the finished obligation at time point t' after t.

After the receipt of the *Engineer*'s decision about the referred dispute, if the $Employer$ or the $Contractor$ is dissatisfied with the decision of the *Engineer*, or if the *Engineer* fails to make a decision towards the referred dispute, both parties are entitled to give notice of their intention to commence the arbitration in 10 weeks, in the first situation from the date of receipt of the Engineer's decision, or in the second situation in 10 weeks from the date in which the 84 days subsequent to the *Engineer*'s decision expired. If no such notices are put forward, the *Engineer*'s decision will become the final decision and both parties of the $Employer$ and the $Contractor$ are bound by this final decision. Which in effect means this dispute ends.

- Action A_3: give the $NoticeOfIntentionToCommenceArbitration$

The rules concerning this action are:

$$
\begin{aligned}
&R_{1.4.0}\text{: } \mathit{NotiOfDeciOfEngi} : t_2, \neg E_{em}A_3 : t_2 + 70, \neg E_{co}A_3 : t_2 + 70 \Rightarrow^{pr}_{C} \mathit{EndOfDisp} : t_2 + 71\\
&R_{1.5.1}\text{: } \mathit{NotiOfDeciOfEngi} : t_2, E_{em}A_3 : t_3, t_3 \in [t_2, t_2 + 70] \Rightarrow^{tr}_{C} \mathit{BeginOfArbi} : t_3\\
&R_{1.5.2}\text{: } \mathit{NotiOfDeciOfEngi} : t_2, E_{co}A_3 : t_3, t_3 \in [t_2, t_2 + 70] \Rightarrow^{tr}_{C} \mathit{BeginOfArbi} : t_3\\
&R_{1.6.1}\text{: } \neg \mathit{NotiOfDeciOfEngi} : t_2, E_{em}A_3 : t_3, t_3 \in [t_1 + 84, t_1 + 154] \Rightarrow^{tr}_{C} \mathit{BeginOfArbi} : t_3\\
&R_{1.6.2}\text{: } \neg \mathit{NotiOfDeciOfEngi} : t_2, E_{co}A_3 : t_3, t_3 \in [t_1 + 84, t_1 + 154] \Rightarrow^{tr}_{C} \mathit{BeginOfArbi} : t_3
\end{aligned}
$$

Amicable settlement In the second stage both parties, i.e., the $Employer$ and $Contractor$ have the obligation towards each other to try and settle the occurred dispute amicably in 8 weeks after the $BeginOfArbi$. In addition this clause gives us a situation leading to the $StartArbitrating$. We have several actions here.

- Action A_4: ($Employer$ or $Contractor$) to try to amicably settle the referred dispute.
- Action $A5$: ($Employer$ and $Contractor$) to amicably settle the referred dispute.

The rules governing this phase are:

$$
\begin{aligned}
&R_{2.1}\text{: } \mathit{BeginOfArbi} : t_3 \Rightarrow_{O_{em}} E_{co}A4 : [t_3, t_3 + 56]\\
&R_{2.2}\text{: } \mathit{BeginOfArbi} : t_3 \Rightarrow_{O_{co}} E_{em}A4 : [t_3, t_3 + 56]\\
&R_{2.3}\text{: } E_{em,co}A5 : t_5, t_5 \in [t_3, t_3 + 56] \Rightarrow^{per}_{C} \mathit{AmicableEndOfDisp} : t_5\\
&R_{2.4}\text{: } \neg E_{em,co}A5 : t_3 + 56 \Rightarrow^{tr}_{C} \mathit{StartArbitrating} : t_3 + 57 = t_6
\end{aligned}
$$

From the above deontic rule $R_{2.1}$ and $R_{2.2}$ we get the corresponding violation rule $R_{2.1.1}$ and $R_{2.2.1}$. If either of the parties in the dispute does not try to settle the referred dispute amicably, it is a violation against the general rule $G_{1.1}$ and $G_{1.3}$.

The amicable settlement period is the last chance for the both parties of the dispute to settle the disagreement by themselves. After this phase the dispute steps into the arbitration phase.

Arbitration We have several actions in this phase:

- Action $A_{7.1}$: ($Arbitrator$) to do arbitration work about the dispute
- Action $A_{7.1.1}$: ($Arbitrator$) to overrule the Certificate Of Engineer as to the dispute
- Action $A_{7.2}$: ($Employer$ or $Contractor$) to provide evidence as to the dispute
- Action $A_{7.3}$: ($Engineer$) to be the witness as to the dispute

- Action A_8: ($Arbitrator$) to draw a final decision about this dispute

Several rules applies to this phase:

$R_{3.1.1}$: $StartArbitrating : t_6 \Rightarrow^{per}_{O_{em,co}} E_{ar}A_{7.1} : t_6$
$R_{3.1.2}$: $StartArbitrating : t_6 \Rightarrow^{per}_{O_{em,co}} E_{ar}A_8 : t_6$
$R_{3.1.1.1}$: $StartArbitrating : t_6 \Rightarrow^{per}_{C} E_{ar}A_{7.1.1} : t_6$
$R_{3.2.1}$: $E_{ar}A_{7.1.1} : t_7 \Rightarrow^{per}_{O_{em}} \neg E_{co}Follow(CertificateOfEngineer) : t_7$
$R_{3.2.2}$: $E_{ar}A_{7.1.1} : t_7 \Rightarrow^{per}_{O_{co}} \neg E_{em}Follow(CertificateOfEngineer) : t_7$
$R_{3.3}$: $StartArbitrating : t_6 \Rightarrow^{per}_{O_{ar}} E_{em}A_{7.2} : t_6)$
$R_{3.4}$: $StartArbitrating : t_6 \Rightarrow^{per}_{O_{ar}} E_{co}A_{7.2} : t_6$
$R_{3.5}$: $StartArbitrating : t_6 \Rightarrow^{per}_{O_{ar}} E_{en}A_{7.3} : t_6$
$R_{3.6}$: $E_{ar}A_8 : t_8 \Rightarrow^{tr}_{C} EndOfArbitration : t_8$

Please pay attention to $R_{3.1.1.1}$, $R_{3.2.1}$ and $R_{3.2.2}$. When/After the arbitration process starting $Arbitrator$ has strong power/permission to overrule the Certificate Of Engineer about the dispute (that is the only reason to bring in the arbitration process). Once the $Arbitrator$ overrules the Certificate Of Engineer the related obligations of $Employer$ and $Contractor$ about the $CertificateOfEngineer$ are dismissed. However in the arbitration stage the $Arbitrator$ instructs the $Employer$ and $Contractor$ not to follow it. According to the priority relation, we should conform to $R_{3.2.1}$ and $R_{3.2.1}$-not to follow the $CertificateOfEngineer$ related with the dispute.

Failure to comply with Engineers decision This clause gives us another situation leading to the phase of *StartArbitrating*. We have several actions in this phase:

- Action $A_{3.1}$: ($Employer$ or $Contractor$) to comply with the *NotiOfDeciOfEngi* about the dispute
- Action $A_{3.2}$: ($Employer$ or $Contractor$) to refer the failure of complying with the *NotiOfDeciOfEngi* about initial dispute to arbitration

The rules governing this phase are:

$R_{4.1.1}$: $EndOfDisp : t_8 \Rightarrow^{per}_{O_{em}} E_{co}A_{3.1} : t_8$
$R_{4.1.2}$: $EndOfDisp : t_8 \Rightarrow^{per}_{O_{co}} E_{em}A_{3.1} : t_8$
$R_{4.1.1.1}$: $EndOfDisp : t_8, \neg E_{em}A_{3.1} : t_9, t_9 > t_8 \Rightarrow^{per}_{C} Vl_{co}(em, R_{4.1.1}) : t_9$
$R_{4.1.2.1}$: $EndOfDisp : t_8 \neg E_{co}A_{3.1} : t_{10}, t_1 0 > t_2 + 70 \Rightarrow^{per}_{C} Vl_{em}(co, R_{4.1.2}) : t_{10}$
$R_{4.2.1}$: $Vl_{co}(em, R_{4.1.1}) : t_{10} \Rightarrow^{per} P_{FIDIC}E_{em}A_{3.2} : t_{10}$
$R_{4.2.2}$: $Vl_{em}(co, R_{4.1.2}) : t_{10} \Rightarrow^{per} P_{FIDIC}E_{co}A_{3.2} : t_{10}$
$R_{4.3.1}$: $E_{em}A_{3.2} : t_{10} \Rightarrow^{tr}_{C} StartArbitrating : t_{10}$
$R_{4.3.2}$: $E_{co}A_{3.2} : t_{10} \Rightarrow^{tr}_{C} StartArbitrating : t_{10}$

Rules $R_{4.1.1.1}$ and $R_{4.1.2.1}$ signal the occurrence of a violation, and the violation can trigger permission for the parties involved to start new procedures, rules $R_{4.2.1}$ and $R_{4.2.2}$.

Here the referred arbitration issue is not the same one as the initial dispute issue. Here is to arbitrate the failure of complying with the $DecisionFromEngineer$ about the initial dispute matter and this arbitration would be directly handled by the $Arbitrator$ without the stage of $Engineer$'s decision and stage of amicable settlement.

3.3 Verification in KB of FIDIC contract management

Generally speaking the verification should have three levels. The first level is to verify the soundness, completeness and consistence in the representation of the knowledge in *FIDIC* contract management. The knowledge of *FIDIC* contract management KB includes the *FIDIC* regulation, *FIDIC* contract, and *FIDIC* business processes. The second level is to verify the soundness, completeness and consistence of proof and reasoning mechanism in the *FIDIC* KB, e.g. the compliance and the conformance between *FIDIC* regulation, *FIDIC* contract and *FIDIC* business processes. The third level is that the *FIDIC* KB domain expert can check the whole KB to see whether it gives a complete and correct picture of the *FIDIC* contract management.

For the first level of verification, the aim is to check the internal consistency of the mapping of *FIDIC* specifications. For this aspect we envisage the use of formal methods techniques (e.g., use of automated theorem prover or model checking) logical properties of the specifications, for example consistency, or that a particular property is guaranteed to hold in specified cases. The framework presented in this paper offers a natural environment for the representation of *FIDIC* regulations and the clauses of a *FIDIC* contract. For the business processes we have the option to model them using a business process/workflow language (e.g., BPMN, EPC, YAWL, ...) or to encode them in Defeasible Logic and then use a Defeasible Logic engine to execute the process.

The verification in the second level concerns the alignment of the various specifications. This means, for example to check that the rule encoding the specific clauses do not violate the general clauses, and similarly for the rules encoding the process. If a process is encoded as a business process, then the verification required at this level corresponds to checking the compliance of a business process with the set of regulations concerning it, for this aspect we can adopt the techniques proposed by [4,13,11].

For the third level, the verification requires to adhere to the isomorphism principle proposed by [1] to ensure a close mapping between the formal representation and the natural language representation. While the mapping proposed here follows the ideas and principles well understood in legal theory and representation of normative documents in formal language [14], the mapping can be understood as an interpretation of a normative documents (or collection of documents). As such the mapping must be validate by domain expert (lawyers).

4 Conclusion

The multi-agent system for contract management and some other normative based business application could be built on a knowledge base in Defeasible Logic. This paper primarily focusses on the Defeasible Logic framework and its effectiveness as a means of representing a knowledge base for contract management. Furthermore, the basic theory and a simple example demonstrating the aforementioned capabilities of Defeasible Logic has been presented. While in this paper we have focused on the mapping of Clause 67, the proposed Defeasible Logic framework seems to able to capture in a natural and conceptual way general conditions in FIDIC regulation and contracts.

Acknowledgements

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

References

1. Trevor Bench-Capon and F. P. Coenen. Isomorphism and legal knowledge based systems. *Artificial Intelligence and Law*, 1(1):65–86, 1992.
2. E.C. Corbet. *FIDIC 4th-A PRACTICAL LEGAL GUIDE*. Sweet and Waxwell, 1991.
3. Guido Governatori, Joris Hulstijn, Régis Riveret, and Antonino Rotolo. Characterising deadlines in temporal modal defeasible logic. In Mehmet A. Orgun and John Thornton, editors, *20th Australian Joint Conference on Artificial Intelligence, AI 2007*, Lecture Notes in Artificial Intelligence, pages 486–496, Berlin, 2–6 December 2007. Springer.
4. Guido Governatori, Zoran Milosevic, and Shazia Sadiq. Compliance checking between business processes and business contracts. In Patrick C. K. Hung, editor, *10th International Enterprise Distributed Object Computing Conference (EDOC 2006)*, pages 221–232. IEEE Computing Society, 2006.
5. Guido Governatori and Antonino Rotolo. A computational framework for non-monotonic agency, institutionalised power and multi-agent systems. In Daniéle Bourcier, editor, *Legal Knowledge and Information Systems*, Frontieres in Artificial Intelligence and Applications, pages 151–152, Amsterdam, 2003. IOS Press.
6. Guido Governatori and Antonino Rotolo. Defeasible logic: Agency, intention and obligation. In Alessio Lomuscio and Donald Nute, editors, *Deontic Logic in Computer Science*, LNAI, pages 114–128, Berlin, 2004. Springer.
7. Guido Governatori and Antonino Rotolo. A computational framework for institutional agency. *Artificial Intelligence and Law*, 16(1):25–52, 2008.
8. Guido Governatori and Antonino Rotolo. On the complexity of temporal defeasible logic. In Thomas Meyer and Eugenia Ternovska, editors, *13 International Workshop on Non-Monotonic Reasoning (NMR 2010)*, CEUR Workshop Proceedings, 2010.
9. Guido Governatori, Antonino Rotolo, and Rossella Rubino. Implementing temporal defeasible logic for modeling legal reasoning. In Kumiyo Nakakoji, Yohei Murakamk, and Eric McCready, editors, *New Frontiers in Artificial Intelligence: JSAI-isAI 2009 Workshops*, LNAI, Berlin, 2010. Springer.
10. Guido Governatori, Antonino Rotolo, and Giovanni Sartor. Temporalised normative positions in defeasible logic. In Anne Gardner, editor, *10th International Conference on Artificial Intelligence and Law (ICAIL05)*, pages 25–34. ACM Press, June 6–11 2005.
11. Guido Governatori and Shazia Sadiq. The journey to business process compliance. In Jorge Cardoso and Wil van der Aalst, editors, *Handbook of Research on BPM*, pages 426–454. IGI Global, 2009.
12. Ho-Pun Lam and Guido Governatori. The making of SPINdle. In Guido Governatori, John Hall, and Adrian Paschke, editors, *Rule Representation, Interchange and Reasoning on the Web*, number 5858 in LNCS, Berlin, 5-7 November 2009. Springer.
13. Shazia Sadiq, Guido Governatori, and Kioumars Naimiri. Modelling of control objectives for business process compliance. In Gustavo Alonso, Peter Dadam, and Michael Rosemann, editors, *BPM 2007*, number 4714 in Lecture Notes in Computer Science, pages 149–164, Berlin, 2007. Springer.
14. Giovanni Sartor. Legal reasoning: A cognitive approach to the law. Springer, 2005.