Legal Consolidation formalised in Defeasible Logic and based on Agents.

Régis Riveret, Monica Palmirani, Antonino Rotolo
C.I.R.S.F.D, University of Bologna,
via Galliera 3, 40100 Bologna, Italy
{rriveret, palmiran, rotolo}@cirsfid.unibo.it

Abstract. Updated legal corpora have been indicated by the European Union as fundamental to eDemocracy, and member states looking to set up eGovernment initiatives are acting on that input. However, the usual automation of legal consolidation presents shortcomings, namely, the collapse of temporal dimensions and local views of normative systems. This paper presents solutions to these shortcomings by providing the formalisation in logic of an appropriate legal temporal model and an investigation of the use of the multi-agent paradigm.

Keywords: Consolidation, knowledge representation and reasoning, agents.

1. Introduction

The need to obtain updated legal corpora has been indicated by the European Union as fundamental to eDemocracy, and member states looking to set up eGovernment initiatives are acting on that input. However, collections of digital legal documents managed within information systems open the way to the automation of legal consolidation so that human users, such as jurists and citizens, and non-human users, such as software agents, can access the updated version of legal provisions. However, usual legal consolidation modules tend to present serious shortcomings, namely, (i) the collapse of the temporal dimensions of force, efficacy and applicability into a flat model, so the modifications are applied to the legal documents in a wrong time sequence with respect to legal principles; (ii) they tend to have a local view of the normative system, failing to get a global view, so they risk producing an incoherent normative system; (iii) they do not deal with conditional modifications; (iv) they are not proactive, in the sense that they do not take initiatives to detect modifications that have not been factored into consolidation.

As a remedy to these drawbacks, we present in this paper a system for managing the process whereby the documents in a normative system get consolidated. These shortcomings are overcome by providing a legal temporal model (Palmirani and Brighi, 2003) that respects legal principles and that is formalised into logic to permit automatic reasoning, and by investigating the use of the multi-agent paradigm in such a way as to allow proactivity and a larger view of the normative system.

The temporal model is briefly presented in Section 2. Its logical expression is presented in Section 3. Finally, the use of the multi-agent paradigm is investigated in Section 4.

2. The Legal Temporal Model

The temporal model allows us to give an accurate account of the dynamics of norms over time and therefore to manage legal consolidation consistently with legal principles. This model (Palmirani and Brighi, 2006) is briefly presented here.
First, we will introduce some terminology. A legal system is defined as a set of documents fixed at a defined time t and which have been issued by an authority and whose validity depends on rules that determine, for any given time, whether a single document belongs to the system. Formally:

$$LS(t) = D_1(t), D_2(t), D_3(t), ...D_m(t)$$

where $m \in \mathbb{N}$, D_i denotes documents and t is a fixed time in a discrete representation. A normative system, in turn, takes the documents belonging to a legal system and organizes them to reflect their evolution over time. A normative system should therefore be more precisely defined as a particular discrete time-series of legal systems that evolves over time. In formal terms:

$$NS = LS(t1), LS(t2), LS(t3), ...LS(tj)$$

where $j \in \mathbb{N}$. The passage from a legal system to another legal system is effected by normative modifications (besides persistence), and these can be of different sorts. A taxonomy of modifications can be defined according their effects, which we divide in four main categories (Palmirani and Brighi, 2006): (i) textual changes that intervene when a law is repealed, replaced, integrated or relocated; (ii) changes of scope consequent on derogation, extension, or interpretation; (iii) temporal changes that impact on the date of force, the date of efficacy, or date of application of the destination norm (the entire act or a part of it, such as an article or a paragraph); (iv) normative-system changes that apply not only to specific documents but to the normative system considered in its entirety. A modification is initiated in a text called *active* norm and produces its effect on another text, a receiving text called *passive* norm. The reference to the passive norms may be incomplete or insufficiently clear, and that makes it necessary to have a distinction between *explicit* and *implicit* modifications according as the reference(s) to the passive norm(s) is accurate and complete (explicit change) or not (implicit change).

As any other legal provision, a modification can be seen as a conditional statement. Accordingly, a modification has three temporal dimensions, these being attached to the conditions, to the effects, and to the overall conditional. These temporal dimensions refer, then, to the efficacy, applicability, and force of the provision, respectively. Furthermore, one has to consider the time of observability of the normative system. Consider a law X of 2001 nullified in 2005: The change affects the entire normative system because the legal text is removed from the system as if it had never been there in the first place (ex-tunc removal). The same would happen with a temporary law decree that does not pass into law, or with a retroactive abrogation, or with an interpretation of a law that comes in as the authentic reading of it (Guastini, 1998)(Pagano, 2001). The peculiarity of system changes shows up when we query the system to retrieve information from it: if today (e.g., 2005) we ask for all the laws in force in 2001, law X will not turn up and the system will look as if that law had never been in force in the first place. But if we enter the query as if we were in 2001, when the annulment had not yet occurred, law X will show up as being in force, and the entire system will reflect that fact. This difference depends on the temporal point of view from which we query the system and this refers to the time of observability of a normative system.

3. Model Representation

The application we are working on is designed to provide practitioners with documents for consolidation, all while giving advice backed by proof of correctness. Advice is provided anytime a modification is detected that has not yet been factored into consolidation. Advice and proofs are generated by inference mechanisms that comply with legal principles. The input is the normative information contained in the system and is encoded in an expressive logic, namely, in Temporal Defeasible Logic (TDL) which is an extension of Defeasible Logic (DL). TDL has proved useful in modelling temporal aspects of normative reasoning, such as temporalised normative positions (Governatori et al, 20005); in addition, the notion of a temporal viewpoint-the temporal position from which things are viewed-allows for a logical account of norm modifications and retroactive rules (Governatori et al, 2005). DL and TDL are briefly presented below.

3.1. DEFEASIBLE LOGIC

The legal temporal model points out the importance of uncertainty due to the addition of new premises that can invalidate formerly derivable consequences. This means that consolidation must proceed on the basis of non-monotonic reasoning. In fact, the reasoning used in consolidation forms part of the wider realm of legal reasoning, which too is deemed to be non-monotonic (Sartor, 2005). Non-monotonic reasoning is supported by a number of non-monotonic logics. Among these, DL (Nute, 1987)(Nute, 94)(Antoniou et al., 2001) is based on a logic programming-like language and it is a simple, efficient but flexible non-monotonic formalism capable of dealing with many different intuitions of non-monotonic reasoning. An argumentation semantics exists (Governatori, 2004) that makes its use possible in argumentation systems (Verheij, 2005). DL has a linear complexity (Maher, 2001) and also has several efficient implementations (Bassiliades et al., 2004).

A Defeasible Logic theory is a structure $D = (F, R, \prec)$ where F is a finite set of facts, R a finite set of rules, and $\prec$ a superiority relation on R. Facts are indisputable statements, for example, “Bob is a minor,” formally written as $minor(bob)$. Rules can be strict, defeasible, or defeaters. Strict rules are rules in the classical sense; whenever the premises are indisputable, so is the conclusion. An example of a strict rule is “Minors are persons,” formally written as $r1$: $minor(X) \rightarrow person(X)$. Defeasible rules are rules that can be defeated by contrary evidence. An example of a defeasible rule is “Persons have legal capacity”; formally, $r2$: $person(X) \Rightarrow haslegalcapacity(X)$. Defeaters are rules that cannot be used to draw any conclusion. Their only use is to prevent some conclusions by defeating some defeasible rules. An example of this kind of rule is “Minors might not have legal capacity,” formally expressed as $r3$: $minor(X) \leadsto \neg haslegalcapacity(X)$. The idea here is that even if we know that someone is a minor, this is not sufficient evidence for the conclusion that he or she does not have legal capacity. The superiority relation between rules indicates the relative strength of each rule. That is, stronger rules override the conclusions of weaker rules. For example, if $r3 \succ r2$, then the rule $r3$ overrides $r2$, and we can derive neither the conclusion that Bob has legal capacity nor the conclusion that he does have legal capacity.

Given a set R of rules, we denote the set of all strict rules in R by R_s, the set of defeasible rules in R by R_d, the set of strict and defeasible rules in R by R_{sd}, and the set of defeaters in R by R_{dft}. $R[q]$ denotes the set of rules in R with consequent q. In the following $\sim p$ denotes the complement of p, that is, $\sim p$ is $\neg p$ if p is an atom, and $\sim p$ is q if p is $\neg q$. For a rule r we will use $A(r)$ to indicate the body or antecedent of the rule and $C(r)$ for the head or consequent of the rule. A rule r consists of its antecedent $A(r)$ (written on the left; $A(r)$ may be omitted if it is the empty set), which is a finite set of literals; an arrow; and its consequent $C(r)$, which is a literal. In writing rules we omit set notation for antecedents.
Conclusions are tagged according to whether they have been derived using defeasible rules or strict rules only. So, a conclusion of a theory D is a tagged literal having one of the following four forms:

$+\Delta q$ meaning that q is definitely provable in D.
$-\Delta q$ meaning that q is not definitely provable in D.
$+\partial q$ meaning that q is defeasibly provable in D.
$-\partial q$ meaning that q is not defeasibly provable in D.

These different notions of provability come of use here because they enable the system to label a suggestion as stronger or weaker depending on the kind of proof associated with it. Provability is based on the concept of a derivation (or proof) in D. A derivation is a finite sequence $P = (P(1),..., P(n))$ of tagged literals. Each tagged literal satisfies some proof conditions. A proof condition corresponds to the inference rules that refer to one of the four kinds of conclusions we have mentioned above. $P(1..n)$ denotes the initial part of the sequence P of length n. In this paper, conditions for $\pm\Delta q$, which only describe forward chaining of strict rules, are omitted due to lack of space. We state below the conditions for defeasibly derivable conclusions:

$+\partial$: If $P(i+1) = +\partial q$ then either
  (1) $+\Delta q \in P(1..i)$ or
  (2) (2.1) $\exists r \in R_{sd}[q] \forall a \in A(r) : +\partial a \in P(1..i)$ and
    (2.2) $-\Delta{\sim}q \in P(1..i)$ and
    (2.3) $\forall s \in R[\sim q]$ either
      (2.3.1) $\exists a \in A(s) : -\partial a \in P(1..i)$ or
      (2.3.2) $\exists t \in R_{sd}[q]$ such that
        $\forall a \in A(t) : +\partial a \in P(1..i)$ and $t \succ s$.

$-\partial$: If $P(i+1) = -\partial q$ then
  (1) $-\Delta q \in P(1..i)$ and
  (2) (2.1) $\forall r \in R_{sd}[q]\ \exists a \in A(r) : -\partial a \in P(1..i)$ or

(2.2) $+\Delta{\sim}q \in P(1..i)$ or
(2.3) $\exists s \in R[{\sim}q]$ such that
(2.3.1) $\forall a \in A(s) : +\partial a \in P(1..i)$ and
(2.3.2) $\forall t \in R_{sd}[q]$ either
$\exists a \in A(t) : -\partial a \in P(1..i)$ or $t \not> s$.

Informally, a defeasible derivation for a provable literal consists of three phases: First, we propose an argument in favour of the literal we want to prove. In the simplest case, this consists of an applicable rule for the conclusion (a rule is applicable if its antecedent has already been proved). Second, we examine all counter-arguments (rules for the opposite conclusion). Third, we rebut all the counter-arguments (the counter-argument is weaker than the pro-argument) or we undercut the (some of the premises of the counterargument are not provable).

3.2. TEMPORAL DEFEASIBLE LOGIC WITH VIEWPOINTS

Defeasible Logic allows us to deal with incomplete information but as such does not provide any natural means to deal with temporal dimensions as exposed in the legal temporal model (see Section 2). Temporal Defeasible Logic with viewpoints (Governatori et al., 2005) is an extension of Defeasible Logic to deal with temporal aspects. It deals importantly with temporal viewpoints to reflect the time of observability from which the normative system is comsidered. Temporal aspects are integrated by two means.

First, we introduce temporal coordinates that allow us to temporalise literals and rules. A temporal coordinate is a concatenation of pairs of the form $[t, x]$ where t is an instant of time which is an element of a totally ordered discrete set of instants of time $Temp = \{t1, t2, ...\}$ and $x \in pers, trans$ indicates whether the element occurring in the scope of the temporal qualification is persistent ($pers$) in time or not ($trans$). For example, $[1970, pers]$ and $[t + 18, trans]$:$[1970, pers]$ are temporal coordinates.
A temporalised literal is an expression of the form $l{:}T$ where l is a literal and T is a temporal coordinate. Intuitively, the meaning of a temporalised literal $l{:}T$ is that l holds at the coordinate T. For example, $major(bob)$:$[1973, pers]$:$[1968, pers]$ means that Bob is major in 1973 (and later) from somebody reasoning in 1968 (and later). Similarly, a temporalised rule is an expression of the form $r{:}T$ where r is a rule and T is a temporal coordinate. An example of a temporalised rule is:

$$(r{:}\ \ born(X){:}[t, trans] \rightarrow major(X){:}[t + 18, pers]){:}[1970, pers]$$

This rule formalises the provision in force in 1970 and later that somebody get its majority at 18 years old.
Time labels allow us to deal formally with the different temporal dimensions in the legal domain. The time labels associated with the antecedents of a legal rule, with the consequents, and with the overall rule can respectively be interpreted as the time of *efficacy*, *applicability*, and *force* of the represented legal provision.

Temporal calculi are driven by the operations of temporal extension, temporal reduction, temporal diminution and temporal progression of coordinates. These operations, as defined and explained below, allow us to compare coordinates.

A temporal extension consists in the extension of a temporal coordinate of a certain length into a new coordinate with a higher length. Whereas temporal extensions are acknowledged to be necessary, their forms are less certain and are subjects of many proposals. For our purposes, one can distinguish two proposals. The first understands temporal extension by the concatenation of the temporal pair $[0, pers]$ with a temporal coordinate. For example, the coordinate $[5, pers]$:$[20, trans]$ can be extended to $[5, pers]$:$[20, trans]$:$[0, pers]$. The second proposal captures temporal extension by the concatenation of the temporal pair $[t, pers]$ with a temporal coordinate of the form T:$[t, x]$. For example, the coordinate $[5, pers]$:$[20, trans]$ can be extended to $[5, pers]$:$[20, trans]$:$[20, pers]$. We propose thus two definitions of temporal extensions in order to represent the proposals.

DEFINITION 1. *A temporal extension of a temporal assertion* $\gamma{:}T$ *is the operation concatenating the pair* $[0, pers]$ *with the temporal coordinate* T.

DEFINITION 2. *A temporal extension of a temporal assertion* $\gamma{:}T$ *is the operation concatenating the temporal pair* $[t, pers]$ *with a temporal coordinate of the form* $T{:}[t, x]$.

While our setting allows us to deal with these two definitions, it is worth emphasing that one using the logical framework has to commit to a unique form of extension.

A temporal reduction is the complement of extension in the sense that it consists in the reduction of a temporal coordinate of a certain length into a new coordinate with a lower length. Since there exists two proposals of temporal extension, we propose thus two definitions of temporal reductions:

DEFINITION 3. *A temporal reduction of a temporal assertion γ:T is the operation that reducts a temporal coordinate of the form T:$[0, pers]$ into a temporal coordinate T.*

DEFINITION 4. *A temporal reduction of a temporal assertion γ:T is the operation that reducts a temporal coordinate of the form T:$[t, x]$:$[t, pers]$ into a temporal coordinate T:$[t, x]$.*

Note that one that has committed to the first (second) definition of temporal extension has to commit to the first (second) definition of temporal reduction. Extensions and reductions are mathematical devices allowing for comparison between coordinates that do not have initially the same length. Using temporal extensions and reductions, coordinates can then be brought into relation to each other by way of the equality operator $=$. At the first sight, that two coordinates $[t_{i1}, x_{i1}]$:...:$[t_{in}, x_{in}]$ noted T_i and $[t_{j1}, x_{j1}]$:...:$[t_{jn}, x_{jn}]$ noted T_j are equals, i.e. $T_i = T_j$, means they have same length and that each pair $[t_{ik}, x_{ik}]$ equals $[t_{jk}, x_{jk}]$, that is $t_{ik} = t_{jk}$ and $x_{ik} = x_{jk}$. For example, consider the coordinates $[5, pers]$:$[0, trans]$ noted $T1$ and $[5, pers]$:$[0, trans]$ noted $T2$, then one can state that $T1 = T2$. One may, moreover, say intuitively that $[5, pers]$:$[0, pers]$ and $[5, pers]$ are also equals since one can make a temporal extension of the later into $[5, pers]$:$[0, pers]$.

DEFINITION 5. *Let $T_i = [t_{i1}, x_{i1}]$:...:$[t_{in}, x_{in}]$ and $Tj = [t_{j1}, x_{j1}]$:...:$[t_{jn}, x_{jn}]$ be two temporal coordinates. T_i equals T_j (noted $T_i = T_j$) if there exists either a temporal extension or reduction or none of these, of T_i into the form $[t_{j1}, x_{j1}]$:...:$[t_{jn}, x_{jn}]$ such that each pair $[t_{ik}, x_{ik}]$ equals $[t_{jk}, x_{jk}]$, that is $t_{ik} = t_{jk}$ and $x_{jk} = x_{jk}$.*

Another temporal operation called temporal diminution consists in substituting a pair of the form $[t_i, pers]$ with a pair $[t_f, trans]$. For example, by temporal diminution, a temporalised assertion a:$[5, pers]$:$[20, pers]$ becomes a:$[5, pers]$:$[20, trans]$.

DEFINITION 6. *A temporal diminution of a temporal assertion γ:T is the operation allowing the substitution of a pair in T of the form $[t_i, pers]$ with a pair $[t_f, trans]$.*

A temporal progression of a temporal assertion is the operation that allows us to "slip" the assertion from an initial coordinate to a new coordinate. For example, the temporal literal a:$[5, pers]$:$[20, trans]$ can progress to a:$[10, pers]$:$[20, trans]$. Temporal progressions are constraints as follows: a pair of the form $[t_i, trans]$ cannot progress while a pair of the form $[t_i, pers]$ can only progress to another pair $[t_f, pers]$ such that $t_i < t_f$.

DEFINITION 7. *A temporal progression of a temporal coordinate T is the operation that substitutes a pair in T of the form $(t_i, pers)$ by a pair $(t_f, pers)$ such that $t_i < t_f$.*

If two coordinates T_i and T_f are such that T_i can progress to T_f then we can write it $progress(T_i, T_f)$.

DEFINITION 8. *Let T_i and T_f, be two temporal coordinates. We have $progress(T_i, T_f)$ if there exists a combination of temporal extensions, reductions, diminutions and progressions from T_i to T_f.*

For example, we can state $progress([5, pers], [10, pers]{:}[20, trans])$ since the coordinate $[5, pers]$ can be extended into $[5, pers]{:}[0, pers]$, progressed into $[10, pers]{:}[20, pers]$ and finally be dismunished into $[10, pers]{:}[20, trans]$. However, we cannot state $progress([5, trans], [10, pers]{:}[20, trans])$, because it does not exists any combination of temporal operations from $[10, trans]$ to $[10, pers]{:}[20, trans]$.

Based on the notion of temporal progression, we define the operator $<$ that allows us to compare coordinates that are not similar.

DEFINITION 9. *Let $T_i = [t_{i1}, x_{i1}]{:}\ldots{:}[t_{in}, x_{in}]$ and $T_j = [t_{j1}, x_{j1}]{:}\ldots{:}[t_{jn}, x_{jn}]$ be two temporal coordinates. T_i strictly precedes T_j (noted $T_i < T_j$) if by substituting any pair of the form $[t_{ik}, trans]$ by a pair $[t_{ik}, pers]$ to form T_i', one can state $progress(T_i', T_f)$.*

For example, consider $T_1 = [5, trans]{:}[5, pers]$ and $T_2 = [10, trans]{:}[10, trans]$. If one substitute any pair of the form $[t_{ik}, trans]$ by a pair $[t_{ik}, pers]$ in T_1 to form $T_1' = [5, pers]{:}[10, pers]$, then one can state $progress(T_1', T_2)$. Hence, $T_1 < T_2$.

The second step towards the integration of time to DL in order to properly model the temporal aspects in normative systems is the expression of normative modifications. Normative modifications are functions that take as input a rule to modify and return as output the rule modified. Thus the function $m(r_1{:}T_1){:}T_2$ returns the rule obtained from r_1 as such at time T_1 after the application of the modification corresponding to the function m and the result refers to the content of the rule at time T_2.

Given this basic notion of modification, we can define some specific rule-modifications. We will limit ourselves to substitution and annulment, but other temporal and textual modifications can be captured similarly. Suppose r is a generic defeasible rule such as $(r{:}\ a_1{:}T_{a1}, \ldots, a_n{:}T_{an} \Rightarrow b{:}T_b){:}T_r$.

- $substitution(r{:}T, x'_1{:}T'_1/x_1{:}T_1, \ldots, x'_m{:}T'_m/x_m{:}T_m){:}T_s$ says that we operate at T_s a substitution which replaces some temporalised literals $x_i{:}Ti$ in the antecedent or consequent of r with other literals $x'_i{:}T'_i$. The new version of r will hold at T_s. For example, $substitution(r{:}T_r, c{:}T_c/a_1{:}T_{a1}){:}T_s$ returns $(r{:}\, c{:}T_c, a_2{:}T_{a2}, \ldots, a_n{:}T_{an} \Rightarrow b{:}T_b){:}T_s$.
- $annulment(r{:}T){:}T_{an}$ says that $r{:}T$ is annulled at T_{an}. The function $annulment(r{:}T_r){:}T_{an}$ returns the empty rule $(r{:}\bot){:}T_{an}$.

Rule modifications oblige us to deal with new conflicts, namely conflicts between modifications. For example, a substitution $substitution(r{:}T,'_1{:}T'_1/x_1{:}T_1, \ldots, x'_m{:}T'_m/x_m{:}T_m){:}T_s$ and an annulment $annulment(r{:}T){:}T_{an}$ are in conflict if $T_s = T_{an}$.

The formalism we have introduced allows us to have rules in the head of rules, thus we have to admit the possibility that rules are not only given but can be derived. Accordingly we have to give conditions that allow us to derive rules instead of literals. Then we have to extend the notation $R[q]$ to $R[\gamma{:}T]$ where γ is an assertion, that is either a literal or a rule. Given a set of rules R and a set of rule modifiers $M = \{m_1, \ldots, m_n\}$, then $R[r{:}T_r] = \{s \in R, C(s) = r{:}T_r\}$ gives the set of rules whose head results in the rule $r{:}T_r$ after the application of the rule modifier; and $R[\sim r{:}T_r] = \{s \in R, C(s) = m_i(r{:}T_r)\}$ gives the set of rules that modify $r{:}T_r$.

A temporal defeasible theory is a structure $D = (Temp, F, R, \succ)$ where $Temp$ is a discrete totally ordered set of instants of time, F is a finite set of temporalised literals, R is a finite set of temporalised rules, and $\succ$ is the usual superiority relation on R. As in standard DL, we denote the set of all strict rules by R_s, the set of defeasible rules by R_d, the set of strict and defeasible rules by R_{sd}, and the set of defeaters by R_{dft}. Orthogonally to these distinctions of strict, defeasible rules and defeaters, we will assume a discrimination between rules that can be modified and rules that cannot be modified. The set of non modifiable rules is noted R^{perm} (*perm* as permanent). A rule that cannot be modified is labelled by a pair x^{perm} where x the identifier of the rule and the symbol *perm* indicates the non-modifiability of the rule. An example of non-modifiable rule is $(r1^{perm}{:}\, a{:}[2, pers] \Rightarrow b{:}[1, trans]){:}[1, pers]$

A conclusion of a theory D is a tagged temporal assertion having one of the following forms:

$+\Delta\gamma{:}T$ meaning that $\gamma{:}T$ is definitely provable in D.

$-\Delta\gamma{:}T$ meaning that $\gamma{:}T$ is not definitely provable in D.
$+\partial\gamma{:}T$ meaning that $\gamma{:}T$ is defeasible provable in D.
$-\partial\gamma{:}T$ meaning that $\gamma{:}T$ is not defeasible provable in D.

Provability is based on the concept of a derivation (or proof) in D. A derivation is a finite sequence $P = (P(1), .., P(n))$ of tagged literals. Each tagged literal satisfies some proof conditions, which correspond to inference rules for the four kinds of conclusions we have mentioned above. Let γ be an assertion, namely either a literal or a rule.

$+\Delta$: If $P(i+1) = +\Delta\gamma{:}T$ then
(1) $\gamma{:}T_i \in F \cup R^{perm}$ and $progress(T_i, T)$, or
(2) $\exists r{:}T_r \in R_s^{perm}[\gamma : T_\gamma]$ and $progress(T_\gamma, T_r)$,
(2.1) $+\Delta r{:}T_r \in P(1..i)$, and
(2.2) $\forall\gamma{:}T_\gamma \in A(r{:}T_r), +\Delta\gamma{:}T'_\gamma \in P(1..i)$ and $T'_\gamma = T_\gamma{:}T_r$.

$-\Delta$: If $P(i+1) = -\Delta\gamma{:}T$ then
(1) $\gamma{:}T_i \notin F \cup R^{perm}$ and $progress(T_i, T)$, and
(2) $\forall r{:}T_r \in R_s^{perm}[\gamma{:}T_\gamma]$ and $progress(T_\gamma, T_r)$,
(2.1) $-\Delta r{:}T_r \in P(1..i)$, or
(2.2) $\exists\gamma{:}T_\gamma \in A(r{:}T_r), -\Delta\gamma{:}T'_\gamma \in P(1..i)$ and $T'_\gamma = T_\gamma{:}T_r$.

A temporalised assertion $\gamma{:}T$ is definitely provable ($+\Delta$) if it belongs to the set of unmodifiable rules or facts, or there exists a unmodifiable temporal rule $r{:}T_r$ whose consequent is $\gamma{:}T_\gamma$ with $progress(T_\gamma{:}T_r, T)$ such that 2.1) the rule itself is provable and 2.2) applicable. To prove that a definite conclusion is not possible we have to show that all attempts to give a definite proof of the conclusion fail. Notice that the inference conditions for negative proof tags are derived from the inference conditions for the corresponding positive proof tag by applying the Principle of Strong Negation.

Here is an example for the proof conditions for strict derivations of literals.

$Temp = \mathbb{N}$
$F = \{a{:}[2, pers]\}$
$R^{perm} = \{(r_1^{perm}{:}\ a{:}[2, pers] \rightarrow b{:}[1, trans]){:}[1, pers]$
$(r_2^{perm}{:}\ b{:}[1, trans] \rightarrow q{:}[3, pers]){:}[2, pers]\}$
$\prec = \emptyset$

Suppose we want to know whether $+\Delta q{:}[4, pers]{:}[4, pers]$, i.e., whether q strictly holds at 4 and later, when we consider the evidence and rules that hold at 4 (the time at which we consider the derivation). The only fact in F makes the r_1 applicable and so we obtain, by means of the persistence of r_2, $+\Delta b$:$[1, trans]$:$[2, pers]$. This makes r_2 applicable and, since r_2 and its conclusion are persistent, we obtain $+\Delta q$:$[4, pers]$:$[4, pers]$. One of the preconditions for strictly applying a rule is that it is strictly derivable: notice that all rules used for derivations are strictly provable, as they belong tothe set of non-modifiable rules R^{perm}.

We now turn our attention to defeasible derivations, that is, derivations giving a temporal assertion $\gamma{:}T$ as a defeasible conclusion of a theory D. Defeasible provability ($+\partial$) consists of three phases. In the first phase, we put forward a supported reason for the temporal assertion that we want to prove. Then in the second phase, we consider all possible attacks against the desired conclusion. Finally in the last phase, we have to counter-attack the attacks considered in the second phase.

$+\partial$: If $P(i+1) = +\partial\gamma{:}T$ then either
(1) $+\Delta\gamma{:}T \in P(1..i)$ or
(2) (2.1) $\exists r{:}T_r \in R_{sd}[\gamma{:}T_\gamma]$ and $progress(T_\gamma{:}T_r, T)$,
(2.1.1)$+\partial r{:}T_r \in P(1..i)$, and
(2.1.2)$\forall\gamma{:}T_\gamma \in A(r{:}T_r)$,
$+\partial\gamma{:}T'_\gamma \in P(1..i)$ with $T'_\gamma = T_\gamma{:}T_r$; and
(2.2) $-\Delta{\sim}\gamma : T \in P(1..i)$, and
(2.3) $\forall s{:}T_s \in R[{\sim}\gamma{:}T_\gamma]$, $T_\gamma{:}T_r \leq T_{\sim\gamma}{:}T_s \leq T$,
$+\partial s{:}T_s \in P(1..i)$, either
(2.3.1) $\exists\beta{:}T_\beta \in A(s{:}T_s)$, $-\partial\beta{:}T'_\beta \in P(1..i)$ and
$T'_\beta = T_\beta{:}T_s$, or
(2.3.2) $\exists w{:}T_w \in R[\gamma{:}T_{w\gamma}]$, $T_{\sim\gamma}{:}T_s \leq T_{w\gamma}{:}T_w$, and
$progress(T_{w\gamma}{:}T_w, T)$, and
$w{:}T_w \succ s{:}T_s$ if $T_{\sim\gamma}{:}T_s = T_{w\gamma}{:}T_w$, and
(2.3.2.1) $+\partial w{:}T_w \in P(1..i)$, and
(2.3.2.2) $\forall\chi{:}T_\chi \in A(w{:}T_w)$,
$+\partial\chi{:}T'_\chi \in P(1..i)$ and $T'_\chi = T_\chi{:}T_w$, and
(2.4) $\forall r \in R, +\partial r{:}T_r \in P(1..i)$, $-\partial m(r{:}T'_r){:}T_m$, $T_m \leq T$, $T'_r \leq T_r$.

Let us illustrate this definition. To show that $\gamma{:}T$ is provable defeasibly we have two choices: 1) We show that $\gamma{:}T$ is already definitely provable; or 2) we need to argue using the defeasible part of D as well. In particular, we require that there must be a strict or defeasible rule $r{:}T_r$ with head $\gamma{:}T_\gamma$ and $progress(T_\gamma{:}T_r, T)$ which can be applied (2.1). But now we need to consider possible attacks, i.e., reasoning chains in support of ${\sim}\gamma{:}T$. To be more specific: to prove $\gamma{:}T$ defeasibly we must show

that $\sim\gamma{:}T$ is not definitely provable (2.2). Also (2.3) we must consider any rule $s{:}T_s$ which has head $\sim\gamma{:}T_{\sim\gamma}$ with $T_\gamma{:}T_r \leq T_{\sim\gamma}{:}T_s \leq T$ (note that here we consider defeaters, too, whereas they could not be used to support the conclusion γ; this is in line with the motivation of defeaters given earlier) which is known to be proved $(+\partial s{:}T_s)$. Basically, each such rule $s{:}T_s$ attacks the conclusion $\gamma{:}T_\gamma{:}T_r$. For $\gamma{:}T$ to be provable, each such rule $s{:}T_s$ must be counterattacked by a rule $w{:}T_w$ which has a head $\gamma{:}T_{w\gamma}$ with $T_{\sim\gamma}{:}T_s \leq T_{w\gamma}{:}T_w$ and $progress(T_{w\gamma}{:}T_w, T)$ with the following properties: $w{:}T_w$ must be defeasibly proved (2.3.2.1) and applicable (2.3.2.2). Note that if $T_{\sim\gamma}{:}T_s = T_{w\gamma}{:}T_w$ then $w{:}T_w$ must be stronger than $s{:}T_s$. So each attack on the conclusion $\gamma{:}T$ must be counterattacked. Finally, any rule $r{:}T_r$ that have participated to the derivation of $+\partial\gamma{:}T$ should not have been modified before T (2.4).
Here is an example for the proof conditions for strict derivations of literals.

$Temp = \mathbb{N}$
$F = \{a{:}[10, trans]\}$
$R_{perm} = \{(r_1{:}\ a{:}[t, trans] \Rightarrow b{:}[t, pers]){:}[0, pers]$
$\qquad (r_2{:}\ b{:}[t, trans] \Rightarrow c{:}[t, pers]){:}[50, pers]$
$\qquad (r_3{:}\qquad \Rightarrow annul(r1{:}[0, pers]){:}[0, trans]){:}[100, pers]\ \}$
$\prec = \oslash$

Suppose we want to know whether $+\partial c{:}[50, pers]{:}[60, pers]$. The fact $a{:}[10, trans]$ makes the rule $r_1{:}[10, pers]$ applicable, and we have $+\partial r1{:}[10, pers]$ so we can derive $+\partial b{:}[10, pers]{:}[10, pers]$, and by temporal persistence $+\partial b{:}[10, pers]{:}[50, pers]$. This later result makes the rule $r_2{:}[50, pers]$ applicable, and we have $+\partial r_2{:}[50, pers]$ hence we can derive $+\partial c{:}[10, pers]{:}[50, pers]$ and temporal progression $+\partial c{:}[50, pers]{:}[60, pers]$. Suppose now that we want to know whether $+\partial c{:}[50, pers]{:}[120, pers]$. The fact $a{:}[10, trans]$ makes the rule $(r_1{:}\ a{:}[t, trans] \Rightarrow b{:}[t, pers]){:}T_{r1}$ applicable, and we have $+\partial r1{:}T_{r1}$ with $T_{r1} < [0, pers]{:}[100, pers]$ (because we can derive $+\partial annul(r1{:}[0, pers]){:}[0, trans]{:}[100, pers]$, see condition 2.4 of the proof conditions), so we have $+\partial b{:}[t_b, pers]{:}T_{r1}$ with $[10, trans]{:}Tr1\ \ [t_b, pers]{:}T_{r1} {<} [0, trans]{:}[100, pers]$. This later result makes the rule $r2{:}T_{r2}$ applicable with $T_{r2} \geq [50, pers]$ and we have $+\partial r2{:}T_{r2}$. Hence we can derive $+\partial c{:}[t_c, pers]{:}T_{r2}$ with $[10, trans]{:}T_{r1} \leq [t_c, pers]{:}T_{r2} < [0, pers]{:}[100, pers]$. Now, since we want to know whether $+\partial c{:}[50, pers]{:}[120, pers]$, substitute tc by 50 and T_{r2} by $[120, pers]$. These values do not verify $[t_c, pers]{:}T_{r2} < [0, pers]{:}[100, pers]$, thus one cannot derive $+\partial c{:}[50, pers]{:}[120, pers]$.

3.3. DIGITAL REPRESENTATION

Logic allows us to express normative knowledge and reason on it. However, it requires being associated to a digital representation so that applications can process the knowledge. In other words, pieces of information contained in legislative documents have to be represented, marked up using any appropriate digital language.
The legislative documents to be consolidated are inserted in the system using dedicated editors, such as Norma-Editor (Palmirani, 2000), that make it possible transform normative documents on the basis of a common XML representation language compliant with the NormeinRete project standards (Circolare 35) (Circolare 40). As has been argued in (Brighi 04)(Palmirani et al., 03), the semantic markup of normative references enables intelligent agents to reason and provide advice in consolidating documents. The concepts on which agents make their decisions are defined in ontologies written in OWL or in any other XML syntax. An OWL ontology of modifications has been defined (Palmirani and Brighi, 2003).

On the other hand, whatever the format of a hypothetical legal-knowledge interchange language built on emerging Semantic Web languages (such as XML, RDF, and OWL) or any other format to facilitate the interaction between legal knowledge systems, a translation into the knowledge-representation format internal to the system can be provided. Such translation can be implemented using XSLT as a translation tool.

4. The Model Architecture

The application we are working on is designed to provide practitioners with documents for consolidation with advice backed by proofs of correctness. This section explains the motivation of a multi-agent architecture for the application, investigates briefly some issues concerning the cooperation of agents, and finally presents an overview of the system.

4.1. MOTIVATION OF MULTI-AGENT ARCHITECTURE

Due to the distributive nature of a normative system, and in order to preserve single administration autonomy and technological/organisational independence (Mecella and Batini, 2001)(De Santis et al., 2005), a fully distributed solution is required. And because the consolidation process is about bringing into relation the information distributed over different elements, the solution has to be such as to allow advanced cooperation among these distributed elements. Furthermore, the normative environment is under constant change consequent on normative modifications that occur unpredictably, which makes the evolution of the normative system unpredictable, too. Consequently, the application needs to be easily scalable, flexible, and adaptive to the unpredictable evolution and heterogeneity of the normative environment, i.e., the application should be able to integrate new information in the process. This means that consolidation requires autonomy, that is, the application relies only scarcely on the prior knowledge of its designer, and largely on new information coming in from the normative system. Moreover, a normative modification occurs by an active provision that modifies a passive provision. So the application needs reactivity (i.e., a capacity to reflect the passive provisions of modifications) as well as proactivity (i.e., a capacity to reflect the active provisions of modifications). Finally, there are many different sources of normative knowledge, different processes of norm production, and different kinds of users, and this heterogeneity makes it necessary to have an interoperable system. Interoperability is usually ensured by meeting certain standards, such as XML. To sum up, the goal of supporting consolidation, coupled with the distributed and heterogeneous nature of the normative environment, requires distributed and standards-based technologies enabling proactivity, scalability, and autonomy as well as advanced cooperation among distributed elements.

There are three basic models for distributed applications: the client/server model, the peer-to-peer model, and the agent-based model. The client/server model makes a distinction of roles between its distributed elements, namely, servers and clients. Usually, servers are reactive, in that they cannot act on their own initiative to communicate with a client; they handle most of the capabilities and must provide stability (they cannot appear and disappear, for example). In contrast, clients can take initiatives (they can take the initiative of communicating with a server but cannot communicate directly with one another); they have few capabilities, and they are allowed to appear and disappear. The client/server architecture is the most widespread architecture of legal applications on the Web. In a peer-to-peer architecture, instead, nodes can function as both servers and clients at any time, depending on the role node acts in. The agent-based model makes no such distinctions of roles. Agents are software paradigms that are proactive (they can take initiatives) and social (they can communicate using high-level protocols); they can be endowed with artificial intelligence and so can reason, adapt to new situations, and act autonomously, without human guidance. Moreover, agents facilitate the design of applications that manage the complexity of a domain (here, the normative system). Indeed, as the complexity of a system arises from the interactions between the elements of the system, agents facilitate their representation and the implementation of their interactions. Finally, the integration of agents using step-by-step procedures allows scalability and flexibility.

So, in view of the requirements specified above and of the technologies available, the agent-based model appears to be the one best suited to handle the characteristics of a dynamic normative system.

4.2. COOPERATION AMONG AGENTS

Agents may not have enough expertise and resources (each one has a partial view of the normative system), so they need to cooperate to exchange information, and vice versa (they need to exchange information to cooperate). In multi-agent settings, the exchange of information is usually done using either direct or indirect message-passing. Typically, direct message-passing refers to the exchange of messages directly between agents using agent communication languages and protocols, while indirect message-passing relates to the exchange of messages indirectly between agents via centralised artefacts such as blackboards. Since a centralised solution is little coherent with administrative federalism, our choice has been for agents to exchange information using direct message-passing. In a direct message-passing setting, messages are written in an Agent Communication Language (ACL). Most of the work done in ACL's is based on the theory of speech acts, so that messages are usually conceived as speech acts. Eventually, agent cooperation may require more than one-shot utterances, i.e., the exchange of a complex series of messages. This can be effected through the use of protocols determining which speech acts can be used at various points in conversation. One can wonder what types of interactions hold between the bodies of a legal system. Or, in technical terms, what are the relevant speech acts and protocols? One may appreciate a set of speech acts and protocols that is isomorphic with the set of real interactions taking place between the bodies of a legal system.

For our purposes, we need both information dialogues and argumentation dialogues. Information dialogues define the content of utterances as essentially propositional, with different speech acts, such as assertive or interrogative. Argumentation dialogues allow agents to exchange arguments, justify themselves, and persuade other agents. The dialogues involve agents putting forward arguments for or against propositions together with justifications for the acceptability of these arguments.

4.3. OVERVIEW OF A MAS APPLICATION

The MAS is designed to support users in consolidating legislative documents by providing advice proactively and autonomously on the basis of the semantic representations contained in the text and the behaviour built into the agents. The system should support users in carrying out some main tasks as follows:

- Detecting proactively and autonomously (a) any events that modify the normative system; (b) the conditions that modifications are dependent on; (c) any inconsistencies that a modification may bring to the normative system; (d) any antinomies; (e) any anomalies, as when a modification is modified; and (f) events that result in a double negation, as when an abrogation is abrogated.
- Giving advice aimed at consolidating documents.
- Providing proofs that the conditions for a consolidation to be effective hold.

If a modification is found to be effective, the information is routed to the user. The user might be sceptical, asking whether this modification can be trusted, and particularly whether all the conditions that produce a valid modification are verified. The agent thus undertakes to provide proof of validity for the modification. The proof is then presented in a user-friendly manner to the practitioner, who can thus check whether a modification is valid or invalid, and who is then responsible for following the advice or not (depending on whether the modification was found to be valid or invalid). If the user accepts the suggestion, a new consolidated version of the passive document is produced, and the entire normative system shall be properly updated.

This MAS application is particularly efficient in verifying the conditions to which a modification is subject in the normative system, so that we can (upon verification) launch proper events and produce updated law in real time. Finally, this way of processing the Semantic Web content embedded in legislative documents enables us to manage the normative system as made up of different agents. These agents can simulate the dynamicity of the normative system, which can react properly to modifications on the basis of behaviours designed to mimic mental attitudes (Boella and van der Torre, 2003).

5. Conclusion

This paper is part of an ongoing effort in Computer Sciences and Law to provide appropriate models, representation of legal reasoning, and computer science paradigms to deal with the dynamics of normative systems. We argued that our solution overcome the main shortcomings of traditional legislative information systems by the formalisation in logic of an adequate temporal model, and by an investigation into the use of the multi-agent paradigm. The expected result is the improvement of the automation of legal consolidation.

References

Studio di fattibilitá per la realizzazione del progetto “Accesso alle norme in rete”. *Informatica e Diritto* No. 1, 2000.

Antoniou, G., D. Billington, G. Governatori and M. Maher. Representation results for defeasible logic. *ACM Transactions on Computational Logic*, 2(2):255-287, 2001.

Bassiliades, N., G. Antoniou and I. Vlahavas. DR-DEVICE: A defeasible logic system for the Semantic Web. In PPSWR04. Springer, Berlin, 2004.

Boella, G. and L.W. N. van der Torre. Attributing mental attitudes to normative systems. In Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS’03), ACM Press, 942-943, 2003.

Boella, G. and L.W. N. van der Torre. Permissions and obligations in hierarchical normative systems. In Proceeding of the 9th International Conference on Artificial Intelligence and Law (ICAIL’2003), ACM Press 109-118, 2003.

Brighi, R. An Ontology for Linkups between Norms. In Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA’04), IEEE Computer Society, 122-126, 2004.

Circolare 22 aprile 2002 n. AIPA/CR/40, Formato per la rappresentazione elettronica dei provvedimenti normativi tramite il linguaggio di marcatura XML. Gazzetta Ufficiale n. 102, 3 May 2002.

Circolare 6 novembre 2001 n. AIPA/CR/35, Assegnazione dei nomi uniformi ai documenti giuridici. Gazzetta Ufficiale Serie generale n. 262, 10 November 2001.

De Santis, L., C. Lupo, C. Marchetti and M. Mecella. The x-Leges System for Legislative Document Exchange. In Proceedings of the 3rd International Workshop on Legislative XML, I Quaderni n. 18, November 2005.

Governatori, G., M. Maher, D. Billington, and G. Antoniou. Argumentation semantics for defeasible logics. Journal of Logic and Computation, 14, 5: 675-702, 2004.

Governatori, G., M. Palmirani, R. Riveret, A. Rotolo, and G. Sartor. Normative Modifications in Defeasible Logic. In Marie-Francine Moens, editor, Jurix’05: The Eighteenth Annual Conference. IOS Press, Amsterdam 2005.

Governatori, G., A. Rotolo, and G. Sartor. Temporalised normative positions in defeasible logic. In Anne Gardner, editor, Procedings of the 10th International Conference on Artificial Intelligence and Law, pages 25-34, ACM Press, 6-10, 2005.

Guastini, R. Teoria e Dogmatica delle fonti. Giuffrè, Milano, 1998.

Maher, M. Propositional Defeasible Logic has Linear Complexity. TPLP 1(6): 691-711, 2001.

Mecella, M. and C. Batini. Enabling Italian E-Government through a Cooperative Architecture. In A.K. Elmagarmid, W.J. McIver Jr. (eds.): Special Issue on Digital Government. IEEE Computer, vol. 34, no. 2, pp. 40-45, 2001.

Nute, D. Defeasible reasoning. In Proc. 20th Hawaii International Conference on Systems Science. IEEE Press, 470-477, 1987.

Nute, D. Defeasible logic. In Handbook of Logic in Artificial Intelligence and Logic Programming, D. Gabbay, C. Hogger, and J. Robinson, Eds. Vol. 3. Oxford University Press, Oxford, 353-395, 1994.

Pagano, R. L'arte di legiferare. Giuffrè, Milano, 2001.

Palmirani, M. and R. Brighi. Time Model for Managing the Dynamic of Normative System. EGOV 2006: 207-218

Palmirani, M., R. Brighi, R. and M. Massini. Automated Extraction of Normative References in Legal Texts. In Proceeding of the 9th International Conference on Artificial Intelligence and Law (ICAIL'2003), ACM Press, 105-106, 2003.

Palmirani, M. Norma-System. In the series "Filosofia, Informatica, Diritto", Bologna, Clueb, 2000.

Sartor, G. Legal reasoning. Springer, 2005.

Verheij, B. Virtual Arguments. On the Design of Argument Assistants for Lawyers and Other Arguers. T.M.C. Asser Press, The Hague, 2005.