The Paradoxes of Deontic Logic Revisited: A Computer Science Perspective

Or: Should computer scientists be bothered

by the concerns of philosophers?

J.-J. Ch. Meyer, F.P.M. Dignum, and R.J. Wieringa

UU-CS-1994-38
September 1994

Utrecht University

Department of Computer Science

Padualaan 14, P.O. Box 80.089,
3508 TB Utrecht, The Netherlands,
Tel. : ... + 31 - 30 - 531454

The Paradoxes of Deontic Logic Revisited: A Computer Science Perspective

Or: Should computer scientists be bothered

by the concerns of philosophers?

J.-J. Ch. Meyer, F.P.M. Dignum, and R.J. Wieringa

Technical Report UU-CS-1994-38
September 1994

Department of Computer Science
Utrecht University
P.O.Box 80.089
3508 TB Utrecht
The Netherlands

ISSN: 0924-3275

The Paradoxes of Deontic Logic Revisited: A Computer Science Perspective

(Or: Should computer scientists be bothered by the concerns of philosophers?)[‡]

J.-J. Ch. Meyer[*][¶]
Utrecht University, Dept. of Comp.Sc., P.O. Box 80.089, NL-3508 TB Utrecht

F.P.M. Dignum
Technical University Eindhoven, Dept. of Comp.Sc.

R.J. Wieringa[¶]
Free University, Amsterdam, Dept. of Comp.Sc.

1. Introduction

In this paper we revisit the infamous paradoxes which have plagued deontic logic throughout its history / development. Deontic logic may be described as the logic of prohibitions, permissions and obligations. A better description of deontic logic is the logic of ideal (according to certain norms) versus actual behaviour. The paradoxes in this field are logical expressions that are valid in a (or even most) well-known logical system(s) for deontic reasoning, but which are counterintuitive in a common-sense reading (cf. [Åqv84], Section II). Of course, matching theorems and intuitions is a general problem of measuring a logic (or even more generally, a formal approach) against the purpose it was devised for. (This is sometimes called the *validation* problem.) But it is remarkable that in the realm of deontic logic these problems appear to be much more serious and persistent through the years than in, for instance, other modal logics such as temporal and epistemic logics. Some of these paradoxes appear and re-appear again in the literature for several decades now without the community of researchers seemingly arriving at a consensus. What is even more surprising is the relative simplicity of the paradoxes. Most of them can be explained to a complete layman in deontic logic in a couple of minutes. Yet they have been haunting deontic logicians for many years now. To mention some of the best known ones:

- Ross's Paradox: ought-to-mail-a-letter implies ought-to-mail-a-letter-or-burn-it.
- Free Choice Paradox: allowed-to-mail-a-letter implies allowed-to-mail-a-letter-or-burn-it.
- No Conflicting Obligations: ought implies permitted

[‡] This article is based on an invited speech held by the first author at DEON'94, Oslo, 6-8 January 1994.

[*] Also partially supported by Nijmegen University and the Free University Amsterdam.

[¶] Partially supported by the ESPRIT III BRWG Project No. 8319 'MODELAGE'.

- Good Samaritan Paradox: ought-to-help-Jones-who-is-robbed implies Jones-ought-to-be-robbed
- Chisholm's Paradox: ought-to-do-A, ought-to-do-(A-implies-B), not-A implies ought-to-do-not-B and actually doing-not-A is inconsistent.

Not all paradoxes are considered equally serious. Of the small list above the contrary-to-duty-imperatives such as Chisholm's paradox are generally considered by far the most awkward and even embarrassing by deontic logicians.

Traditionally, deontic logic has been proposed and used for reasoning about ethical and legal aspects. This has resulted in a critical evaluation of the logic in its capacity of supposedly giving an adequate representation and reasoning mechanism for these applications. Originally, also the paradoxes were discovered and judged against this background. However, recently deontic logic has gained a completely new audience: that of computer scientists and AI researchers interested in the specification of advanced systems, e.g. knowledge based or intelligent systems where norms (or rather normative versus nonnormative behaviour) plays a role. This includes a wide variety of systems (and hence also a wide variety of computer scientists), ranging from the specification of fault-tolerant behaviour for use in advanced software to electronic contracting systems for use in organisations, as well as integrity constraints in databases and legal expert systems (cf. [MW93b], in particular [WM93]). Even systems such as libraries require a lot of 'deontics' which have to be formalised and specified in an executable way when one is thinking of automating these systems (cf. [JS93]).

The question that will be addressed in this paper is whether this new and unexpected use of deontic logic opens a new perspective on old problems such as the deontic logic paradoxes. Computer science may provide a different view in at least two ways:

- it may offer a dynamic perspective, taking notions of action and time into account.
- it may also offer a pragmatic perspective.

With respect to the former, computer science may really give some new insight and help the traditional area of deontic logic by offering new interpretations and logics. With respect to the latter, we admit that computer scientists will mainly help themselves, (ab?)using deontic logic the way they need, not being bothered by old and profound questions. So, on the one hand, one has deontic logicians being plagued by long-standing paradoxes without seemingly arriving at a consensus, while on the other hand, one has computer scientists wanting to apply deontic logic to their field of interest. What should be their attitude towards deontic logic, and the paradoxes in particular? Do these present insurmountable problems for them, too? To put it more bluntly, is it wise to use deontic logic or is the area of deontic logic still too much underdeveloped to be used in practice?

We believe this not to be the case (to a certain extent): we will argue that, although, of course, not all long-standing problems become clear or irrelevant when looking at these matters from the perspective of computer science, it is nevertheless undoubtedly the case that some of the problems (i.e., paradoxes) lose their bite when one has less ambitious motives to use deontic logic and one is interested in such prosaic things like a robot system in which a robot is allowed / forbidden to do certain actions rather than the use of deontic logic for reasoning about the moral behaviour of man. On the other hand, some serious problems remain, especially when one is interested in natural language representation and processing.

2. The Paradoxes of Deontic Logic

The paradoxes of deontic logic are logical expressions (in some logical language) that are *valid* in a (or even most, as is often the case) well-known logical system for deontic reasoning, but which are *counterintuitive in a common-sense reading*. Of course, there is the general problem of measuring a formal approach (logic) against the purpose it was devised for: do formal theorems in the logic match the intuition? But in the realm of deontic logic this problem seems to be more serious and persistent than in other (modal) logics such as temporal and epistemic logic (although in epistemic logic there is, for instance, the problem of logical omniscience see, e.g., [MH94]).

2.1. Some well-known paradoxes

The following is a list of the most well-known paradoxes from the deontic logic literature:

1. Empty normative system	$O\varphi$ (φ a tautology), e.g. $O(\varphi \vee \neg\varphi)$
2. Ross' paradox	$O\varphi \to O(\varphi \vee \psi)$
3. No free choice permission	$(P\varphi \vee P\psi) \leftrightarrow P(\varphi \vee \psi)$
4. Penitent's paradox	$F\varphi \to F(\varphi \wedge \psi)$
5. Good Samaritan paradox	$\varphi \to \psi \vdash O\varphi \to O\psi$
6. Chisholm's paradox	$(O\varphi \wedge O(\varphi \to \psi) \wedge (\neg\varphi \to O\neg\psi) \wedge \neg\varphi) \to \perp$
7. Forrester's paradox of gentle murder	$(F\varphi \wedge (\varphi \to O\psi) \wedge (\psi \to \varphi) \wedge \varphi) \vdash \perp$
8. Conflicting obligations	$\neg(O\varphi \wedge O\neg\varphi)$, or equivalently, $O\varphi \to P\varphi$
9. Derived obligation	$O\varphi \to O(\psi \to \varphi)$ $F\psi \to O(\psi \to \varphi)$ $\neg\varphi \to (\varphi \to O\psi)$
10. Deontic detachment	$(O\varphi \wedge O(\varphi \to \psi)) \to O\psi$
11. Kant's ought implies can	$O\varphi \to \Diamond\varphi$
12. Epistemic obligation	$OK\varphi \to O\varphi$

Remarks. 1. says that every tautology is obligated. This is by some authors (including Von Wright [vW51]) viewed as an undesirable property of a deontic logic, since necessary and therefore inevitable things cannot be obligated in a true sense. 2.'s strangeness is often illustrated with an instance like "if one is obliged to mail a letter, one is obliged to mail the letter or burn it." This sounds paradoxical in ordinary language usage. 3. implies that if φ is permitted but ψ is not, nevertheless "φ or ψ" is permitted. This is sometimes felt as counter-intuitive since "φ or ψ is permitted" suggests that one is free to choose either doing φ or doing ψ. 4. is analogous to 3. It has as an instance that if someone is forbidden to do a crime, one is also forbidden to do a crime and do penitence. 5. says that every logical consequence of something that is obligated is obligated itself. This is amenable to debate, as in an instance like "if Jones helps Smith who has been injured implies Smith has been injured, so if it is obliged that Jones helps Smith who has been injured, it is obliged that Smith has been injured". 6. is an abstract version of the set "if you are obliged to go to a party; it is obliged that, if you go, you tell you're coming; but if you do not go, you are obliged to not tell you're coming; and, in fact, you do not go to the party". This is intuitively consistent, but inconsistent in Standard Deontic Logic (SDL) that we shall treat below in Section 2.1. 7.'s paradoxical nature is exemplified by the macabre instance: "one is forbidden to murder; still, if one murders someone, one has to do it gently (i.e. one has to commit a gentle murder); moreover, a gentle murder implies a murder; one murders someone". This set is inconsistent again in SDL, whereas it makes perfect (common) sense. 8. states that there is no conflict of duties, which is manifestly not in line with daily-life situations. 9. are deontic versions of the paradoxes of material implication in classical logic. 10. states that obligations are closed under implication. This is controversial in the same sense as 5. with which it is strongly related. 11. states that only things (actions) that *can* be (done) might be obliged. This denies the possibility to specify that one is really obligated to do the impossible. 12. is puzzling as shown by the instance: "if you ought to know that your spouse is committing adultery, it ought to be that your spouse is committing adultery."

2.1. Standard Deontic Logic (SDL)

One of the first systems for deontic logic that really was a serious attempt to capture deontic reasoning was the now so-called "Old System" of Von Wright ([vW51]), of which a modal (Kripke-style) version has become known as Standard Deontic Logic (SDL).

SDL consists of the following axioms and rules:

(K) $O(\varphi \to \psi) \to (O\varphi \to O\psi)$

(D) $\neg O\bot$

(N) $\varphi / O\varphi$

(P) $P\varphi \leftrightarrow \neg O\neg\varphi$

(F) $F\varphi \leftrightarrow O\neg\varphi$

(T) the tautologies of propositional logic (or just enough of them)

(MP) $\varphi, \varphi \rightarrow \psi / \psi$

Here $\perp$ stands for falsum.

SDL has a Kripke-style modal semantics based on a set of possible worlds(, a truth assignment function of primitive propositions per possible world) and an accessibility relation associated with the O-modality (cf. e.g. [MW93a]). This accessibility relation points to "ideal" or "perfect deontic alternatives" of the world under consideration. The crux behind this is that in some possible world something (say φ) is obligated, if φ holds in all the perfect alternatives of this world, as indicated by the accessibility relation.

A few theorems of **SDL**:

($O\wedge$) $O(\varphi \wedge \psi) \leftrightarrow (O\varphi \wedge O\psi)$

($P\wedge$) $P(\varphi \wedge \psi) \rightarrow (P\varphi \wedge P\psi)$

($F\wedge$) $(F\varphi \vee F\psi) \rightarrow F(\varphi \wedge \psi)$

($O\vee$) $(O\varphi \vee O\psi) \rightarrow O(\varphi \vee \psi)$

($P\vee$) $P(\varphi \vee \psi) \leftrightarrow (P\varphi \vee P\psi)$

($F\vee$) $F(\varphi \vee \psi) \leftrightarrow (F\varphi \wedge F\psi)$

(Cfl) $\neg(O\varphi \wedge O\neg\varphi)$

In **SDL** all paradoxes 1. - 10. are theorems. (We leave 11. and 12. out of our discussion since they involve modalities, viz. knowledge and ontological possibility, which are not present in SDL.) Moreover, it is worthwhile to note what axioms are responsible for them (discarding the principles P and F, which are viewed in this context as mere abbreviations. Of course, one is also free to tempt these as proper definitions of permission and prohibition.) Paradox 7 depends on (K), (N) and (D), while paradoxes 6 and 8 depend on (K) and (D). (K) and (N) are responsible for the paradoxes 2, 3, 4, 5 and the first two of 9. The third one of 9. is even valid in propositional logic. Paradox 1 is an immediate consequence of the necessity rule (N) and paradox 10. is just a consequence of the modal axiom (K). So, apart from the paradoxes 6, 7 and 8, which can be avoided by denying the principle (D), the rest of the paradoxes are direct and unavoidable consequences of viewing the obligation operator O as a ("normal" in the sense of [Che80]) Kripke-style modality, which necessarily satisfies (K) and (N)! (Moreover, denying principle (D) is also not done painlessly, since, historically, it is the very axiom in modal logic that is associated with deontics!) Do we have to conclude from this that this is not a good approach? Well, this depends on the situation at hand. In fact, we shall argue that sometimes it is not, but in other situations this style of semantics is sufficient in adequacy.

3. A Diagnosis of the Problems

At least part of the problems arise from the following three confusions:
(i) confusion between *ought-to-be* and *ought-to-do* [→ dynamic perspective, see below]. Since the inception of deontic logic there has been some confusion about the meaning of Oφ, or rather of the meaning of φ is this context: is it a description of a state-of-affairs or does it denote an act(ion)? We will argue that if one views φ as an action description (i.e., one is interested in so-called ought-to-do's) viewing the O-modality as a modal operator in SDL-style is not adequate, while for ought-to-be's where φ denoted a state-of-affairs this might be adequate for concrete applications [→ pragmatic perspective, see below].
(ii) confusion between the formal interpretation (of the logical operators $\wedge$, $\vee$ and $\rightarrow$, for example) and the natural language (commonsense) reading of these [→ pragmatic perspective].
(iii) confusion between ideality and actuality; and, in particular, an overestimation of ideality and the notion of perfect alternative as it appears in the formal semantics. In particular, norms can conflict in reality! In formal terms: O$\bot$ is not equivalent with $\bot$! (But, of course, in practice, when we encounter conflicting duties, we must try and resolve these [→ dynamic perspective].)

4. The Computer Science Perspective

Computer science and AI offer a new perspective for at least the following three reasons:

(i) In computer science / AI one is interested in the specification of advanced systems such as knowledge-based and intelligent systems) where norms play a role, too. This includes a wide range of systems such as databases, legal expert systems, electronic contracting and fault-tolerant systems (cf. [WM93]). The purpose here is to obtain precise descriptions of (the behaviours of) these systems and the possibility of (mechanical) manipulation of these descriptions to be able to derive certain properties from them. For instance, integrity constraints (IC's) for databases do not only *de*scribe how universe of discourse (UoD) entities behave, but they also *pre*scribe how these entities should behave, which involves normative or deontic notions (cf. [WMW89]).

(ii) Naturally, in computer science a *dynamic* perspective emerges: since traditionally in this area the execution of programs over time is studied, one can imagine that notions and techniques developed for analyzing these may also shed some new light on more general notions of actions and time as they appear in deontic logic (but in other areas such as linguistics as well ([vB93]). We shall pursue this in section 5 below.

(iii) But last but not least, there is in computer science the *pragmatic* viewpoint: what are the relevant (and which are the irrelevant) issues when dealing with specifications of down-to-earth, concrete systems? To put it bluntly, when specifying concrete systems within a specific and limited context one need not solve all deep problems of analytical and moral philosophy!

5. The Dynamic Perspective

As said before, computer science provides a dynamic perspective. Basically this is because in this area one is interested in a way to compute (constructively) things, and since this process generally takes time, this naturally involves changes of computer states over time. We can distinguish at least the following three particular issues that are studied in computer science:

1. First and foremost computer science has to do with (computer) *programs*. Since its inception computer science has to do with algorithms to achieve certain goals and computer programs written in some computer language to enable one to execute these algorithms. These programs contain statements or instructions to express what has to be done. In other words programs express how certain actions involving (the hardware of) the computer should be performed. To reason about programs, special logics have been developed such as Hoare's logic and dynamic logic ([Hoa69], [Har79]). (Also temporal logic has been employed for this purpose (see e.g. [Krö87], [MP92]). In some way one might view temporal logic as a dynamic logic in which one abstracts away from the particular actions that take place, and only considers the flow of time while executing a program. On the other hand, one might also view dynamic logic as a kind of temporal logic where records are kept of what exactly happens in a time step.) In these logics it is possible to express precisely what the effect is of the execution of (parts of) programs. One can use them to reason about pre- and postconditions of (executions of) programs. Since especially in dynamic logic this is done in a way that abstracts from the basic programming actions, it is an easy and straight-forward step to abstract away from the particular application to computer programs and consider general actions, whether they are supposed to be executed by computers or by humans. In this way, dynamic logic may be viewed as a general logic for reasoning about actions.

We ([Mey87, Mey88], [MWW89], [MWM89], [DM90], [WWMD91], [DMW94a, b]) have employed this idea in order to get a deontic logic for ought-to-do (i.e. obliged *actions*). We will sketch this approach below in section 5.1. Admittedly, also in the philosophical literature on deontic logic there have been proposals to distinguish between actions and assertions ([vW81], [Cas81], also cf. [Hil93]), but we believe that the explicit connections with formalisms to reason about programs such as dynamic logic has made things more concrete. Interestingly, recent proposals for a philosophical theory of action as put forward by Krister Segerberg (e.g.

[Seg89]), are also strongly influenced by the dynamic logic approach stemming from computer science and are very close in spirit to our approach.

2. More generally computer science is concerned with the study of *processes*. Processes might be viewed as generalizations of executions of standard (sequential) programs, where also parallel (and nondeterministic) execution is catered for. In fact, there is a whole branch of theoretical computer science dealing with so-called process theory, which ranges from concurrency semantics in which models for concurrent or parallel (nondeterministic) programming are investigated (e.g. [Win82], [BM88], [vG90]) to process algebra, in which one tries to give an algebraic calculus for this kind of programming ([Mil80], [BK86], [BW90]). In these algebraic calculi (or *process algebras*, as they are usually called) one is actually interested in calculating equivalences of processes on the basis of certain observational criteria: if two processes are indistinguishable with respect to some observational criterion, they are regarded as equal, so that one can reduce terms denoting complex processes to those representing simpler ones. This appears to be very useful when considering correctness issues of these processes: by algebraic calculation one aims at verifying certain properties of processes at hand such as protocols for communication between 'agents' (computers or processors within a multi-processor computer system).

We employed the dynamic perspective in the sense of using dynamic logic over process (algebraic) terms to treat the paradox of free choice permission. In [MW91], [WM91], [WM93a] we used an algebraic approach on process terms to distinguish between two choice operators representing free and imposed choice and obtained a framework (albeit complicated) in which both could be used intertwined. Here we mixed techniques from process algebra (and universal algebra) with dynamic logic to get a solution for this well-known problem in deontic logic. (In [DMW94] we gave a less involved solution to this problem using another (stronger) definition of permission together with admissible contexts for actions, but also here dynamic logic is a crucial ingredient of the approach.)

3. In AI research, dynamics comes in when considering complex reasoning patterns, such as defeasible (nonmonotonic) reasoning (cf. e.g. [Luk90], [MT93] as excellent introductions to the area of nonmonotonic reasoning; in [LHM94] the emphasis is on the dynamics of default reasoning viewed from the stand-point of a reasoning agent. However, the dynamics of reasoning is also studied in a more general context of complex reasoning patterns like in meta-level reasoning architectures, cf. e.g. [HMT94]). In the subject of defeasible or nonmonotonic reasoning one considers (sequences of) steps while having the disposal of incomplete and overcomplete (inconsistent) information and "jumping to conclusions". Here we encounter dynamics in the epistemic state of the agent, like belief revision and truth (reason) maintenance.

Actually, this also plays a role when encountering normative information that is incomplete and inconsistent. Historically some of the first work on defeasibility was done in the realm of normative reasoning ([AM81]), where it is imperative to keep one's duties straight in a situation with inconsistent norms. One might speak of "*duty maintenance*" in this case. Recently, mainly inspired by work done in AI on defeasibility over the last decade or so, there have been developing a rapidly increasing interest in the defeasibility approach to deontic logic and legal reasoning more in general (cf. e.g. [MW91a], [MW93a], [Rya93], [Hor93], [Pra93, 94], [Jon93], [PS94], [vdT94]). Clearly, this topic has now been taken up very seriously by researchers in the AI & Law Community, and it is to be expected that a lot of this line of research will dominate deontic logic and reasoning about norms for the years to come.

5.1 Our Approach: The System PDeL of Deontic Logic as Based on Dynamic Logic

PDeL, introduced in [Mey88], is a version of dynamic logic especially tuned to use as ought-to-do style deontic logic. It is based on the idea of Anderson's reduction of ought-to-be style deontic logic to alethic modal logic, but instead it reduces ought-to-do deontic logic to dynamic logic ([Har79]). The basic idea is very simple: some action is forbidden if doing the action leads to a state of violation. In a formula: $F\alpha \leftrightarrow_{def} [\alpha]V$, where the dynamic logic formula $[\alpha]\varphi$ denotes that execution / performance of the action α leads (necessarily) to a state (or states) where φ holds, and V is a special atomic formula denoting violation. Formally, we say that the meaning of action α is captured by an accessibility relation $R_\alpha \subseteq S \times S$ associated with α, where S is the set of possible worlds. This relation R_α describes exactly what possible moves (state transitions) are induced by performance of the action α: $R_\alpha(s, t)$ says that from s one can get into state t by performing α. (In concurrency semantics and process algebra this is often specified by a so-called transition system which enables one to derive (all) transitions of the kind $s \rightarrow_\alpha t$, which in fact defines the relation R_α for all possible actions α.) Now the formal meaning of the formula $[\alpha]\varphi$ is given by: $[\alpha]\varphi$ is true in a state (possible world) s iff all states t with $R_\alpha(s, t)$ satisfy φ. This then provides the formal definition of the F-operator, as given above.

The other deontic modalities are derivatives of F: permission is not-forbidden ($P\alpha \leftrightarrow \neg F\alpha$), and obligation is forbidden-not-to ($O\alpha \leftrightarrow F\bar{\alpha}$), where $\bar{\alpha}$ has the meaning of "not-α". The formal semantics of this negated action is non-trivial, especially in case one considers composite actions, cf. [Mey88], [WMW89], [DM90], [MW91], [WM93a]. In these papers we considered connectives for composing non-atomic actions, such as '$\cup$' (choice, the dynamic analogue of disjunction in a static setting), '&' (parallel, the analogue of conjunction), '$-$' (non-performance, the analogue of negation), and ';' (sequential composition, which has no analogue in a static setting). Without giving a formal semantics here (see the papers mentioned above for that), the meaning of these are as follows: $\alpha_1 \cup \alpha_2$ expresses a choice between α_1 and α_2 (this—roughly—corresponds to taking $R_{\alpha_1 \cup \alpha_2}$ as the set-theoretic union of R_{α_1} and R_{α_2}), $\alpha_1 \& \alpha_2$ a parallel performance of α_1 and α_2 (this amounts to more or less taking $R_{\alpha_1 \& \alpha_2}$ to be the intersection of R_{α_1} and R_{α_2}), $\bar{\alpha}$ the non-performance of α, as stated above (it more or less amounts to taking $R_{\bar{\alpha}}$ to be some complement of R_{α}, but see also the discussion below), and $\alpha_1 ; \alpha_2$ the performance of α_1 followed by that of α_2. A nice feature of PDeL is that most of the paradoxes appearing in SDL are either not expressible or, if they are, not valid (cf. [Mey88], [MW93a]). To be more specific, if we consider (dynamic variants of) the paradoxes 1 - 10 of section 2, we get:

1'.	$O(\alpha \cup \bar{\alpha})$	*not* valid in PDeL
2'.	$O\alpha \rightarrow O(\alpha \cup \beta)$	valid in PDeL
3'.	$(P\alpha \vee P\beta) \leftrightarrow P(\alpha \vee \beta)$	valid in PDeL
4'.	$F\alpha \rightarrow F(\alpha \& \beta)$	valid in PDeL
5'.	not expressible in PDeL	
6'.	Chisholm's paradox: see the discussion in the next section.	
7'.	Forrester's paradox of gentle murder: see the discussion in the next section.	
8'.	$\neg(O\alpha \wedge O\bar{\alpha})$	*not* valid in PDeL
	$O\alpha \rightarrow P\alpha$	*not* valid in PDeL
9'.	the first two not expressible in PDeL	
	$\neg\varphi \rightarrow (\varphi \rightarrow O\alpha)$	valid in PDeL
10'.	not expressible in PDeL	

Thus we see that 1' and 8' are not valid in PDeL; we still are left with Ross' paradox and the related ones of free choice permission and the penitent 2' - 4') (However, many authors do not really see them as paradoxes once one realizes the mismatch between the natural language reading and the formal semantics of these expression, cf. section 6. Nevertheless, for readers interested in attempts how to overcome even these paradoxes formally we refer to [Mey92], [WM93a], [DMW94a].) We can say something sensible about contrary-to-duty imperatives involved in the paradoxes of Chisholm's and Forrester's (see next subsection). The third expression of 9' is valid as it is just an instance of propositional logic (and particularly the paradox of the material implication) again. Finally, some words about the paradoxes of SDL that not even have direct counterparts in PDeL (5', 9' (first two), 10'). It is perhaps slightly too easy to just say these are not expressible, so there is no paradox any more. One might, for instance, introduce a dynamic counterpart of the logical implication, as follows:

$$\alpha \gg \beta$$

with as reading "action α involves action β", and as formal semantics: $R_\alpha \subseteq R_\beta$: all states reachable by doing action α are also reachable by doing action β. This means, of course, that all results of action β are also results of action α, so that we have the following proposition:

PROPOSITION. $\alpha \gg \beta$ implies the validity of $[\beta]\varphi \rightarrow [\alpha]\varphi$ for all assertions φ.

To give an informal example, "murdering someone gently" involves "murdering someone". (Cf. the discussion of Forrester's paradox in the next subsection.) We refer to PDeL extended with the operator » as PDeL(»).

Now we can express dynamic counterparts of 5':

5'. $\alpha \gg \beta \vdash O\alpha \rightarrow O\beta$. Whether this is a validity in PDeL(») depends on the interpretation of the negated action $\bar{\alpha}$. As we have discussed in e.g. [WM93] there is not an obvious unique choice for this. A possible choice—which, by the way, we purposely did *not* follow in our original paper [Mey88]—would be that the accessibility relation $R_{\bar{\alpha}}$ associated with $\bar{\alpha}$ is the set-theoretical complement of that (R_α) associated with α, i.e., $R_{\bar{\alpha}} = (S \times S) \setminus R_\alpha$, where S is the set of all possible worlds. In this case we have that $\alpha \gg \beta$ implies $\bar{\beta} \gg \bar{\alpha}$ (since now $\alpha \gg \beta \Leftrightarrow R_\alpha \subseteq R_\beta \Leftrightarrow R_{\bar{\beta}} \subseteq R_{\bar{\alpha}} \Leftrightarrow \bar{\beta} \gg \bar{\alpha}$). In its turn this implies that under this interpretation of negated action we have that 5' is a validity (since $\alpha \gg \beta \Rightarrow \bar{\beta} \gg \bar{\alpha} \Rightarrow \models [\bar{\alpha}]V \rightarrow [\bar{\beta}]V \Rightarrow \models O\alpha \rightarrow O\beta$). Note, however, that this is only true in this particular interpretation of $\bar{\ }$.

However, we have that the first two cases of 9' and that of 10' are still not expressible. Note that the obvious attempts, viz. $O\alpha \rightarrow O(\beta \gg \alpha)$, $F\beta \rightarrow O(\beta \gg \alpha)$, and $(O\alpha \wedge O(\alpha \gg \beta)) \rightarrow O\beta$ are not well-formed in PDeL(»), since $O(\alpha_1 \gg \alpha_2)$ is not (O should have an action as an argument). So to deal with this we should really construct a hybrid logic of both ought-to-do and ought-to-be operators. We have done some work in this direction (see [AMW93]), but this would be beyond the scope of the present paper.

5.2 Case Study: Contrary-to-Duty Imperatives

One of the most serious paradoxes in (standard) deontic logic involves the notion of *contrary-to-duty imperatives*. These have to do with the specification of norms in case some other norm(s) have already been violated. The best-known example is the one given by Chisholm ([Chi63]): consider the following statements in natural language:

(i) You ought to go to the party

(ii) If you go to the party, you ought to tell you're coming

(iii) If you don't go, you ought not to tell you're coming

(iv) You don't go to the party.

Intuitively, this set of statements is perfectly understandable and consistent, and none of the four statements seems to be redundant in the set. However, if we try to represent this set in SDL, we run into serious trouble. A more or less natural way to represent the Chisholm set in SDL is the following:

(i') Op

(ii') $O(p \rightarrow q)$

(iii') $\neg p \rightarrow O\neg q$

(iv') $\neg p$

In SDL this set (i') - (iv') is inconsistent, contrary to the intuitions about (i) - (iv):

In SDL we have as a theorem $O(p \rightarrow q) \rightarrow (Op \rightarrow Oq)$, which together with (ii') yields Oq. On the other hand, (iii') and (iv') give $O\neg q$. And in SDL $Oq \wedge O\neg q$ is inconsistent.

Note furthermore that it is not really coincidental that (ii) and (iii) are represented in a different way (which may be questionable, of course). If we would, for instance, replace (iii') by

(iii") $O(\neg p \rightarrow \neg q)$

we have that (iii") is derivable from (i') in SDL, which does no justice to the intuition that all of (i) - (iv) are independent from each other. Similarly, if we would replace (ii') by (ii") $p \rightarrow Oq$, this statement (ii") would be derivable from (iv'), contrary to the intuitions concerning independence.

Here we like to show how the dynamic perspective and particularly PDeL can help us solve the problems with the Chisholm set. To analyse the problem in PDeL we need to be a little more specific about the order in which the actions take place. (This distinction does not really occur in the SDL representation since here everything is formalized in a static way.) We can in fact distinguish three versions of the Chisholm set. We here use rather abstract versions. See the thesis of Tina Smith ([Smi94]) for some very nice daily-life instances.

1. The *"forward"* version of the Chisholm set.

(1i) it is obligatory to do α

(1ii) if you do α, you have to do β *afterwards* (i.e., *after* α)

(1iii) if you don't do α, you have to do non-β
(1iv) you don't do α

This version is the easiest one to formalize in PDeL. We immediately can represent it as follows:

(1i') $O\alpha$
(1ii') $[\alpha]O\beta$
(1iii') $[\bar{\alpha}]O\bar{\beta}$

(Remark: the fourth premise of the set, which expresses that some action is actually performed, cannot be represented in PDeL. In some sense, statements of actions in PDeL and the underlying dynamic logic is of a hypothetical nature: "if one (would) perform the action, the following holds". The implication implicit in an formula $[\alpha]\varphi$ is therefore more like a conditional in conditional logic. As such, it is not really important what actually happens. Here and in the sequel we shall just ignore the fourth assertion in the formal representation.)

In PDeL one may derive from this representation that it holds that $O(\alpha\ ;\ \beta) \wedge [\bar{\alpha}](V \wedge F\beta)$, in other words: it obligatory to perform the sequence α followed by β, and moreover, if α has not been done, one is in a state of violation but nevertheless also forbidden to do β (see [Mey88]). This is exactly as one would expect.

2. The *"parallel"* version of the Chisholm set

(2i) it is obligatory to do α
(2ii) you have to do β *while* α is being done
(2iii) if you don't do α, you have to do non-β
(2iv) you don't do α

In fact, this version is very much related to another infamous paradox involving contrary-to-duties, viz. Forrester's. This is also known as the paradox of the gentle murderer:

(F1) One is forbidden to commit murder
(F2) Still, if one murders someone, one should do so gently
(F3) Jones murders someone.

In SDL this is a big problem again. There one would use a formalization like (using m for committing murder and g for committing a murder gently):

(F1') Fm
(F2') $m \rightarrow Og$
(F3') m
together with the implicit necessary truth
(F4') $g \rightarrow m$

Now we can derive: (F4':) $g \rightarrow m \vdash_{SDL} O(g \rightarrow m) \vdash_{SDL} Og \rightarrow Om$, so that by (F3') and (F2') we arrive at Om, which, apart from being quite absurd, is inconsistent (in SDL) with (F1')! Castañeda dubbed Forrester's paradox as the "deepest" in deontic logic ([Cas84]): the main reason being that distinguishing between assertions and actions ("*practitions*") did not help in his approach to deal with it. He placed emphasis on the fact that there is an *adverb* ("*aspect*"), viz. gently, involved. However, we believe that the problem is the *simultaneity* of actions: one should be gentle (or rather act gently) *while* murdering (see also [Mey87]). if we would consider a "forward" variant of Forrester's set, we get something which is very easy to formulate in PDeL:

e.g.
(a) You are forbidden to go
(b) Yet, if you go, you have to close the door *afterwards*

In PDeL, using obvious abbreviations:

(a') Fg
(b') [g]Oc

In fact, this is "one half" of the forward Chisholm set (viz. isomorphic to the set {(1i'), (1iii')}, since Fg is equivalent with $O\bar{g}$, so that taking $\alpha = \bar{g}$ results in the aforementioned set). Now, however, we have a parallel version, for which the above representation is erroneous. As we stated in [Mey87], in some sense the Forrester paradox (but also the parallel version of the Chisholm paradox) is the hardest to represent in PDeL. As was shown there the intuitive representation

(F1")	Fm	it is forbidden to murder
(F2")	$O(\bar{m} \cup g)$	one ought to not-murder or murder gently

is not adequate either, since in PDeL the latter (which is equivalent with $F(m \& \bar{g})$) is derivable from the first, which contradicts the intuition. But, as we indicated in [Mey87] too, there is an easy way out by using multiple violation atoms (as we have also been using frequently in subsequent papers). In short, assuming that we have a set $\{V_i\}_i$ of distinct violation atoms and using the abbreviations $F_k\alpha \leftrightarrow_{def} [\alpha]V_k$ and $O_k\alpha \leftrightarrow_{def} \neg F_k\alpha$ we can now adequately represent both Forrester's set as

(F1''') F_1m
(F2''') $F_2(m \,\&\, \bar{g})$

which has as a desirable consequence $[m \,\&\, \bar{g}](V_1 \wedge V_2)$, i.e., after murdering someone non-gently one is guilty of two offences: having murdered someone and having been not gentle while murdering,

and the parallel version of the Chisholm set as:
(2i') $O_1\alpha$ one ought to do α
(2ii') $F_2(\bar{\beta} \,\&\, \alpha)$ it is forbidden to do both α and non-β at the same time
(2iii') $F_3(\beta \,\&\, \bar{\alpha})$ it is forbidden to do both β and non-α at the same time

3. The *"backward"* version of the Chisholm set. This is the original one, as formulated by Chisholm:

(3i) it is obligatory to do α
(3ii) if you do α, you have to do β *first* (i.e., *before* α)
(3iii) if you don't do α, you have to do non-β (first)
(3iv) you don't do α

This is perhaps the more difficult one to express in PDeL, since PDeL does not contain an operator of the kind "if you do α, you have to do β *first* ". Surely again, if one would use the same representation as for the forward version, one does not get the intended result, e.g. (2ii) would be misrepresented. In [Mey87] it is argued that the best representation in PDeL is:

() $O_1\alpha$ one ought to do α
() $F_2(\bar{\beta} \,;\, \alpha)$ it is forbidden to do non-β followed by α
() $F_3(\beta \,;\, \bar{\alpha})$ it is forbidden to do β followed by non-α

See [Mey87] for a more elaborate discussion on this representation. We admit that it would be far nicer to have a representation closer to the natural language representation, but this would call for a non-trivial extension of PDeL, in which one also can reason "backward" directly.

Finally in this section, we must also make some qualifying remarks about the dynamic perspective. Although we feel that some issues have definitely been clarified as discussed above, it cannot be said, of course, that this perspective solves *all* problems of deontic logic. For instance, in [AMW93] we discussed the problem of ought-to-be's. However you view this topic, in the end you still have to do with a static notion of obligation (the obligation of a desired state-of-affairs), which is needed, too, for practical purposes in system specification (cf. [AMW93]), but must be solved without resort to dynamic interpretations. However, as we shall argue in the next section, in these practical contexts we can also look at the problems from the other perspective that is offered by computer science, viz. the *pragmatic* perspective.

6. The Pragmatic Perspective

When one is interested in the specification of concrete systems one can imagine to be interested in different problems than one is analyzing deep philosophical problems. This appears also to be the case when one is dealing with normative issues. Specifying normative systems, systems in which norms play an important role, requires a more pragmatic view on matters pertaining to reasoning about norms. It has now been recognized by several authors that deontic logic may be useful for normative system specification, but which deontic logic is best suited for this purpose? It appears to be the case that the most important role of the logic to be used is to accommodate for the need to distinguish between *ideal* and *actual* behaviour of the system. So, in the ought-to-be setting, we would use $O\varphi$ to mean "φ is desirable" or "ideally φ is the case" rather than "φ is obliged". We would then have the possibility to specify what is to be done if the actual behaviour deviates from the ideal or desirable one. This enables us to specify fault-tolerant behaviour of the software or system at hand. When $O\varphi$ is read as "ideally φ" even the standard system SDL or Anderson's variant of this is useful: most paradoxes lose their bite and become perfectly understandable and intuitive properties of "ideality". So in this case we need not bother too much about deep problems!

For example, one may use the following clause in a Library KB:

$$[(\text{borrow}(p, b)]\ O(\text{return}(p, b))_{\leq 21d}$$

(when a person p borrows a book b, he should return it within 3 weeks), or

$$V_{\text{return}(p,\ b)} \rightarrow O(\text{pay}(p, \$2, b),$$

i.e., when a person fails to fulfil his obligation to return the borrowed book, he has to pay a fine of $2.

Sometimes also database integrity constraints (that *per se* have nothing to do with norms) are regarded as deontic (e.g. [Kwa93]):

$$\mathrm{DB} \models \mathrm{O(IC)}$$

stating that the database must satisfy (ideally satisfies) the integrity constraints IC, so that, when violated, integrity-recovering actions can be specified.

Occasionally even software specification is considered deontically ([Coe93]). For instance, the specification of a program that computes the value of N! and stores it in variable x, while in case of an overflow arising the variable of x is set to 0, might look like this using a form of dyadic deontic logic ([vW64], [MW93a]):

$$\mathrm{O}(x = N!) \wedge \delta(\mathrm{ovf})\mathrm{O}(x{=}0).$$

However, as Coenen remarks, this might be too easy a specification since a "lazy programmer" may just implement a program that raises an overflow and sets x to 0, while still satisfying this specification. This shows also the down side of pragmatics, raising new problems, and an adequate solution to these is still a matter of research.

7. Discussion and Conclusion

In the previous section we argued that in some cases of normative system specification we need not bother about the problems of deontic logic too much and just use it! However, not all old and traditional "philosophical problems" can be eliminated this easily. Especially in AI systems dealing with (common sense) norms (such as legal expert systems, but also natural language interfaces) we need common sense notions of obligation, prohibition and permission, which are very close to—if not the same as—the notions of deontic logic as studied by philosophers. Here we encounter most (or all) problems with the paradoxes *a fortiori*, since now these problematic notions should be made precise *up to the level of implementation in a programming language*. Here, the whole philosophical literature on deontic logic is relevant again, but even not nearly sufficient! Issues of implementation and computation should also be dealt with, while, for instance, modal logic programming (modal resolution) is still in its infancy.

In conclusion, we may say that seeing things from a stand-point of computer science may provide new insight(s) to the traditional problems with deontic logic, where one may adhere to a dynamic point of view considering actions as first-class citizens within the framework, and considering the issue of defeasibility of norms. Moreover, the pragmatic view of specifying concrete systems may overstep / overcome some of the traditional problems of deontic logic. However, computer scientists also may (or even need to) profit from the vast philosophical literature on deontic logic and reasoning with norms in general. While this literature is less relevant for some applications, it appears to be most relevant for some other applications that include common sense reasoning with norms, although not yet sufficient to deal with the new typically computational and implementational problems they encounter.

References

[AM81] C.A. Alchourrón & D. Makinson, Hierarchies of Regulations and Their Logic, in: New Studies in Deontic Logic (R. Hilpinen, ed.), Reidel, Dordrecht / Boston, 1981, pp. 125-148.

[AMW93] P. d'Altan, J.-J. Ch. Meyer & R.J. Wieringa, An Integrated Framework for Ought-to-Be and Ought-to-Do Constraints, VU-Report IR-342, Amsterdam, 1993; extended abstract in Working Notes AAAI Spring Symposium on Reasoning about Mental States, Stanford University, Palo Alto, pp. 48-57; preliminary version of full paper in: Working Papers of the Workshop on Deontic and Non-Monotonic Logics (Y.H. Tan, ed.), EURIDIS, Rotterdam, 1993, pp. 14-45.

[Åqv84] L. Åqvist, Deontic Logic, in: Handbook of Philosophical Logic, Vol. II (D. Gabbay & F. Guenthner, eds.), Reidel, Dordrecht, 1984, pp. 605-714.

[BM88] J.W. de Bakker & J.-J. Ch. Meyer, Metric Semantics for Concurrency, BIT 28, 1988, pp. 504-529.

[vB93] J.F.A.K. van Benthem, Logic and the Flow of Information, in: D. Prawitz et al., eds., Proc. 9th Int. Congress on Logic, Methodology and Philosophy of Science, North-Holland, Amsterdam, 1993.

[BW90] J.C.M. Baeten & W.P. Weijland, Process Algebra, Cambridge University Press, 1990.

[BK86] J.A. Bergstra & J.W. Klop, Algebra of Communicating Processes, in: Math. & Comp. Sc. I (J.W. de Bakker, M. Hazewinkel & J.K. Lenstra, eds.), CWI Monographs 1, North-Holland, Amsterdam, 1986, pp. 89-138.

[Cas81] H.-N. Castañeda, The Paradoxes of Deontic logic: The Simplest Solution to All of Them in One Fell Swoop, in: R. Hilpinen (ed.), New Studies in Deontic Logic: Norms, Actions and the Foundation of Ethics, Reidel, Dordrecht, 1981, pp. 37-85.

[Cas84] H.-N. Castañeda, Aspectual Actions and the Deepest Paradox of Deontic Logic, Davidson Conference, Rutgers University, April 28, 1984.

[Che80] B.F. Chellas, *Modal Logic: An Introduction*, Cambridge University Press, Cambridge / London, 1980.

[Chi63] R,M. Chisholm, Contrary-to-Duty Imperatives and Deontic Logic, Analysis 24, 1963, pp. 33-36.

[Coe93] J. Coenen, Top-Down Development of Layered Fault-Tolerant Systems and Its Problems—A Deontic Perspective, Annals of Math. and Artif. Intell. 9(1, 2), 1993, pp. 133-150.

[DM90] F.P.M. Dignum & J.-J.Ch. Meyer, Negations of Transactions and Their Use in the Specification of Dynamic and Deontic Integrity Constraints, in: Semantics for Concurrency, Leicester 1990, (M.Z. Kwiatkowska, M.W. Shields & R.M. Thomas, eds.), Springer-Verlag, London/Berlin, 1990, pp. 61-80.

[DMW94a] F. Dignum, J.-J. Ch. Meyer & R.J. Wieringa, Contextual Permission: A Solution to the Free Choice Paradox, in Proc. 2nd Int. Workshop on Deontic Logic in Computer Science (DEON'94), (A.J. Jones & M. Sergot, eds.), Tano A.S., Oslo, 1994, pp. 107-135;

[DMW94b] F. Dignum, J.-J. Ch. Meyer & R.J. Wieringa, A Dynamic Logic for Reasoning about Sub-Ideal States, in Proc. ECAI'94 Workshop "Artificial Normative Reasoning" (J. Breuker, ed.), Amsterdam, 1994, pp. 79-92.

[For84] J.W. Forrester, Gentle Murder, or the Adverbial Samaritan, J. of Philosophy 81(4), 1984, pp. 193-197.

[vG90] R.J. van Glabbeek, Comparative Concurrency Semantics, with Refinements of Actions, Ph. D. Thesis, Free University, Amsterdam, 1990.

[Har79] D. Harel, First-Order Dynamic Logic, Springer-verlag, Berlin, 1979.

[Hil93] R. Hilpinen, Actions in deontic Logic, in: J.-J. Ch. Meyer & R.J. Wieringa (eds.), Deontic Logic in Computer Science: Normative System Specification, John Wiley & Sons Ltd., Chichester, 1993, pp. 85-100.

[Hoa69] C.A.R. Hoare, An Axiomatic Basis for Computer Programming, Communications ACM 12, 1969, pp. 576-580.

[HMT94] W. van der Hoek, J.-J. Ch. Meyer & J. Treur, Formal Semantics of Temporal Epistemic Reflection, extended abstract in pre-proc. META'94 (4th Int. Workshop on Meta Programming in Logic), Techn. Report TR-7/94, Univ. di Pisa; to appear in Proc. META'94.

[Hor93] J.F. Horty, Deontic Logic as Founded on Nonmonotonic Logic, Annals of Math. & Artif. Intell. 9(1, 2), 1993, pp. 69-91.

[Jon93] A.J.I. Jones, Towards a Formal Theory of Defeasible Deontic Conditionals, Annals of Math. & Artif. Intell. 9(1, 2), 1993, pp. 151-166.

[JS93] A.J.I. Jones & M. Sergot, On the Characterization of Law and Computer Systems: The Normative Systems Perspective, in: J.-J. Ch. Meyer & R.J. Wieringa (eds.), Deontic Logic in Computer Science: Normative System Specification, John Wiley & Sons Ltd., Chichester, 1993, pp. 275-307.

[Krö87] F. Kröger, Temporal Logic of Programs, Springer-Verlag, Berlin, 1987.

[Kwa93] K.L. Kwast, A Deontic Approach to Database Integrity, Annals of Math. and Artif. Intell. 9(1, 2), 1993, pp. 205-238.

[LHM94] The Dynamics of Default Reasoning: Acting in the Absence of Complete Information, Techn. Report Utrecht University, in preparation.

[Luk90] W. Lukaszewicz, Non-Monotonic Reasoning, Formalization of Commonsense Reasoning, Ellis Horwood, New York, 1990.

[MP92] Z. Manna & A. Pnueli, The Temporal Logic of Reactive and Concurrent Systems: Specification, Springer-Verlag, Berlin, 1992.

[MT93] V.W. Marek & M. Truszczynski, Nonmonotonic Logic, Context-Dependent Reasoning, Springer-Verlag, Berlin, 1993.

[Mey87] J.-J.Ch. Meyer, A Simple Solution to the `Deepest' Paradox of Deontic Logic,
in Logique et Analyse 117-118 (1987), pp. 81-90; also in Proc. 6th Amsterdam Colloquium (1987), pp. 281-290, under the title "The Ethics of Murder".

[Mey88] J.-J.Ch. Meyer, A Different Approach to Deontic Logic: Deontic Logic Viewed as a Variant of Dynamic Logic, Notre Dame J. of Formal Logic 29(1), (1988), pp. 109-136.

[Mey92] J.-J.Ch. Meyer, Free Choice Permissions and Ross's Paradox: Internal vs External Nondeterminism, in Proc. 8th Amsterdam Colloquium (1991) (P. Dekker & M. Stokhof, eds.), ILLC, Univ. of Amsterdam, Amsterdam, 1992, pp. 367-380.

[MH94] J.-J. Ch. Meyer & W. van der Hoek, Epistemic Logic for AI and Computer Science, Cambridge University Press, in preparation.

[MWW89] J.-J.Ch. Meyer, H. Weigand & R.J. Wieringa, A Specification Language for Static, Dynamic and Deontic Integrity Constraints, in Proc. MFDBS 89, J. Demetrovics & B. Thalheim (eds.), Visegrád, Hungary, LNCS 364, Springer, Berlin, 1989, pp. 347-366.

[Mil80] R. Milner, A Calculus of Communicating Systems, Lect. Notes in Comp. Sc. 92, Springer-Verlag, Berlin, 1980.

[MW91] J.-J. Ch. Meyer & R.J. Wieringa, Actor-Oriented System Specification with Dynamic Logic, in Proc. TAPSOFT/CCPSD '91 (S. Abramsky & T.S.E. Maibaum, eds.), Brighton, LNCS 494, Springer, 1991, pp. 337-357,

[MW91a] J.-J. Ch. Meyer & R.J. Wieringa (eds.), Proceedings of the First International Workshop on Deontic Logic in Computer Science (DEON'91), VU Amsterdam, 1991, 412 p.

[MW93] J.-J. Ch. Meyer & R.J. Wieringa (eds.), special issue Annals of Mathematics and AI "Deontic Logic in Computer Science" (Vol. 9(I-II)), J.C. Baltzer AG, Basel, 1993.

[MW93a] J.-J. Ch. Meyer & R.J. Wieringa, Deontic Logic: A Concise Overview, in: J.-J. Ch. Meyer & R.J. Wieringa (eds.), Deontic Logic in Computer Science: Normative System Specification, John Wiley & Sons Ltd., Chichester, 1993, pp. 3-16.

[MW93b] J.-J. Ch. Meyer & R.J. Wieringa (eds.), Deontic Logic in Computer Science: Normative System Specification, John Wiley & Sons Ltd., Chichester, 1993, xiv + 317 p.

[Pra93] H. Prakken, An Argumentation Framework in Default Logic, Annals of Math. & Artif. Intell. 9(1, 2), 1993, pp. 93-132.

[Pra94] H. Prakken, Two Approaches to Defeasible Reasoning, in: Proc. DEON'94 (A.J.I. Jones & M.J. Sergot, eds.), Tano A.S., Oslo, 1994, pp. 281-295.

[PS94] H. Prakken & M.J. Sergot, Contrary-to-Duty Imperatives, Defeasibility and Violability, in: Proc. DEON'94 (A.J.I. Jones & M.J. Sergot, eds.), TANO, Oslo, 1994, pp. 296-318.

[Rya93] M. Ryan, Towards Specifying Norms, Annals of Math. & Artif. Intell. 9(1, 2), 1993, pp. 49-67.

[Seg89] K. Segerberg, Bringing It About, J. of Philos. Logic 18(4), 1989, pp. 327-347.

[Smi94] T. Smith, Legal Expert Systems: Discussion of Theoretical Assumptions, Ph.D. Thesis, Utrecht University, 1994.

[vdT94] L.W.N. van der Torre, Violated Obligations in a Defeasible Deontic Logic, Proc. ECAI'94 (A.G. Cohn, ed.), John Wiley & Sons Ltd., Chichester, 1994, pp. 371-375.

[WM91] R.J. Wieringa & J.-J. Ch. Meyer, Actor-Oriented Specification of Deontic Integrity Constraints, in Proc. MFDBS '91 (B. Thalheim, J. Demetrovics & H.-D. Gerhardt, eds.), Rostock, LNCS 495, Springer, 1991, pp. 89-103.59.

[WM93] R.J. Wieringa & J.-J. Ch. Meyer, Applications of Deontic Logic in Computer Science: A Concise Overview, in J.-J. Ch. Meyer & R.J. Wieringa (eds.), Deontic Logic in Computer Science: Normative System Specification, John Wiley & Sons Ltd., Chichester, 1993, pp. 17-40.

[WM93a] R.J. Wieringa & J.-J. Ch. Meyer, Actors, Actions, and Initiative in Normative System Specification, Annals of Mathematics and Artificial Intelligence 7, 1993, pp. 289-346.

[WMW89] R.J. Wieringa, J.-J.Ch. Meyer & H. Weigand, Specifying Dynamic and Deontic Constraints, Data & Knowledge Engineering 4(2), 1989, pp. 157-190.

[WWMD91] R.J. Wieringa, H. Weigand, J.-J.Ch. Meyer & F.P.M. Dignum, The Inheritance of Dynamic and Deontic Integrity Constraints, Annals of Math. and Artif. Intell. 3, 1991, pp. 393-428.

[Win82] G. Winskel, Event Structure Semantics for CCS and Related Languages, in: Proc. 9th ICALP, Aarhus (M. Nielsen & E.M. Schmidt, eds.), Lect. Notes in Comp. Sc. 140, Springer-Verlag, Berlin, 1982, pp. 561-576.

[vW51] G.H. von Wright, Deontic Logic, Mind 60, 1951, pp. 1-15.

[vW64] G.H. von Wright, A New System of Deontic Logic, *Danish Yearbook of Philosophy* 1, 1964, pp. 173-182.

[vW81] G.H. von Wright, On the Logic of Norms and Action, in: R. Hilpinen (ed.), New Studies in Deontic Logic: Norms, Actions and the Foundation of Ethics, Reidel, Dordrecht, 1981, pp. 3-35.