The Knowledge Engineering Review, Vol. 11:3, 1996, 289–292

Third international workshop on deontic logic in computer science

MARK A. BROWN

Philosophy Department, Syracuse University, NY, USA 13244

JOSE CARNO

Department of Mathematics, Instituto Superior Tecnico, Av Rovisco Pais, 1096 Lisboa Codex, Portugal

1 The workshop (ΔEON'96)

The Third International Workshop on Deontic Logic in Computer Science (ΔEON'96) took place in Sesimbra, Portugal, from 11–13 January 1996. It consisted of 12 refereed technical presentations and four invited talks. The invited speakers were Nuel Belnap (Pittsburgh University, USA), Andrew Jones (Oslo University, Norway), Krister Segerberg (Uppsala University, Sweden) and Marek Sergot (Imperial College, UK).

Only 12 technical presentations were accepted to allow ample time for discussions among the participants—one of the main goals of this interdisciplinary workshop. The technical presentations, collected in (Brown & Carmo 1996), covered topics ranging from the theoretical to the applied. At the theoretical end of the spectrum were presentations on deontic logic and the logic of action, defeasible reasoning, the logical basis for decision, and ethical and legal theories. At the applied end there were presentations on a variety of issues relevant to expert systems in the law, automation of defeasible reasoning, specification of responsibilities and powers in organisations, normative systems specification and confidentiality in database systems.

2 The ΔEON workshops

Deontic logic has to do with the logical study of obligation, permission and prohibition, and other concepts related to the representation of norms and the normative use of language. It has its origins in the analytic study of law and ethics, and naturally some of its earliest applications in computer science and artificial intelligence were connected with the construction of legal expert systems. However, in recent years interest in using deontic logic as a formal tool has also appeared in other areas of computer science and information technology, for reasons discussed later.

Recognition of these connecting interests has led to the decision to organise workshops on this topic at roughly two-year intervals, to promote research and cooperation in this interdisciplinary area. The first workshop (ΔEON'91) was held in Amsterdam in December 1991, and the second (ΔEON'94) was held in Oslo in January 1994. The next (and fourth) workshop is scheduled for January 1998 in Belfast.

Although we may say that the prime objective of these workshops has been to show the usefulness of deontic logic for computer science and related areas, the opposite direction of influence has also manifested itself: the application of other formalisms (such as dynamic logic) to deontic problems, new perspectives on the long-standing "paradoxes of deontic logic", and a greater concern with the automation of deontic reasoning are just some examples. The third workshop also succeeded in promoting the logic of action/agency, recently studied in philosophy but almost unknown outside that community, and showing its usefulness, in combination with deontic logic, in such areas as artificial intelligence, computer science and management science.

3 Deontic logic, conditional obligation and defeasibility

The standard approach to deontic logic sees it as a branch of modal logic, where the necessity operator $\Box$ is interpreted as expressing ethical/legal necessity, i.e., as meaning "it is obligatory that" (and is often denoted by O to suggest this interpretation) and its dual $\Diamond$ ($= \neg\Box\neg$) is interpreted as expressing ethical/legal possibility, i.e., as meaning "it is permitted that" (and often denoted by P), with "it is forbidden that" (denoted by F) construed as $O\neg$. Axiomatically, Standard Deontic Logic (SDL) is the weakest normal modal logic that contains the D schema ($OA \rightarrow PA$). Semantically, SDL is characterised by the class of standard models of modal logic that are serial, with the accessibility relation R in these models interpreted as follows: $w_1 R w_2$ iff w_2 is an ethically/legally ideal version (or a deontic alternative) of w_1. Few systems of logic have been as heavily criticised as SDL: it construes every tautology as obligatory; it does not allow us to express contradictory obligations (since it is a theorem that $\neg(OA \wedge O\neg A)$; $P(A \vee B)$ does not express "free choice permission", since it is not equivalent to $PA \wedge PB$; and SDL gives rise to a series of "paradoxes", in particularly ones arising from the closure of O under entailment.

Although the computer science community (with its more pragmatic outlook) has shown that in some applications some of these problems are not so important, and can be tolerated, there is nevertheless a class of paradoxes that cannot be forgotten in most applications. Consider the following set of sentences: (i) there must be no dog; (ii) if there is no dog, there must be no warning sign; (iii) if there is a dog, there must be a warning sign; (iv) there is a dog. Although intuitively this set seems to be consistent and its members logically independent of one another, it is impossible to represent it in SDL without getting inconsistency or non-independence. This is an instance of a contrary-to-duty paradox, since (iii) expresses an obligation which comes into play only if another obligation (i) has been violated, as expressed by (iv). (Note that we can have more levels of contrary-to-duty obligation: (v) if there is a dog and no warning sign, there must be a high fence; (vi) there is no warning sign.) The paradox just described, and others like it, confront us with the question of how to represent conditional obligations, which combines the general problem of representing conditionals with the more specific problem of how to represent contrary-to-duties. Deontic logic is by its nature concerned with the specification not only of ideal behaviour, but also of behaviour which aims to correct, or repair, situations in which the ideal has been violated. It is precisely this capacity to describe simultaneously ideal and sub-ideal behaviour that has made deontic logic so attractive and useful, as a formal tool, in some applications (addressed in ΔEON'91 and in ΔEON'94) related to the specification of fault-tolerant programs and systems, contracts and some kinds of integrity constraints for databases (that can be violated, in which case "recovery" and/or "penalty" procedures are wanted).

There have been three main kinds of approach to the problems presented by these paradoxes: ones that introduce actions into the logic; ones that consider a temporal dimension; and the ones that defend the view that neither actions nor time are essential to the nature of the paradoxes. These three approaches have been presented, respectively, in the invited talks of Krister Segerberg, Nuel Belnap and Andrew Jones.

From a different perspective, contrary-to-duty obligations may be seen as handling exceptions to (primary) obligations. This suggests connections with the problem of specifying allowable exceptions and default reasoning. However, the connections are not so simple: for instance, when a contrary-to-duty imperative comes into force because of some violation, we do not want simply to derive this (new) actual obligation and to say that the violated obligation has ceased to be in force; we want to be able to recognise that an (original) obligation existed and was violated. The problem of representing and reasoning with conditional obligations and its connections with default reasoning was a central issue of the second ΔEON workshop (ΔEON'94), and was also addressed in this workshop in some of the presentations and in the talk of Andrew Jones.

4 Action/agency logics and normative positions

In the computer science world there has been a great amount of work related to the specification of actions and their effects. In particular, specific modal operators of the form [a]A (meaning that, if

action "a" is executed then after its execution A will be the case) have been proposed to specify the effects of actions. These dynamic logics (as they are known) originally appeared in the programming area (where "a" is a program) and were then generalised to other classes of actions and widely used, e.g., in systems specification. Moreover, they have also been applied in other domains, such as doxastic and epistemic logic (logics of belief and knowledge) and deontic logic (for instance, in Meyer's initial approach "action a is forbidden" was represented as [a]V, where V is an atom expressing the occurrence of a violation (see, e.g., (Meyer & Wieruiga 1993)), and we may say that dynamic logic is probably the most important contribution of computer science to the development of modal logic.

However, underlying dynamic logic we find the assumption that we can represent the relevant actions syntactically (in a "finite way"): in general they are defined from a set of atomic actions by a finite number of "regular" constructions (choice, sequence, iteration, etc.). Although this assumption can be made, with obvious advantages, in many computer applications, it is not clearly appropriate when we try to model the behaviour and interaction of human agents. This has led philosophers to develop other kind of logics where no specific actions are named, and where agent's actions are identified only indirectly by their effects. Such logics use a modal operator of the form E_iA meaning that "agent i effects (or brings about) the state of affairs "A" or "agent i sees to it that A". There have been three main approaches to the semantic definition of such operators: one approach through Boolean combinations of normal modal operators, one using minimal model semantics (which emphasises the role of sets of possible outcomes), and one (Belnap's) using models with branching time, with choice points.

There are important differences between the two action operators [a] and E_i, starting at the conceptual level: the dynamic operator [a] is a kind of conditional operator, whereas the "sees to it" operator E_i is a success operator (in the sense that it validates the T schema $E_iA \rightarrow A$); [a] is centred on the actions and E_i is centred on their results (abstracting from the particular actions that have been performed). These differences affect both semantics and axiomatics: semantically the truth of [a]A is evaluated in the state before the hypothetical execution of the referred action, whereas the truth of E_iA is evaluated in the state after the execution of the relevant action(s); and [a] is a normal modal operator that does not validate schema T, whereas E_i validates T but in general is assumed to be non-normal (in general, it is a theorem that $\neg E_iT$, where T is any tautology, corresponding to the idea that the truth of E_iA involves two components: one positive, stressing that after the actions done by i the truth of A is assured, and one negative, stressing that without the agency of i the truth of A might not have been assured).

There is also a difference in expressive power between the two operators. Using the "sees to it" operator E_i one can express several different positions in which an agent might be with respect to a certain state of affairs A, such as E_iA (did), $E_i\neg A$ (averted), $E_i\neg E_iA$ (refrained), and $\neg E_iA \wedge \neg E_i\neg A$ (remained passive), as well as (at least in some of its logics) notions of control of other agents E_iE_kA (make k do), $E_i\neg E_kA$ (make k avoid), etc. Moreover, combining E_i with deontic operators we can now talk about the different normative positions in which one or more agents might be, and use that to express legal concepts and relations like rights, duties, powers and privileges, as has been done, for example, in (Lindahl 1977) (and in one of the presentations at the workshop).

But the legal domain is not the only one where this might be useful: in computer applications related to implementing system security policies, access control and other normative aspects of human-computer interaction, where we need to interpret concepts like rights, permissions, obligations, authorisation, etc., with great precision, and not merely approximately (note that, for instance, practical possibility, permission and authorisation may mean different things), it is important to have a formal tool where we can express different possible meanings in order to choose the appropriate one. In such cases it is also important to have automatic procedures to develop such different normative descriptions, and that was one of the topics of the talk of Marek Sergot.

Combining logic of agency with deontic logic is also useful in systems for describing organisations, stating contracts and describing or specifying other forms (institutionalised or not) of agent

interaction. Moreover, these operators may also be combined with ones expressing ability, intention, belief and tense, to get powerful logical tools for describing the general behaviour and interaction of agents. Many of the presentations at the workshop concerned this topic, but much more needs to be done with this complex subject and in particular with the notion of "collective agency".

One can even conceive of applications in which we might want to deal with both kinds of action operators. For instance, in systems specification we might think of methodologies, where we at first only specify which state of affairs must be achieved by whom and in reaction to what (without specifying the means by which this is done), and later refine the specification by introducing the concrete possible actions (specified by their component atomic actions, in the fashion of dynamic logic) to specify precisely which actions must be executed, under what conditions and by whom. If strategies like this turn out to be useful then we will need logical systems to combine both approaches to actions: the delta operator of Segerberg is one possibility, but not the only one. The comparison of the two approaches to action was precisely the topic of an extended discussion session that closed the workshop.

5 Conclusions

Several (potential or actual) applications of deontic logic in artificial intelligence, computer science and management have been discussed. (For other concrete applications consult {Wieruiga & Meyer 1993) and the proceedings of the three ΔEON workshops (Meyer & Wieruiga 1993, Jones & Sergot 1994, Brown & Carmo 1996.) In general, we may say that deontic logic is useful for the design and specification of any normative system, seen as any set of interacting agents (humans or computers) whose behaviour can be regarded as norm-governed, where norms prescribe how the agents should behave and specify their rights, but also make provision for the possibility that actual behaviour may deviate from the ideal. Since, at an appropriate level of abstraction, the law, computer systems, businesses, societies, and many other organisational structures may all be seen as instances of, or subject to, normative systems, we may easily see the wide spectrum of potential application of deontic logic.

The development of an information society, in which large numbers of databases—or more generally information systems—are connected in a network and accessed and modified by very large numbers of users, only reinforces the importance of this research. If we want to develop and implement policies to ensure the security and integrity of such systems, it is essential that concepts like rights, permissions, obligations, authority, authorisation, responsibility and delegation, be precisely understood and defined, and not simply treated informally.

We believe that deontic logic and the logics of action, agency and ability, together with epistemic and temporal modalities and with default reasoning, are all important formal tools for the specification of different kinds of normative systems, and that this workshop has provided one more step towards understanding and exploiting their potential.

References

Brown, M and Carmo, J, (eds.), 1996. Deontic logic, agency and normative systems, *ΔEON'96: Third International Workshop on Deontic Logic in Computer Science*, Sesimbra, Portugal, 11–13 January 1996, Springer-Verlag.

Jones, A and Sergot, M, (eds.), 1994. *Workshop Proceedings: Second International Workshop on Deontic Logic in Computer Science; CompLex*, **1/94**, Norwegian Research Centers for Computers and Law, Oslo. (Revised copies of selected papers also appear in a special volume of *Studia Logica*, Volume 57, 1996.)

Lindahl, L, 1977. *Position and Change – A Study in Law and Logic*, Synthese Library 112, Reidel, Dordrecht.

Meyer, JJ Ch and Wieringa, RJ, (eds.), 1993. *Deontic Logic in Computer Science: Normative System Specification*, John Wiley, Chichester.

Wieringa, RJ and Meyer, JJ Ch, 1993. "Applications of deontic in computer science: a concise overview". In: JJ Ch Meyer and RJ Wieringa (eds.), *Deontic Logic in Computer Science: Normative System Specification*, John Wiley, Chichester, pp. 17–40.