12

On the Characterisation of Law and Computer Systems: The Normative Systems Perspective

ANDREW J.I. JONES, MAREK SERGOT

12.1 INTRODUCTION

Because of its origins in the analytic study of law and ethics, it is natural to expect that the earliest and most obvious applications of deontic logic in computer science should appear in the construction of what are sometimes called 'legal expert systems' —systems intended to support in the analysis of legal texts, in drawing consequences from them, and in applying them to actual and hypothetical cases—and indeed there are (numerous) *references* to deontic logic in the literature on legal knowledge representation. However, examples of systems that have made explicit or implicit use of a deontic logic are few and far between; and compared with the mass of published material that is devoted to the representation of law in computer systems (see for instance [Ser90] for a survey), deontic logic has received only sporadic attention.

Originally, our aim in this paper was to address the role of deontic logic in legal knowledge representation. However, we now feel that this question cannot, and should not, be divorced from consideration of a much broader set of questions concerning the

Deontic Logic in Computer Science: Normative System Specification
J.-J.Ch. Meyer and R.J. Wieringa (editors)

role of deontic logic in computer science. This paper is concerned with sketching out some of these broader issues; the topic of legal knowledge representation is included, but is addressed more specifically and in more detail elsewhere [JS92a].

The general position which we here develop and illustrate is that—at the appropriate level of abstraction—law, computer systems, and many other kinds of organisational structure may be viewed as instances of normative systems. We use the term to refer to any set of interacting agents whose behaviour can usefully be regarded as governed by norms. Norms prescribe how the agents ought to behave, and specify how they are permitted to behave and what their rights are. Agents may be human individuals or collections of human individuals, or computer systems or collections of computer systems. Normative systems include systems of law, abstract models of computer systems, and hybrid systems consisting of human and computer agents in interaction. Our particular concern is to indicate the value of adopting the normative systems perspective and to illustrate the roles of deontic logic and the logic of action—in the representation of law, in the formulation of models of computer systems, and in the specification of regulations designed to govern human-computer interaction (for instance, access-control regulations).

The paper is divided into two parts.

In Part I we begin to illustrate our general position more concretely by using a small, but nevertheless rather rich, example taken from the library regulations that govern the borrowing and issuing of books at Imperial College. The essential point for which we argue here is that deontic logic—in one form or other—needs to be taken seriously whenever it is necessary to make explicit, and then reason about, the distinction between what ideally is the case on the one hand, and what actually is the case on the other, or as we also say, between the ideal and the actual. We use the example to illustrate points arising both in legal knowledge representation and in the specification of computer systems. This part of the paper overlaps with the core of our companion paper [JS92a] on deontic logic in legal knowledge representation, which contains more detail and more examples from the genuinely legal domain.

Part II is concerned with applications and further development of the Kanger-Lindahl theory of normative positions. This theory, constructed using the formal tools of deontic logic and the logic of action, is a specific instance of work that has its origins in the study of *legal* positions and *legal* relations, but which can find application in the analysis and representation of many types of organisations and other species of normative system besides law. The presentation of Part II is taken from our earlier paper [JS92b]: the concrete example concerns computer-controlled access to sensitive medical information in a mental hospital, a case study in the field of computer security identified in [Tin90].

12.2 THE EXAMPLE

The main points we wish to make in Part I can be illustrated by reference to one rather small example—a fragment of the Library Regulations at Imperial College (cf. [Ser82]), consisting of four rules and a specification of 'book allowances':

LIBRARY REGULATIONS

1. A separate form must be completed by the borrower for each volume borrowed.
2. Books should be returned by the date due.
3. Borrowers must not exceed their allowance of books on loan at any one time.
4. No book will be issued to borrowers who have books overdue for return to the library.

Book allowances:

Undergraduates	6
Postgraduates	10
Academic staff	20

The small size of the example is an advantage, making the discussion to follow more manageable, and avoiding unnecessary clutter; from this it should not be inferred that the example is too *simple*. It will prove to be rich enough to permit the expression of a number of representational options and subtleties.

Before proceeding, it is necessary to forestall the possible objection that, despite its merits, our example is after all not a piece of *legislation*, so whatever conclusions we may draw by using it could not readily be extended to apply to the genuinely legal domain. To this we would reply that, in common with much of the law, our example contains formulations of norms, designed to regulate the behaviour of individuals—in this case borrowers and library administrators. Furthermore, our example exhibits a number of the features which are often said to be characteristic of the law; although these features will *not* be our focus of attention here, they are worth mentioning at this point, just to help forestall the objection. For instance, the Library Regulations exhibit 'open texture' (When is a book *returned*? What counts as a *member of Academic Staff*?). The Library Regulations are part of a 'seamless web' of regulations, governing the College. (We happen to know that no degree may be conferred on a student who has books overdue for return to the Library.) The Library Regulations do not exist in a vacuum, divorced from a wider context of laws and principles. (The Librarian would not be allowed to fix a smaller allowance of books for black students.) Particular cases may arise, perhaps not explicitly mentioned in the Regulations, on which decisions will have to be made by the administrators of the Library. (Andrew becomes a Visiting Researcher at the Department of Computing—it is decided to classify him as a member of Academic Staff.) And there exist committees authorised to change the Regulations, decide on difficult cases, and hear appeals against allegedly unfair treatment. We find, in short that there are no clear reasons for refusing to accept the applicability of what we are about to say to the genuinely legal domain.

The example has a further advantage in that it is easy to imagine how a computer system might be introduced to automate part of the library's operation. This allows us

to raise a different set of problems: problems that arise in the specification of computer systems.

Imagine the following scenario. The Chief Librarian wishes to improve the efficiency of the library, and asks us to construct a computer representation of the library regulations.

One way to proceed would be to choose some appropriate formal language, use this to formulate a precise representation of what we think the regulations say, and then animate the representation with an automated reasoning system: given a description of some real or hypothetical state of affairs, this computer program would derive what legal, here quasi-legal, consequences would seem to follow. It could be used, for example, to advise on the specific obligations and rights that borrowers and librarians have in given circumstances. This is the kind of program that most immediately comes to mind in discussing artificial intelligence applied to law. It would be an example of a 'legal analysis program', or an 'academic legal expert system' in the terminology of Susskind [Sus87]—'academic' because here there is no attempt to simulate the actual reasoning of an expert lawyer.

But is this the kind of computer system the Chief Librarian has in mind? A legal analysis program might be an effective complement to distributing the library regulations on paper, but still it would describe the library as it *currently exists*; operations in the library would remain as before. Perhaps what the Chief Librarian has in mind is a *different* library, a library where some of the operations are carried out by computer. In the second case, the library regulations are given to us as a description of how the new library ought to function, after computerisation.

At the very least, we must ask the Chief Librarian to choose between two different options. (1) Does he want us to construct a system that advises on the obligations and rights of the various users of the library as it currently exists? Or, (2), does he want us to take the regulations as a specification of how the library ought to function, giving us the task of constructing a computer system which automates the library, at least as regards the issuing and returning of books? We shall examine the second of these two possibilities first, since that will allow us to introduce, in a simple way, a fundamental distinction which will figure prominently throughout: the distinction between the actual and the ideal.

12.3 THE LIBRARY REGULATIONS AS SYSTEM SPECIFICATION

For the first scenario, let us suppose that the Chief Librarian wants us (the system engineers) to introduce a system of computers and other administrative procedures so designed and constructed as to guarantee that the library functions in the way that it should. The Chief Librarian wants us to view the Library Regulations *as a specification of how the system should operate*. Accordingly, we may call this the *systems specification* scenario. The task here facing the system designer is to construct a library system which actually behaves in the way the Library Regulations require.

Taking the Chief Librarian at his word, we might consider how we can *force* actuality and ideality to coincide in this example. Since librarians cannot be forced to perform like automata, and borrowers cannot be trusted to return their books on time, we

might try to shift the entire operation onto a device which can be expected to perform according to our requirements.

Here is a design proposal: first, maintain a database of who borrowed what and when, incorporating 'integrity constraints' to filter out spurious input. Then, acquire the appropriate hardware so that, in order to issue a book, the borrower's library card is first inserted, whereupon the machine calculates the current allowance, either from information stored in the card, or from personal details in the database, or both. If the allowance is already used up, the card is ejected, and will not be accepted again by the machine until either the borrower's book allowance has been increased, or else the borrower returns at least one book. If the allowance is not already used up, the system makes an electronic version of the requested book available to the reader, on appropriate equipment, for precisely the period of time specified by the Chief Librarian. The borrower may exercise his or her right to 'return' the book prior to expiry of the maximum loan period, by tapping in an appropriate message.

If this is what the Chief Librarian wants, and this is what the engineers provide, then we have an example of 'representation of the Library Regulations' only in the sense that conformity to these regulations is implemented in the system—a strategy we shall refer to generally as 'regimentation' in later sections. (The Collins English Dictionary defines the verb 'to regiment' as 'to force discipline or order on, especially in a domineering manner'.) So in the library after regimentation, regulation 1 is obsolete, because forms were part of the old library's system for borrowing books and play no role in the new system. More to the point, regulation 4 is obsolete because it is just not possible to have a book overdue; and what corresponds to regulations 2 and 3 are (at first sight) the facts that books *are* 'returned' by date due and that borrowers *do not* exceed their allowance.

We were given a specification for how the library should function. We designed a computer system accordingly. How can we demonstrate that our proposed implementation meets the Chief Librarian's requirements as expressed by the regulations? How could we at least systematically investigate this claim? Deontic logic has been proposed as a component in formal specification languages for computer systems (see e.g. [ML85, KM87, FM91]). But is deontic logic required here?

An adequate characterisation of what is meant by 'meets the requirements' is in itself far from trivial. (See for instance [Coe91] for a discussion of additional complications that arise when deontic logic is used in the specification of fault tolerance.) Besides this, there are two things that we need to express precisely:
(a) the Chief Librarian's specification of how the library system should be;
(b) our specification of how the computer system should be.

For the Chief Librarian's specification, we would say that a deontic logic is necessary. One immediate reason is that at least one of the regulations (regulation 4) only applies in the case of a violation (of regulation 2): the language we use to formulate this specification must allow for the consistent expression of violation, that is to say, the case where actuality and ideality are different; the distinction between the ideal and the actual is exactly the province of deontic logic.

For specification of our proposed computer system, the need for a deontic logic depends on what the specification encompasses. If we assume that all components of the computer system work flawlessly, and assume that no other kind of violation can occur, or if we choose to ignore the possibility of violation and make no provision for

it in the specification, then there is no need to distinguish between the actual and the ideal, and no need for deontic logic either.

But let us examine our assumptions about the computer system we are proposing. Even for the extreme kind of implementation where regimentation delegates all library operations to automata, *and even if we assume that all these automata work flawlessly*, there is still the possibility of violation, because of the influence of extraneous factors.

Besides the possibility of changes in the regulations (or the Chief Librarian's requirements), here are two examples of violations that can arise in the proposed implementation:

- Andrew leaves Imperial College, while he has books on loan from the library. His allowance goes to zero and is therefore exceeded. Either we admit the possibility of violation, or we must arrange for an extension of the library's computer system (so that access to the library's electronic 'books' is automatically cancelled when a person leaves Imperial College, for example).
- Marek is promoted (or demoted) by the Department of Computing, and his library allowance reduces as a result. (If this is fanciful, we might mention that a research assistant in the Department has just become a full-time PhD student. His allowance has thereby reduced from the 20 books allocated to Academic Staff to the 10 books allocated to postgraduate students.) Whatever, Marek is demoted, his allowance changes, and is exceeded because he has a number of books on loan at the time. Again, either we admit the possibility of violation, or we design the system so that the possibility of violation is eliminated: the library's computer system removes Marek's access to books, or somehow we arrange for it to block the demotion until the right number of books is 'returned'.

We see that, even where we opt for a regimented implementation where everything is performed by automata, the possibility of violation remains:

- because of faults in components of the system;
- because of the influence of extraneous factors beyond the system's control.

If we were now to design the computer system with the possibility of violation in mind, we might wish to include in the specification additional provisions for dealing with cases of violation as they arise; to express these provisions precisely we need a deontic logic.

The first implementation we proposed is very extreme, and might not be acceptable to the Chief Librarian for any number of reasons. In practice, we would probably not regiment the entire operation, but introduce computer systems only for part of it. The methodological points we have been making would still apply.

Here is another design proposal. As before, we maintain a database of who has borrowed what and when, with the appropriate 'integrity constraints'. We also keep the procedure whereby the borrower's library card is inserted and the remaining allowance is computed from details recorded in the database. But now instead of access to electronic books, the borrower is given the actual book, which he can take away. The librarian sweeps the book across an optical reading device; if the allowance is not exceeded and the borrower has no book overdue for return, the system updates the database of what is out on loan and the book is 'issued'. (This is in fact almost exactly the system currently in operation at Imperial College.)

Notice that this library system has no control over the behaviour of borrowers: in particular it cannot guarantee that books are returned by date due. In the context of the previous discussion, the borrowers are either components of the library system who cannot be assumed to perform flawlessly, or extraneous factors over which the library system has no control. The actual behaviour of the librarians is irrelevant (but see section 12.5). Assuming no faults in the computer, no book is issued to a borrower who has a book overdue, or to one who is at the limit of his allowance: any attempt to record such an update will be rejected by the library's database.

How might we implement such a system? As part of the database, we clearly need integrity constraints to filter spurious input. At the very least, an attempted update saying that book b has been issued to borrower x can only be accepted if (i) b is a book in the database (or we trust the optical reading device); (ii) x is a borrower in the database (or we trust the library card); (iii) borrower x is not currently at the limit of his allowance; and (iv) borrower x has no book overdue for return. These conditions could all be expressed with standard integrity constraints.

But consider now the possibility of violation, because of extraneous factors. Suppose Andrew leaves Imperial College while he has a book out on loan, and the librarian attempts to record his departure in the database. The integrity constraint requiring that books can only be on loan to borrowers is violated: either the update is rejected, in which case Andrew's departure cannot be recorded; or the update is accepted, in which case the database enters a state where at least one of its integrity constraints is not satisfied.

Although there are other possible ways of dealing with updates like these, one approach is to introduce 'softer' *deontic* integrity constraints. These are constraints that *ideally* all database states should satisfy; they differ from standard integrity constraints in that we can accept sub-ideal states of the database which do not satisfy deontic integrity constraints, whereas we reject states of the database which do not satisfy the standard ones. (Deontic integrity constraints were suggested in Sergot's earlier discussion of the library regulations [Ser82] but the idea is not developed there. There is also a proposal in [WMW89] for distinguishing a class of deontic integrity constraints though the intended purpose here seems somewhat different from what we have in mind.) We have been led to consider what kind of logic would be adequate for treating the deontic integrity constraints we want. Though the investigation is not complete and we have no space to present the details here, it looks as if a rudimentary deontic logic might suffice for the deontic part: it may even be that the features which make SDL (Standard Deontic Logic) inadequate as a theory of deontic reasoning in general (see section 12.4.2) are not problematic in the context of databases. Note that the checking of deontic integrity constraints would require a (simple) automated theorem prover for whatever deontic logic is used to express them.

In summary we have identified in this section three roles for deontic logic in the *systems specification* scenario.

1. We might require a formal language in which to express precisely the specification of an organisation (here the library together with its computer and other administrative procedures). Such a specification must usually make provision for the possibility of violation, where actual behaviour deviates from the ideal, and for this a deontic logic is necessary.

2. We might require in addition a formal language in which to specify precisely the intended operation of a computer system. A deontic logic is necessary for specifying computer systems if we want to make provision for violations—whether resulting from faulty components or from extraneous factors. We should like to be able to reason with these specifications, for example to test the internal consistency of the specification, or to determine whether one is a logical consequence of another. And—without meaning to give the impression that we underestimate the task—we should *like* to be able to investigate systematically whether the specification of a computer system meets the specification of the organisation into which it is introduced.
3. We might wish to use an automated theorem prover for a deontic logic as a means of *implementing* some of the (software) components of a computer system. We have given one very simple indication of such a use in the remark on the checking of 'soft' deontic integrity constraints in databases. (Another example comes in section 12.4.2.)

We shall return to the specification of computer systems in Part II: section 12.9 refines the discussion of what we have called 'regimentation' and identifies some further requirements and distinctions.

12.4 THE LIBRARY REGULATIONS AS NORMS DIRECTED AT USERS

We turn now to the other scenario, where the Chief Librarian wants us to construct a system which is capable of giving advice to users and librarians concerning their obligations and rights, or more generally of reasoning with a normative representation of the regulations. Of course the example is so small, and the general process of borrowing books from libraries is so familiar, that it is difficult to imagine that anyone would actually want to construct a system of this type for practical purposes. But the points we shall be making are easily extended to more complicated and less familiar domains where such systems have been seen as a valuable aid.

We divide the discussion in this section into two parts, both relating to species of naivety which we believe can be detected in the existing literature. First there is a naive view of where deontic logic is needed; and second, there is a naive view of how to do deontic logic when it is needed.

12.4.1 The Definitional Component

Sergot has observed [Ser82, Ser88] that for many practical purposes, substantial amounts of legislation are, or can be taken to be, essentially definitional in nature. By this is meant at least the following: that many regulations, including many embodied in legislation, may be viewed as *qualification norms*, which specify the conditions under which some entity x counts as an entity of a particular type F. Thus, to cite a well known example in the current literature [SSK+86], the main point of the British Nationality Act 1981 is to spell out the conditions under which an entity (in this case, a person) qualifies as a British citizen. For certain purposes (to be discussed shortly),

the norms may be represented in the following form:

$$british\_citizen(x) \leftarrow born\_in\_UK(x)$$
$$british\_citizen(x) \leftarrow parent\_of(x, y)\ \&\ british\_citizen(y)$$
... and so on.

The actual norms are of course much more complicated than this but the simplified version retains all the features that are presently of interest. Likewise, norms qualifying a person as entitled to Supplementary Benefit [BRRS87] may be cast in the form:

$$entitled\_to\_supp\_ben(x) \leftarrow \quad \ldots \text{(conditions)}$$

And part of the Library Regulations may be viewed as defining borrowing allowances:

$$allowance(x, 6) \leftarrow undergraduate(x)$$
$$allowance(x, 10) \leftarrow postgraduate(x)$$
$$allowance(x, 20) \leftarrow academic\_staff(x)$$

As in the last example, some qualification norms may be seen as specifying the definition of a *relation* but nothing turns on this. For example, the British Nationality Act also lays down the conditions for determining whether two persons fall in the *parent_of* relation; and as a matter of fact, it actually defines four different categories of British citizen:

$$british\_citizen(x, category\_1) \leftarrow \quad \ldots \text{(conditions)}$$
$$\vdots$$
$$british\_citizen(x, category\_4) \leftarrow \quad \ldots \text{(conditions)}$$

Substantial fragments of legislation have been represented as logic programs in this fashion; it is the ability of these programs to deal with the mass of detail that gives them their practical utility. Of course, logic (here predicate logic) is not universally accepted as an appropriate representational formalism even for norms of the qualificatory type. Others have preferred to use other kinds of representational devices and the literature on artificial intelligence and law abounds with a wide variety of proposals. But the points we wish to make here are independent of such considerations. We wish to make three fundamental points about the representation of qualification norms, in whatever formalism.

Our first point concerns the *surface formulation* of qualification norms. Qualification norms are often expressed in legislation and regulations using a form of words where apparently deontic modal auxiliaries occur. ('A person *shall* be a British citizen if....', 'A person *shall* be entitled to Supplementary Benefit if....', 'A postgraduate *shall* have an allowance of 10 books'.) This has led some to suggest (see in particular [Sus87]) that these modalities must be captured somehow, else the representation does not give a faithful account of what the legislation says. We argue almost the opposite. Where legislative texts are taken to express qualification norms, the occurrence of deontic modalities may be *ignored*, for most practical purposes, in the representation. For *which* purposes? Well, obviously, if one is interested in designing a system which will *in fact* classify entities in a manner which *conforms* to the norm stipulated, then

the distinction between what *shall be* and what *is* becomes redundant. And for many practical purposes, in public administration, or in the administration of a library, for instance, one might indeed be interested in achieving just that. There may be simply no point in designing a system which fails to classify x as entitled to Supplementary Benefit even though x ought to be classified as entitled to Supplementary Benefit. And where the job of the system is to *advise on* qualification norms rather than administer them, there may be no point in dealing with the case where x fails to be classified as a British citizen even though x ought to be classified as a British citizen—because there is nothing specific that can be said about this state of affairs. And likewise for the other examples.

This is not to say that deontic modalities in the formulation of qualification norms should always be ignored, for all purposes. Legislation that specifies a definition can sometimes be seen as a directive also: so the British Nationality Act might be seen as expressing an obligation on the judiciary, or on the immigration authorities, or on other citizens, to recognise as a British citizen anyone who satisfies the conditions laid down in the Act. And if the purpose of the representation is to capture this reading then the directive force of the legislation, and the possibility of violation, cannot be ignored. But this has nothing to do with the form of words used in the text. The same would apply if a different form of words had been chosen and no modal auxiliary appeared in the text. It is naive to expect a meaningful representation of a legislative text to emerge by mechanically replacing all occurrences of the word 'and' by truth-functional conjunction; it is no less naive to replace all occurrences of 'shall' or 'must' by the obligation symbol of a deontic logic.

The second point we wish to make concerns the *limitations* of restricting attention to norms of the qualificatory type. The point is a very obvious one, but we feel it needs to be made explicitly to avoid misunderstandings. A representation of the conditions under which a person is (or ought to be) classified as a British citizen is no more than that. It does not attempt to say what follows from the classification of a person as a British citizen: what package of rights, or obligations, or duties or privileges British citizenship confers on an individual. But then neither does the British Nationality Act. A more complete representation of the British legal system might be constructed that represents in addition the norms expressed in the Immigration Act or in the (unwritten) Constitution—but construction of all this would be a separate exercise. Similarly, a representation of the qualification norms that classify a person as entitled to Supplementary Benefit does not represent what follows from such a classification. And a representation of the definition of borrowing allowances in the Library Regulations does not pretend to be also a representation of the other regulations.

It might be felt that restricting attention to qualification norms and definitions is pointless, because to ignore the existence of norms that refer to these definitions is to ignore everything of any essence. This is a separate consideration, because it concerns the *adequacy* of a representation *relative to its intended purpose*. Clearly, we could not insist that representation of the British Nationality Act is pointless unless every norm in the British legal system of relevance to British citizenship is represented as well. Somewhere a line must be drawn, and a decision must be made about what needs to be represented and what does not. Perhaps the essential problem in determining whether to issue a British passport to a given individual reduces in practice to the problem of determining whether the person ought to be classified as a

British citizen. Perhaps the problem of determining whether to grant Supplementary Benefit requires consideration of the qualification norms only. And perhaps the main problem in administering the Library Regulations is the computation of an individual's borrowing allowance, not reasoning about what the other regulations say. If the last example seems ridiculous, then imagine a system which deals also with the definitions of *undergraduate* and *academic_staff* and takes into account the precedents that have been set by authoritative rulings in the past. As an aside, we believe it is fair to say that the existing literature on artificial intelligence and law has been largely preoccupied with the representation of norms of the qualificatory type, where the main focus has been on questions of vagueness and 'open texture' and the adequacy or propriety of the various techniques that have been proposed. This would certainly account for the—perhaps surprising—fact that so little attention has been paid to deontic logic in 'legal expert systems'.

The third point we wish to make is a *practical suggestion*. The following section (12.4.2) will be concerned with the representation of norms proper, that is to say, norms not of the qualificatory type. What we suggest here is a way of avoiding the representation of genuine norms, and stretching the applicability of techniques that apply only to qualification norms. The suggestion relies on the observation that many of the norms encountered in complex statutes and regulations can be *factored* into two components: a genuinely normative component, and a component that is definitional only.

Consider the following example (which is a simplified form of section 74 of the Road Traffic Act 1972, cited in [TM82]):

> All four-wheeled vehicles shall carry two white lamps at the front.
> All two-wheeled vehicles shall carry one white lamp at the front.

Factoring these two norms yields the following paraphrase:

> All vehicles shall carry the requisite number of white lamps at the front.
> The requisite number of white lamps is:
>
> - for four-wheeled vehicles, two;
> - for two-wheeled vehicles, one.

Now, for some practical purposes, it might be sufficient to represent only the definitional component: 'vehicle', 'white lamp', 'at the front', and now also 'requisite number of lamps'. Of course, the resulting representation would ignore the normative component; but as already discussed, a line must be drawn somewhere, and there are circumstances where restricting attention to a definitional component is enough.

12.4.2 The Normative Component

For certain purposes it would not be realistic to ignore the distinction between what actually happens and what ideally happens, or more simply, between the actual and the ideal. And it may be that the Chief Librarian is not satisfied with a system that restricts itself to a representation of the definition of 'allowance', but requires, in addition, a system that can reason about the permissions and obligations of borrowers and library staff, some of whom, some of the time at least, fail to abide by the rules;

indeed, as already observed, one of the regulations (regulation 4) applies only in the case where a violation has occurred.

So at the very least there will be, for these purposes, a need to mark the distinction between what shall be done and what is done, between what shall be the case and what is the case. Some appear to have suggested that there is no need to do more, that some kind of syntactical marker is enough. For example, MacCormick [Mac90] has proposed distinguishing a special class of predicates which he calls *normative* predicates, but he provides neither an axiomatic nor a model-theoretic characterisation of the logical properties of this special class of predicates. Such a syntactical marker might be of some value as part of the documentation of a representation, but it obviously does not contribute to an analysis of deontic reasoning.

It is one thing to insist that uninterpreted syntactic markers fail to supply an analysis of deontic reasoning; it would be quite inappropriate to go on from that truism to the claim that we must always set our sights on achieving an *elaborate* theory of deontic reasoning. For certain purposes it is conceivable that no more than a very *rudimentary* deontic logic would be required—one which, say, allowed the consistent assertion of 'ought-A' and 'not-A', and the derivation of 'permitted-A' from 'ought-A', but which otherwise remained silent on the logical properties of the deontic modalities. (Perhaps something as elementary as this was all that was required for the ESPLEX system, for instance [BMT87].) But if, on the other hand, something fancier is needed, then there are very good reasons for supposing that some tricky problems can easily emerge.

We outline some of these problems shortly. But it is first worth mentioning that, although there are perhaps as many competing hypotheses about how to deal with these problems as there are deontic logicians, there is nevertheless one point on which there is a good deal of agreement: and this is that Standard Deontic Logic (SDL)—a normal modal system of type KD in Chellas' classification [Che80]—is unable to provide an adequate theory of deontic reasoning. Criticism of SDL has been the point of departure for much recent work in the field, and since the proposed alternatives to SDL often differ markedly, it is not surprising to find that the shortcomings of SDL receive various types of diagnosis in the literature. (See, e.g. [ÅH81, LM83, JP85, McC86, McC83, Mey88].) But here again one point is quite generally agreed upon: that SDL lacks the means for supplying an acceptable analysis of deontic *conditionals*, for instance of sentences of the type 'if A then it ought to be the case that B'.

Much of [Jon90] was concerned with the task of showing that problems with deontic conditionals easily infect rather mundane application areas. In both the Library Regulations and statute law (e.g. in contract law) one finds structures of what Chisholm called the 'contrary-to-duty' type [Chi63], in which the conditional obligation to do B has, as its condition, the violation of some other obligation. Here, for instance, is a translation of parts of two sections of the Norwegian Sale of Goods Act of 1988, slightly simplified:

> The goods shall be delivered within a reasonable period of time following their purchase. [section 9[1]]
>
> If the seller is prevented from completing the transaction within the appropriate time period, he shall inform the buyer of the delay .... If the buyer does not receive this information within a reasonable period of time ... then the buyer may demand compensation for any loss which could have been avoided had he been given reasonable notice of the delay. [section 28]

In [JS92a] we present another, slightly more complicated, example of this type of structure, taken from the Vienna Convention on the International Sale of Goods.

Given the occurrence of structures of these kinds, it is by no means far-fetched to suppose that scenarios of just the type Chisholm described could easily arise in actual practice (the point is more fully developed in [Jon90]). The formal characterisation of such scenarios involves confronting some central problems of deontic logic. The point here is not to make claims about how these problems should be solved: it is simply to indicate that they can easily arise.

In fact it can be argued that genuine problems of deontic logic arise even before one gets to 'contrary-to-duty' conditionals and the Chisholm scenario. For just the matter of deciding on which *detachment principles* to accept for deontic conditionals itself raises questions of a non-trivial kind. Suppose, for instance, that it ought to be that A, and that it ought to be that B given that A. Do we then accept a *deontic detachment* principle by means of which we conclude that it ought to be that B? Or do we say, rather, that the logical conclusion is not that it *actually* ought to be the case that B, but merely that it ought *prima facie* to be the case that B ? But then how is the distinction—if accepted— between *actual* and *prima facie* obligation to be articulated?

Or suppose that it ought to be that B given that A, and that A is *in fact* the case; do we then accept a *factual detachment* principle by means of which we conclude that it ought to be that B ? Would we accept that such an inference can be made *unrestrictedly*, i.e., no matter what else, apart from A, is true of the circumstances at hand? Or should we, rather, view the conditional as holding by default only—as holding only in regard to the typical, non-exceptional circumstances in which A is true, being possibly defeated/overturned in other, exceptional circumstances? An affirmative answer to the latter question would lead the analysis into the thorny problems of default reasoning. But can the logic of deontic conditionals fail to be confronted by such issues? (For further discussion of these points, see [JP91]. For the moment, the only aim here is to indicate how easily the analysis of deontic reasoning leads quickly into deep, in parts uncharted, waters.)

The argument of this section may also be used as a basis for critical response to the central thesis of [Ben89]. For the problems here indicated, concerning the proper analysis of reasoning with and about deontic conditionals, are by no means confined to the domain of so-called 'hard' cases, where there is some lack of clarity regarding what the facts of the case are, or regarding which rules to apply, or both. For even if it *could* be argued that the Chisholm scenario is relatively eccentric, problems about how to define detachment principles for deontic rules are clearly going to arise quite independently of whether particular cases are straightforward or hard.

Our companion paper [JS92a] contains a number of further examples from the legal domain, together with the beginnings of a methodology for the use of deontic logic in legal knowledge representation.

Part II: NORMATIVE POSITIONS

12.5 INTRODUCTION

The Library Regulations, in common with many other sets of regulations, leave a number of questions about the rights and obligations of the participant agents unanswered. For instance, when it is said (regulation 2) that books should be returned by the date due, this is ordinarily taken to mean that a borrower b is under an obligation to return his books on time. But if we look at this regulation from a relational point of view, in terms of the relationship between the borrower b and a librarian a, then it is natural to ask also what the regulation says about the position of librarian a. The librarian a is not under an obligation to return b's books, although presumably a is permitted to do so. And furthermore, in most libraries, the librarian a is obliged not to prevent borrower b from returning books on time. Likewise, if there was a rule saying that b was permitted to borrow books up to the limit of his allowance, this would ordinarily be understood to mean, not only that b is not prohibited from borrowing books, but also that librarian a was under an obligation to issue him the books he asks for, up to his allowance. There are other pairs of agents one could pick on, including the Chief Librarian, or the library authorities, or Imperial College. The point is that the Library Regulations do not completely specify the relative *normative positions* of librarian and borrower, or librarian and library, or borrower and library, vis-à-vis the particular types of acts that are involved in the business of the library.

Now it may appear that, for the example at hand, nobody would ever have a practical interest in mapping out in complete detail the normative positions concerned. Whether or not that is true, there are clearly other contexts in which a complete specification of this kind might be deemed highly desirable. In the context of artificial intelligence and law, for instance, Allen and Saxon in particular have long advocated that an approach in this spirit should be adopted to the analysis of legislation, arguing that there are important nuances and ambiguities that remain undetected otherwise (see e.g. [AS86]). Identical points could be made in the context of computer system specification too. For instance, in those computer systems where the security of sensitive or confidential data is of high priority, it will be essential to provide a precise and exhaustive characterisation of the access rights of various classes of users to this information. This applies to the need for a precise specification of the mechanisms for controlling access to data, as well as to more general constraints pertaining to the organisation into which the computer system is introduced.

In this part of the paper we consider the contribution that can be made to such questions by formal methods developed originally in the analytic study of law, using the tools of deontic logic and the logic of action. Our main source of examples will be a series of questions raised by Ting in a paper presented at the 1989 IFIP WGII.3 Workshop on Database Security [Tin90]. Ting discusses the case where a computerised database of patients' medical records was introduced into a mental health institution and the security requirements that were thrown up by this application. His main theme is that, in such applications, the design of the computer system's access control mechanisms cannot be divorced from more general considerations concerning the operation of, in this case, the hospital; his observation is that the relationships between the organisational and computer-technical requirements are poorly understood; and

his main concern is that there is an urgent need for a formal framework in which these requirements can be expressed and analysed precisely.

Our specific proposal is to apply the Kanger-Lindahl theory of normative positions to the analysis and specification of security policies of the type identified by Ting. More precisely, our proposal is to apply the *method* by which these theories are constructed rather than their results directly. The theory of normative positions seeks to explicate what the legal theorist Hohfeld called the 'fundamental legal conceptions'—duty, right, privilege, immunity, power, and so on, leading to questions of authority and responsibility. The work falls in the jurisprudential tradition of Austin, Bentham and Hohfeld; the pioneering contribution of Kanger was to apply formal techniques—deontic logic and the logic of action—to the analysis of these complex notions. Lindahl develops Kanger's theory further; his exposition [Lin77] provides an excellent introduction to the general area.

Ting's example was identified and brought to our attention by Philip Morris, in response to a suggestion (by Jones) that the Kanger-Lindahl theories could make a valuable contribution to the solution of specification problems arising in computer security. In [MM91] Morris and McDermid present Kanger's version, in which there are 26 distinguishable notions of 'right', together with a proposed way of applying the results of this theory to express access rights in computer systems. This proposal, which requires the right-granting authority to be identified, is not the way we would have chosen to present or apply Kanger's results. But much more importantly than this, we believe that the main contribution of the Kanger-Lindahl framework to software engineering lies not in their specific results, but in the method they use to construct their theories. The point, as we see it, is not that there are 26 distinguishable notions of 'right' (in Kanger's theory of two-agent rights relations), or 35 (in Lindahl's individualistic two-agent theory), or 127 (in Lindahl's collectivistic two-agent theory), or whatever. The fundamental point is that there is a formal framework, and a systematic procedure, for undertaking an analysis of at least some of the key questions raised by Ting concerning the specification of access rights.

Part II is organised as follows. Section 12.6 presents our variant of the Kanger-Lindahl method, and uses an example taken from Ting to motivate the development and to illustrate some of the complexities that can be brought out. Section 12.7 identifies some further refinements to the formal language that we plan to incorporate to increase expressive power. Section 12.8 gives a brief description of the software tools we are developing to support practical applications. Section 12.9 makes some further remarks concerning the specification of computer systems and the implementation strategy we referred to as 'regimentation' in Part I.

12.6 THE METHOD

Ting [[Tin90], p5] says:

> "... the patient does not have the right to read his or her own record ..."

Is the formalisation of this sentence a straightforward matter? We shall argue that where a precise and exhaustive characterisation of rights is called for then complexities of various kinds are unavoidable. We shall also argue that there nevertheless exist

formal methods whose application facilitates the systematic investigation of these complexities.

We start by changing the example in two ways: presumably, the patient's reading behaviour will be controlled by exercising control over his or her access to the records, so we first rephrase the example as:

> "The patient does not have the right to have access to his or her own record."

A less trivial change, on which we shall comment at the end of this section, replaces the reference to 'right' by a reference to 'permission':

> "The patient is not permitted to have access to his or her own record."

Let a, b, c, ... be names of individual agents. Let f be a function which creates names from names, and let fa be read "the file or record of a". Let H be a two-place relation between individuals and the files of individuals. So let $aHfa$ be read "a has access to the file of a", $aHfb$ be read "a has access to the file of b", and so on. For immediate purposes, let a name some individual in the category of patient, and let the sentence-letter A abbreviate $aHfa$.

Obviously, there are just two possible 'fact positions' in regard to A:

(F_1) A

(F_2) $\neg A$.

We now construct a class of normative positions employing a modified version of a procedure used by Kanger, Pörn and Lindahl. Some specific references to this work are given below.

The first step is to generate four obligation expressions by prefixing each of the two fact positions F_1, F_2 by (first) the operator O and (second) O$\neg$, where O is read as 'it is obligatory that':

$$OA, O\neg A, O\neg A, O\neg\neg A.$$

Clearly two of these are identical, and the first and the last are logically equivalent, according to properties of the operator O which will be specified below. We are thus left with two expressions. We next form two more obligation expressions from their negations, yielding the set:

$$OA, O\neg A, \neg OA, \neg O\neg A.$$

The second step is to arrange these four expressions as two truth-functional tautologies:

(i) $OA \vee \neg OA$

(ii) $O\neg A \vee \neg O\neg A$.

Exactly one disjunct of each of (i) and (ii) must be true. There are clearly four available combinations:

(0) $OA \wedge O\neg A$

(1) $OA \wedge \neg O\neg A$

(2) $\neg OA \wedge O\neg A$

(3) $\neg OA \wedge \neg O\neg A$

We can eliminate one of these combinations (0) as logically inconsistent, for which we need to say something about the logic of the operator O.

As we mentioned in Part I, there are many different systems of deontic logic; apart from some qualifications that we make later in section 12.7, the main points we wish to make about the theory of normative positions can be adequately expressed using SDL.

According to the logic of SDL (and of some other approaches to deontic logic)

$$\vdash \mathrm{O}p \rightarrow \neg\mathrm{O}\neg p \qquad \text{i.e. } \vdash \neg(\mathrm{O}p \wedge \mathrm{O}\neg p) \qquad \text{i.e. } \vdash \mathrm{O}\neg p \rightarrow \neg\mathrm{O}p$$

where p is any sentence. Conjunction (0) must be eliminated (it does not express a logical possibility), and (1) and (2) contain redundancies. So we get:

(P_1) $\mathrm{O}A$

(P_2) $\mathrm{O}\neg A$

(P_3) $\neg\mathrm{O}A \wedge \neg\mathrm{O}\neg A$

In addition to the operator O, SDL also has the operator P, read 'it is permitted that', and the following interdefinability axiom:

$$\mathrm{P}p =_{Def.} \neg\mathrm{O}\neg p$$

In virtue of this axiom, P_3 may be re-formulated as

(P_3) $\mathrm{P}A \wedge \mathrm{P}\neg A$

$\{\mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3\}$ we call the set of *normative fact positions*.

By construction, the normative fact positions are mutually exclusive, and their disjunction is a tautology: precisely one of $\{\mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3\}$ holds. If the rule/regulation is as specified in (our simplified version of) the Ting example, then the choice is obviously P_2 (assuming, as we have, that a is a patient). In other words, we are saying that, according to the hospital's regulations, the normative status of the state of affairs described by $aHfa$ is designated by $\mathrm{O}\neg aHfa$ or, equivalently, $\neg\mathrm{P}aHfa$.

Consider now the question: does P_2 supply us with enough information about the normative status of (the state of affairs described by) $aHfa$? One reason for maintaining that it does not is this: the formula $\mathrm{O}\neg aHfa$ tells us nothing about *whose action* is supposed to secure the result that $\neg aHfa$. Is the patient a himself obliged to see to it that $\neg aHfa$ is true? Or is the intention essentially to direct the behaviour of other hospital personnel? One might reply by saying that the regulation as formulated (by Ting, or whoever) is not clear on these matters. *Who* is supposed to take responsibility is not specified. To which again one should reply that it is just these questions which need to be answered if a *complete* and *precise* specification of the hospital's regulations is to be obtained. In which case, it would be highly convenient to have a systematic framework in which can be represented not only normative positions of the types P_1, P_2, P_3, but also the *actions* of the agents whose behaviour was intended to be governed by the regulations concerned. One would want also a methodology for constructing an exhaustive analysis of the complete space of logically possible *normative act-positions* of these agents; one would then ask the policy formulator to make his choice.

A framework and a methodology of just these kinds are to be found in the work of Kanger and Lindahl, to which we now turn.

One-agent act-positions

What we did above was to construct a class of normative positions by applying deontic operators to descriptions of possible 'fact positions'. What we do now, by contrast, is first to construct the class of (individual agent) *act-positions*, and then use *that* as a basis for building a new class of normative positions. We then move on to two-agent positions, at which point some further complexities involved in the analysis of Ting's example can be portrayed. Along the way we indicate briefly the differences and relationships between our framework and those of Kanger and Lindahl.

The treatment of act descriptions employed by Kanger and Lindahl may well appear unfamiliar to many computer scientists. It employs a relativised monadic operator E_x,where expressions of the form

$$\mathrm{E}_x p$$

are read 'agent x brings it about that p' or 'agent x sees to it that p is the case'. The development of this approach to the logic of action stems from [Pör70], and more refined versions have been presented in [Kan72, Pör77, Elg92]. It is important to note that this kind of approach to the logic of action abstracts away from considerations of state change and the temporal dimension, focusing essentially on the agent concerned and the states of affairs that he or she brings about.

The formal properties of this action operator have been extensively investigated in the sources mentioned above. There are several variants, but for present purposes we shall be making use of just two properties, which are common to all of them:

$$\vdash \mathrm{E}_x p \rightarrow p$$

$$\text{If} \quad \vdash p \leftrightarrow q \quad \text{then} \quad \vdash \mathrm{E}_x p \leftrightarrow \mathrm{E}_x q$$

The first axiom captures the idea that the action operator is a 'success operator': if x brings it about that p then p is indeed the case. The second axiom is just closure under logical equivalence.

Returning to the two possible fact positions A and $\neg A$, we first generate the set of act-positions for the agent a vis-à-vis these fact positions. Following the procedure we used earlier, the first step is to generate four act expressions by prefixing each of the two fact positions by (first) the operator E_a and (second) $\mathrm{E}_a\neg$, yielding

$$\mathrm{E}_a A,\ \mathrm{E}_a\neg A,\ \mathrm{E}_a\neg A,\ \mathrm{E}_a\neg\neg A$$

Again two of these are identical, and the first and the last are logically equivalent, leaving two act-expressions. We next form two more act expressions from their negations, yielding the set:

$$\mathrm{E}_a A,\ \mathrm{E}_a\neg A,\ \neg\mathrm{E}_a A,\ \neg\mathrm{E}_a\neg A$$

We can form two tautologies:

(iii) $\mathrm{E}_a A \vee \neg\mathrm{E}_a A$

(iv) $\mathrm{E}_a\neg A \vee \neg\mathrm{E}_a\neg A$

which in turn generate four conjunctions:

(4) $\mathrm{E}_a A \wedge \neg\mathrm{E}_a\neg A$

(5) $\mathrm{E}_a\neg A \wedge \neg\mathrm{E}_a A$

(6) $\neg\mathrm{E}_a A \wedge \neg\mathrm{E}_a\neg A$

(7) $\mathrm{E}_a A \wedge \mathrm{E}_a\neg A$

Since we assume that E_a is a success operator, (7) must be removed (it is logically inconsistent), and (4) and (5) may be simplified (logical implications), yielding three one-agent *act-positions*:

(A_1) $\mathrm{E}_a A$

(A_2) $\mathrm{E}_a\neg A$

(A_3) $\neg\mathrm{E}_a A \wedge \neg\mathrm{E}_a\neg A$.

Note again that the act-positions are mutually exclusive, and their disjunction is a tautology. In A_1 the agent a brings it about that A; in A_2 the agent a brings it about that $\neg A$; A_3 expresses that agent a remains 'passive' with respect to A (perhaps the terms 'indifferent' or 'neutral' might also be employed).

We now generate six normative expressions by prefixing each of A_1–A_3 with (first) O, and (second) O$\neg$. By negating each of these six we generate six more expressions, and the set of twelve may be displayed in the usual way as six tautologous disjunctions:

$\mathrm{O}\mathrm{E}_a A \vee \neg\mathrm{O}\mathrm{E}_a A$

$\mathrm{O}\mathrm{E}_a\neg A \vee \neg\mathrm{O}\mathrm{E}_a\neg A$

$\mathrm{O}\neg\mathrm{E}_a A \vee \neg\mathrm{O}\neg\mathrm{E}_a A$

$\mathrm{O}\neg\mathrm{E}_a\neg A \vee \neg\mathrm{O}\neg\mathrm{E}_a\neg A$

$\mathrm{O}(\neg\mathrm{E}_a A \wedge \neg\mathrm{E}_a\neg A) \vee \neg\mathrm{O}(\neg\mathrm{E}_a A \wedge \neg\mathrm{E}_a\neg A)$

$\mathrm{O}\neg(\neg\mathrm{E}_a A \wedge \neg\mathrm{E}_a\neg A) \vee \neg\mathrm{O}\neg(\neg\mathrm{E}_a A \wedge \neg\mathrm{E}_a\neg A)$

From these six, $2^6 = 64$ distinct conjunctions (each of six conjuncts) may be formed. Using the logical properties of the O and E_a operators already specified, and in addition the rule of consequential closure

$$\text{If} \quad \vdash p \to q \quad \text{then} \quad \vdash \mathrm{O}p \to \mathrm{O}q$$

(which is also a characteristic feature of SDL), 57 of these 64 conjunctions are internally inconsistent. The remaining 7 are exactly Lindahl's 'basic types of one-agent legal positions', numbered T_1–T_7 as in [[Lin77], p92, with the typographical error in T_1 corrected], and with logical redundancies eliminated:

(T_1) $\mathrm{P}\mathrm{E}_a A \wedge \mathrm{P}\mathrm{E}_a\neg A \wedge \mathrm{P}(\neg\mathrm{E}_a A \wedge \neg\mathrm{E}_a\neg A)$

(T_2) $\mathrm{P}\mathrm{E}_a A \wedge \mathrm{O}\neg\mathrm{E}_a\neg A \wedge \mathrm{P}(\neg\mathrm{E}_a A \wedge \neg\mathrm{E}_a\neg A)$

(T_3) $\mathrm{P}\mathrm{E}_a A \wedge \mathrm{P}\mathrm{E}_a\neg A \wedge \mathrm{O}(\mathrm{E}_a A \vee \mathrm{E}_a\neg A)$

(T_4) $\mathrm{O}\neg\mathrm{E}_a A \wedge \mathrm{P}\mathrm{E}_a\neg A \wedge \mathrm{P}(\neg\mathrm{E}_a A \wedge \neg\mathrm{E}_a\neg A)$

(T_5) $\mathrm{O}\mathrm{E}_a A$

(T_6) $\mathrm{O}(\neg\mathrm{E}_a A \wedge \neg\mathrm{E}_a\neg A)$

(T_7) $\mathrm{O}\mathrm{E}_a\neg A$

We shall refer to T_1–T_7 as *normative one-agent act-positions*.

At this point we turn our attention back to the Ting example: precisely one of T_1–T_7 holds for patient a vis-à-vis the act described by $E_a aHfa$ (that is, vis-à-vis a's bringing it about that he has access to his own file). Which do the hospital regulations intend?

It is, for instance, by no means clear that T_7 is the appropriate choice: is there an obligation on the patient to *see to it* that he does not have access to his own file? He is surely *permitted* to see to it that $\neg aHfa$ is true, but the dominant *obligation* involved is merely that $\neg E_a aHfa$ shall obtain. Earlier, we identified $O\neg A$ as the appropriate 'normative fact position' in Ting's example. There are *three* normative act-positions—T_4, T_6, T_7—that are logically consistent with $O\neg A$.

The distinction between T_4 and T_7 relates to the comments we made above when motivating the introduction of the action operator: for T_7, unlike T_4, places the patient a under an *obligation* to see to it that he does not have access to his own file; T_4 by contrast, as is indicated by its third conjunct, allows the patient a degree of passivity with respect to the bringing about of A. This conjunct may be formulated equivalently as $\neg O(E_a A \vee E_a \neg A)$: perhaps indeed it is not a's responsibility to regulate his access/non-access to his file. This is an important point: the choice between T_4 and T_7 may well turn on considerations pertaining to the intended attribution of *responsibility.* Note that T_6 on the other hand *obliges* patient a to remain passive, which seems bizarre in the circumstances. We are thus drawn to accepting T_4 as the most likely intended interpretation, given some reasonable assumptions about where responsibility is going to be attributed in this context. Our claim, of course, is not that T_4 *must* be the appropriate choice; nor are we suggesting that we have provided an exhaustive analysis of the concept of responsibility itself. Our point is that these formal tools may be used to identify the class of interpretations possible at this level of complexity of the formalisation, express differences of meaning between them, and hence enable the policy formulator to sharpen—make more precise—the intended specification.

The Kanger theory takes its point of departure in act positions A_1 and A_2 only

(A_1) $E_a A$

(A_2) $E_a \neg A$

generating eight normative expressions which, arranged as four tautologies, are:

$$OE_a A \vee \neg OE_a A$$
$$OE_a \neg A \vee \neg OE_a \neg A$$
$$O\neg E_a A \vee \neg O\neg E_a A$$
$$O\neg E_a \neg A \vee \neg O\neg E_a \neg A$$

From these four, $2^4 = 16$ distinct conjunctions (each of 4 conjuncts) may be formed. The logics for O and E_a dictate that 10 of these are internally inconsistent. The six that remain are (following the numbering at [[Lin77], p100]):

(K_1) $PE_a A \wedge PE_a \neg A$

(K_2) $O\neg E_a A \wedge O\neg E_a \neg A$

(K_3) $OE_a A$

(K_4) $PE_a A \wedge P\neg E_a A \wedge O\neg E_a \neg A$

(K_5) $OE_a\neg A$

(K_6) $O\neg E_a A \wedge PE_a\neg A \wedge P\neg E_a\neg A$

The relationship between T_1–T_7 and K_1–K_6 is this (cf. [[Lin77], pp.100–101]):

K_1	is logically equivalent to	$(T_1 \vee T_3)$
K_2	... ...	T_6
K_3	... ...	T_5
K_4	... ...	T_2
K_5	... ...	T_7
K_6	... ...	T_4

Comparing K_1 with T_1 and T_3, we see that K_1 permits the agent a to bring it about that A, and permits him to bring it about that $\neg A$, but it remains silent on the question whether a is nevertheless obliged to perform one of these actions. T_1 and T_3, on the other hand, supply the missing information. If a is obliged to perform one of the two actions, then T_3 applies; if, however, he is permitted to remain passive with respect to the bringing about of A, then T_1 applies.

The explanation of the difference between the Kanger set and the Lindahl set is that Kanger built the normative one-agent act positions on an incomplete characterisation of the underlying act-positions, failing to single out the 'passive', 'indifferent' or 'neutral' position expressed by A_3. We have been suggesting that this omission may deny the Kanger set the opportunity to provide an explication of some central intuitions pertaining to agent-responsibility.

We note in passing that Lindahl generates the same set T_1–T_7 but using a method which differs from ours. His method is simpler in the sense that it requires fewer steps; it generates the same set of normative one-agent act-positions in this case but it relies on certain properties of SDL—the respective roles of truth-functional logic on the one hand and modal logical principles on the other are not so cleanly separated as they are in our procedure. We have been very much concerned with explicitness: we want a method that can be guaranteed to generate the complete set of 2^n conjunctions of normative expressions (where n is the number of tautologies) without trading on any assumptions about the properties of the modal logics employed. In certain cases, short-cuts (such as Lindahl's) can be devised, but these optimisations depend on the specific choice of modal logics. We shall explore these issues in a future paper devoted to a technical presentation of the methodology.

Two-agent act positions

The next move is to the two-agent situation.

Rename T_1 as $T_1[a; A]$ in order to make explicit the agent a, that is, the subscript of the action operator, and the embedded fact position A; rename T_2–T_7 in the same fashion.

Now consider the set $T_1[b; A]$– $T_7[b; A]$ (so that, e.g., the unabbreviated version of $T_7[b; A]$ will read $OE_b\neg aHfa$). And let us suppose, for purposes of further illustration, that b is a doctor in the hospital where a is one of the patients. To investigate the class of normative two-agent act-positions (for agents a and b vis-à-vis the state of affairs

described by $aHfa$) we need to consider the 49 possible conjunctions obtainable by selecting one conjunct from

$$\{\mathrm{T}_1[a;A],\ldots,\mathrm{T}_7[a;A]\}$$

and the other conjunct from

$$\{\mathrm{T}_1[b;A],\ldots,\mathrm{T}_7[b;A]\}.$$

As established in [[Lin77], p128] just 14 of the 49 conjunctions are internally inconsistent. The remaining 35 cases can be examined in the light of our attempt to analyse Ting's hospital regulation forbidding patient a access to his own file: i.e. we need to examine those of the 35 which contain $\mathrm{T}_4[a;A]$.

By inspection it is readily seen that $\mathrm{T}_4[a;A]$ may be consistently conjoined with any one of $\mathrm{T}_1[b;A]$– $\mathrm{T}_7[b;A]$, except $\mathrm{T}_5[b;A]$. (The reason is simply that $\mathrm{T}_5[b;A]$ is OE_bA which is of course inconsistent with the second conjunct of $\mathrm{T}_4[a;A]$, and indeed with $\mathrm{O}\neg A$ itself.) The intended interpretation of the hospital regulation (of its import vis-à-vis the doctor b's behaviour) rules out $\mathrm{T}_1[b;A]$, $\mathrm{T}_2[b;A]$ and $\mathrm{T}_3[b;A]$ immediately, since each contains PE_bA, which says that b is permitted to see to it that a has access to a's file. Consideration of the appropriateness of $\mathrm{T}_4[b;A]$ raises again the question of responsibility: if it is deemed to be the doctor's job (part of his role) to make sure that a does not have access to a's file, then $\mathrm{T}_4[b;A]$ has to be rejected, because the third conjunct $\mathrm{P}(\neg\mathrm{E}_bA\wedge\neg\mathrm{E}_b\neg A)$ allows b to remain 'passive' in regard to the state of affairs described by A. Such considerations would also of course rule out $\mathrm{T}_6[b;A]$, which makes b's passivity (vis-à-vis A) obligatory.

Following this line of interpretation, we would then be left with $\mathrm{T}_7[b;A]$, according to which b is obliged to see to it that a does not have access to a's file: responsibility for securing compliance with the prohibition expressed by the hospital regulation lies on b's shoulders.

The combined, two-agent position $\mathrm{T}_4[a;A]\wedge\mathrm{T}_7[b;A]$ (which is Lindahl's R_{25} [[Lin77], p130]) may be reduced to three conjuncts, because $\mathrm{OE}_b\neg A$ entails $\mathrm{O}\neg\mathrm{E}_aA$:

$$\mathrm{T}_4[a;A]\wedge\mathrm{T}_7[b;A] \quad = \quad \mathrm{OE}_b\neg A\wedge\mathrm{PE}_a\neg A\wedge\mathrm{P}(\neg\mathrm{E}_aA\wedge\neg\mathrm{E}_a\neg A).$$

It is instructive to compare the above with other cases, where some agent c is assumed to fill a role other than that of doctor. For instance, if c is a fellow patient, then perhaps passivity *is* required from him (vis-à-vis A), so perhaps the appropriate choice is $\mathrm{T}_4[a;A]\wedge\mathrm{T}_6[c;A]$. The theory of normative two-agent act-positions does not (of course) tell us what the correct interpretation is: it simply maps out in an exhaustive and precise fashion what the available interpretations are, at the given level of complexity.

Lindahl [[Lin77], Ch. 5] extends his theory to what he calls 'collectivistic two-agent types', to cover the not uncommon situation where, for instance, there is an obligation on two agents which does not necessarily apply to either of them individually:

$$\mathrm{O}(\mathrm{E}_ap\vee\mathrm{E}_bp)\wedge\neg\mathrm{OE}_ap\wedge\neg\mathrm{OE}_bp.$$

Lindahl is there concerned with those regulations which require or permit agents to

co-ordinate their activity. We shall not pursue this matter further here, short of saying that the procedure we have outlined above may be applied to generate the class of normative *collectivistic* two-agent act-positions.

One ambiguity in the term 'right'

We began this section by simplifying Ting's example, replacing reference to 'rights' by reference to permission. We indicated that this was a non-trivial alteration, and we are now in a position to explain this point.

For instance, suppose that there is also a regulation granting the patient the *right* to change his physician ([[Tin90], p5] has a slightly more complicated variant of this). Then this might well be taken to mean, not only that the patient is *permitted* to change his physician, but moreover that no agent is permitted to *prevent* him from changing his physician if this is what he chooses to do. Returning to our original example from Ting, it is conceivable that the patient is *permitted* to access his own file, yet he does not have that *right*, because there are other agents in the hospital who are permitted to see to it that he does not have access to his own file.

The formal apparatus introduced so far is capable of articulating nuances of this kind: for instance, the conjunction

$$\mathrm{PE}_a A \wedge \mathrm{PE}_b \neg A$$

is perfectly consistent according to the logical principles we have chosen.

It can be argued that there are also many other ambiguities associated with the term 'right' which the formal apparatus developed so far cannot accommodate. Some of these complexities may, however, be captured by further refinements of the logical machinery, of the kind to which we now turn.

12.7 FURTHER REFINEMENTS

We have above indicated how the logical space of normative positions may be constructed from two bases: the class of 'fact positions' and the class of 'act positions', respectively. In the latter case we also showed how to develop the theory to cover the two-agent situation. Its further development to cover n-agent situations ($n > 2$) is a routine matter. The size of n will of course depend on the content of the regulations to be formalised, but it is nevertheless reasonable to suppose that, in many organisations (hospitals included), regulations will be directed at *categories* of individuals (individuals occupying certain work-roles, or having a particular level of security clearance, for instance), and that many regulations will concern only a small number of categories (for instance some of Ting's examples concern just three categories: patient, patient's physician, consulting physician). However, it is fair to say that the task of manually checking through the list of conjunctions (the potential normative positions), to identify logical inconsistencies and to eliminate redundancies in the remaining consistent cases, rapidly becomes very tedious for $n > 2$. Automated procedures are clearly called for here (and elsewhere in the developments of these tools)—but such procedures can be readily supplied (see section 12.8).

The size of n is not the only factor which can be varied, of course. For certain purposes we may need also to consider changing the initial characterisation of the class of act-positions from which the normative positions are constructed. Some regulations pertain not to *individual* agent positions of the form $\mathrm{E}_a p$, but to what we may call interpersonal *control* positions, e.g., of type $\mathrm{E}_b \mathrm{E}_a p$ or $\mathrm{E}_b \neg \mathrm{E}_a p$ (cf. [[Pör77], Ch.3]). For instance [[Tin90], p6] says that it is the responsibility of physicians or psychiatrists to protect the confidentiality of their patients' records. Now it might be intended that, at least in part, a regulation of this kind obliges the physician/psychiatrist to see to it that it is not the case that a particular class of actions is performed by a particular class of agents. That is to say, the physician/psychiatrist is obliged to exercise control of the preventive kind vis-à-vis certain actions of certain others. We shall not develop this possibility in detail: the essential point is that the methods and tools we have already identified can also be applied to the investigation of normative positions of these sorts. The construction of the logical space of normative control positions (for two agents, a, b, where a is in the role of 'controller') begins with the set of six control-action expressions (vis-à-vis state of affairs p):

$$\{\mathrm{E}_a \mathrm{E}_b p,\ \mathrm{E}_a \neg \mathrm{E}_b p,\ \mathrm{E}_a \mathrm{E}_b \neg p,\ \mathrm{E}_a \neg \mathrm{E}_b \neg p,\ \mathrm{E}_a (\neg \mathrm{E}_b p \wedge \neg \mathrm{E}_b \neg p),\ \mathrm{E}_a \neg (\neg \mathrm{E}_b p \wedge \neg \mathrm{E}_b \neg p)\}$$

and their respective negations. The class of conjunctions to be investigated here will of course be large, and we see again the need for automated techniques; but the principles for generating this class are clear.

The reader familiar with deontic logic will no doubt remark that the logical principles which determine inconsistencies and redundancies in the generated sets of conjunctions include the principle of closure of the O and P operators under logical implication ('consequential closure'). We are fully aware that a case may be made for maintaining that it is precisely the consequential closure principle which creates some of the notorious 'paradoxes' of SDL. (See, e.g., [JP85] for a discussion.) However, the specific instances of the consequential closure principle which we have here employed would seem to be innocuous. For example, if it is permitted that an agent brings it about that p, then it must also be permitted that p; even if the closure principle in its full generality were to be rejected, a good case could be made for retaining those instances of it which we have here used. Furthermore, as we indicated above in our comments on Lindahl's method, our procedure for generating normative positions effects a clean separation between the truth-functional and non-truth-functional (modal-logical) principles involved. Using our procedure, one can readily replace SDL by some other deontic logic which does not validate the consequential closure principle in its full generality, and investigate the class of normative positions thereby generated.

Returning to further refinements in the logic, one very obvious need is for the introduction of quantifiers. It will often be necessary to specify that every individual—or at least every individual in a certain category—occupies this or this normative position. The example from which we started, for instance, is clearly a prohibition which applies to everyone in the category of patient. While the combination of quantification and modality generates a number of controversial philosophical issues, formally adequate treatments of quantified modal logic are nevertheless available.

A more challenging logical problem concerns the equally clear need for the introduction of *conditional* structures in the specification of normative positions.

> "A nurse under the supervision of a physician may access the patient's record..." [[Tin90], p.7]

clearly attaches a normative position of type permission to a condition: being under supervision of a physician. Furthermore, it is often very natural to build conditional structures into universally quantified forms; our first example might be said to be of the form:

> For all x, if x is a patient then it is not permitted that x has access to the file of x.

As already observed in Part I, the formal treatment of deontic conditionals has presented some difficult logical problems. The key question to raise in the present context is whether problems about the analysis of deontic conditionals should be allowed to have a bearing on the methods used for determining the logical space of normative positions. Our inclination is to keep these two issues apart. In other words, we propose to retain the procedures for generating the class of normative positions which (following the Kanger-Lindahl tradition) we have outlined above. The consistent conjunctions generated by these methods may then be employed as *consequents* in deontic conditional structures. In essence we are suggesting that the general form of a deontic conditional is:

$$\text{If} \quad [\textit{condition}] \quad \text{then} \quad [\textit{normative position}]$$

and that the 'If ...then ...' must be interpreted as a default conditional (see, e.g., [JP91]). Thus, if T_4 (above, section 12.6) is selected as the appropriate normative position in the interpretation of our initial example from Ting, then we assign the form:

$$\text{If} \quad [a \text{ is a patient}] \quad \text{then} \quad \mathrm{O}\neg \mathrm{E}_a aHfa \wedge \mathrm{P}\mathrm{E}_a \neg aHfa \wedge \mathrm{P}(\neg \mathrm{E}_a aHfa \wedge \neg \mathrm{E}_a \neg aHfa)$$

There are clearly methodological advantages to be gained from keeping separate the problem of conditionals, on the one hand, and the definition of position-generating procedures, on the other.

Sometimes regulations require or permit not merely that some action or other is performed, but that it is performed in a certain way or ways. In other words, the regulation specifies both the *end* which is required or permitted and the *means* to that end. For example [[Tin90], p.6]:

> "The physician may use administrative staff and other medical professionals to maintain medical records ..."

expresses the idea that the physician is allowed to employ a particular means (the assistance of administrative staff and other medical professionals) to achieve a particular end (maintaining medical records). (Elsewhere the regulations also say that the physician is required to achieve this end.)

There are grounds for supposing that the action logic employed above cannot adequately capture *dyadic* act descriptions of this kind, i.e. of the type 'a brings it about

that B by means of C'. Nevertheless, proposals exist in the literature for constructing a dyadic action logic (within a framework in which monadic act descriptions, of the type considered above, may also be expressed). (See, e.g., [Elg92].) And there is no doubt that one could reconstruct the theory of normative positions on the basis of a dyadic action logic. For instance, for the one-agent case and a given pair B, C of 'end' and 'means', one would first define the class of all logically possible *dyadic act positions*, and then proceed, from that base, to the construction of normative positions in the ways indicated earlier.

It might also be argued that our stock of action-logical tools needs to be expanded to give place to a notion of 'practical possibility', so that it is possible to represent, and to reason about, what an agent is *actually able to do*. The 'may/can' of permission, and the 'may/can' of practical possibility are clearly distinct concepts, and in reasoning about the regulation of action it is often important to maintain this distinction explicitly. For instance, when it is said [[Tin90], p.5] that

> "A patient has the right to request that a physician transfer his or her medical record to another physician and have that physician become the primary physician."

this might be taken to require—of the doctors and hospital administrators—not merely that they *permit* the patient to get his record transferred (etc.), but that they also make it *possible* for the patient to do just that. (Cf. the discussion in [Kan85] of what he called the 'realisation' of rights.) The issue here may be said to concern another kind of inter-agent *control*, or *influence*. Whereas above we considered interaction structures of the type $\mathrm{E}_a\mathrm{E}_b p$, for instance, expressing the idea that a brings it about that b brings it about that p, we are now focusing on control/influence relations of a weaker type, of the form $\mathrm{E}_a\mathrm{Can}\mathrm{E}_b p$, which expresses the idea that a makes it possible for b to bring it about that p. It is clear that structures of this sort may also provide a basis from which to construct a class of normative positions. For instance, as we have indicated, if a has an obligation to *realise* b's right to do p, then $\mathrm{O}\mathrm{E}_a\mathrm{Can}\mathrm{E}_b p$ is true. The distinction between permission and practical possibility takes on added significance in the context of computer systems, as we shall indicate in section 12.9 below.

So far as we can see, there is no reason to suppose that this line of development would generate any difficulties regarding the methods to be employed in constructing the logical space of normative positions (for, e.g., a given pair of agents and a given type of state of affairs); furthermore proposals exist in the literature [Kan72, Pör77, Elg92] for the logic of the modality Can—and of its dual, which expresses a notion of 'practical unavoidability'.

The Ting paper also provides examples of another common feature of sets of regulations: *authorisation*. The point is that rules designed to guide interaction within an organisation frequently specify who is *empowered* to grant permissions, or impose obligations—who is entitled to make alterations to the normative positions obtaining/in force at any given time.

For instance [[Tin90], p.5]:

> (X) "An adult patient must grant permission for his or her spouse, close relatives or legal guardians to ... access certain medical information."
>
> (Y) "...[the patient] must maintain the rights to give permission to have other physicians or medical professionals access the record."

These are relatively complicated structures; there is nevertheless reason to believe that the logical tools presented so far can capture the core of authorisation rules. Suppose, for instance, that a is an adult patient and b his/her spouse, and let the 'certain medical information' be a's file. Then part, at least, of what (X) is saying is the following:

(X′) $\quad \neg(\mathrm{PE}_b bHfa \wedge \neg \mathrm{E}_a \mathrm{PE}_b bHfa)$

for (X) states that b is permitted this access only if the permission has been created by a. Presumably (X) must also be taken to contain:

(X″) $\quad \mathrm{PE}_a \mathrm{PE}_b bHfa.$

And when, in (Y), we are told that the patient 'must *maintain* the rights to give permission' this presumably means that nobody *else*, other than the patient, has that right; only the patient can grant this permission.

So we are concerned here with acts of creating/changing normative positions, and with the assignment of normative position *to such acts*. The suggestion is, then, that structures of these kinds are analysable, within the existing framework, by means of appropriate iteration of deontic and action operators. Furthermore, the methodological procedures outlined above should also be applicable to the construction of classes of normative positions of this 'meta-level' kind. (See [[Lin77], part II] for an attempt to develop some aspects of this theory.)

12.8 APPLICATIONS AND SOFTWARE TOOLS

Perhaps the best way to illustrate how we envisage these tools being used is by turning again to our original example from Ting. We are told that according to the regulations of the hospital, or according to the regulations that the hospital authorities have in mind, "a patient does not have the right to access his or her own medical record".

We would begin the analysis by establishing first which of the three normative fact positions here applies. So, the first question to consider is whether the state of affairs in which a patient has access to his own files is or is not a permitted state of affairs. Since the regulation is expressed in terms of the complex notion of 'right', the answer to this question is not immediately obvious. Suppose the answer is 'no'. Then we have uniquely identified one of the three normative fact positions as the appropriate choice, and we would turn now to consider the normative one-agent act positions which are consistent with this initial choice. We have already shown in the earlier section that there are just three of these, differing we have suggested, with respect to attribution of responsibility. We might consider in turn various categories of agent: the patient himself, fellow patients, the patient's own doctor, other doctors in the hospital, and so on. Once the choices have been made at this level of detail, we might turn to an analysis of the interpersonal control positions consistent with these choices, or to authorisation questions, or to any of the other dimensions we have mentioned for finer grained analysis of the normative positions involved. We are not saying that all of these dimensions will always have to be explored; how detailed the analysis is going to be is itself a matter of choice.

If on the other hand, the answer to the first question had been 'yes' then two normative fact positions become candidates. One could discriminate between them by then asking whether the state of affairs concerned was or was not obligatory. The answer to *this* question will again lead to unique identification of the appropriate normative fact positions, from which point we can proceed to the more detailed analysis, as before.

It is perfectly conceivable that the hospital administrator, or whoever is formulating the policy, may feel unable to make a categorical decision about which interpretation to select at any stage. The purpose of the tool is not to force a decision in these circumstances but to show the available options, and to indicate very precisely the consequent normative positions which will be determined by any particular choice. The idea is that the tools may guide the policy formulator by bringing to his attention distinctions and considerations that might otherwise have remained undetected, until such time as he feels that no further distinctions can usefully be made.

In order to turn this analytical procedure into a practical tool, automated support is required. We have such software tools under development, which we describe in terms of three separate levels.

At the first level, there is an implementation of the basic method for constructing classes of normative positions: normative fact positions, normative one-agent act positions, and so on. As we have indicated, generating first the complete class of conjunctions, and then from that, the class of consistent conjunctions with redundancies removed, is a mechanical task—automation is required because the number and the size of the expressions to be manipulated grows very rapidly, and the task cannot be undertaken manually for anything beyond the most elementary positions. It is not necessary to build this implementation on top of full-blown theorem provers, since only fragments of the underlying modal logics are involved; the propositional logic component is obviously not a problem. An implementation (in Prolog) at this level exists; we are currently extending it to provide a range of other routines that are useful for manipulating the symbolic expressions that occur.

The software tools at level one are intended primarily for our own use, in exploring the logical properties of this framework. The software tools at level two are designed to provide more direct support for the kind of application we outlined above. They are constructed by including an additional layer of software on top of the level-one routines. This additional layer provides two main features. The first is just an interface which allows the user to keep track of choices that have been made so far in the interpretative process, and provides a range of commands that wrap up the most common combinations of uses of the level-one routines. The second feature is an additional facility that is often helpful in choosing between possible interpretations. As already illustrated, it is possible to discriminate between the three normative fact positions by asking (at most) two questions—is the state of affairs permitted? And then, is it obligatory? Whatever the level of complexity, it is always possible to generate a series of questions in order to discriminate between several candidate positions. In the worst case, this can be done by going through each of the positions and asking about each of the conjuncts in turn; it is also easy to devise simple heuristics which will reduce the number of questions, or generate questions in some more 'natural' order. We omit the details. It is useful to distinguish between these two levels of software, because it is only the first (level-one) that involves automation of the logical principles; the additional layer for level-two just makes the level-one routines more convenient to use

during the interpretative process. Whilst we shall no doubt want to refine and extend this second layer, its implementation involves no new technical-logical questions.

In the longer term, we plan to provide a third level of support, by adding a layer—a 'front end'—to the level-two system. This third layer is intended to make the system more accessible to non-specialists, and to provide strategic advice on how to use the level-two software in practice. At this stage we can say very little about it, except to indicate in general terms what we have in mind. We suppose that it will be possible to identify some methodological pointers for making best use of these formal tools. For instance, our advice would be to complete an analysis of the normative *fact* positions before moving to one-agent act-positions. But whether one should consider control positions next, or questions of authorisation, or other dimensions, we cannot say until we have gained more experience with the practical uses of these tools. We have no doubt that some kind of level-three support can be provided, but this observation is based only on experience with front-ends to other kinds of software packages. We have mentioned level-three support as a way of indicating how we see the development of the software tools proceeding in the longer term.

12.9 FURTHER REMARKS ON REGIMENTATION

Much of what Ting says applies quite independently of whether the sensitive (medical) information is stored in a computer system or on paper; more to the point, many of his example regulations do not concern just the computer system itself, but cover more general organisational aspects of the hospital as a whole. Thus, the requirements that a patient maintains the right to permit access to his or her medical record by certain others, or that a patient has the right to change his or her physician, or that physicians have a duty to access their patients' records on a regular basis, or that they also have responsibility for ensuring the accuracy of these records, all make important contributions to security and confidentiality, but they are not regulations that would necessarily be incorporated into the computer system directly. The main theme of Ting's discussion paper is that computer-technical issues concerning the access control mechanisms provided as part of the computer system cannot be divorced from these more general organisational concerns—

> "... the information security requirement must be considered as an integral part of the application" [Tin90].

However, it is also the case that Ting is explicitly concerned with how these general security-related policies and regulations impact on the design and implementation of the computer system's access control mechanisms. This question takes us back to issues raised in Part I, where we pointed out that one way of proceeding is to design the system in such a way that conformity to regulations is *forced* by the implementation: a process we called regimentation. As we also put it, regimentation forces ideality and actuality to coincide. In the context of the library regulations, the examples we focused on concerned the regimentation of obligations and prohibitions; the system was to be designed in such a way that these would not in fact be *violated*. In the context of the hospital regulations, however, many of the examples focus on permissions and rights;

regimentation here will be a matter of ensuring that these are always in fact *realisable*. To illustrate this we provide two more examples from [[Tin90], p7]:

> "A nurse ... [in certain circumstances] ... can only access the portion of the medical record and data which is relevant to the nurse's duty and function."

> "A medical technician ... [under certain circumstances] ... may access certain portions of the medical record."

Viewed as part of an (informal) specification of the computer system's mechanisms for controlling access to patients' records, such examples can be cast into the following form:

> a user of category C may/is permitted/has the right to access records of type R if and only if conditions X are satisfied.

What regimentation amounts to in such cases is that the 'may/can' of permission gets transformed to the 'may/can' of *practical possibility*: the system will be designed in such a way that, among the repertoire of actions available to users of category C in circumstances X, there is provided an action which has the result of giving the user access to records of type R.

In the light of these remarks, we need to refine our earlier comments in Part I concerning the regimentation of obligations and prohibitions. If regimentation transforms permission to practical *possibility*, then it is natural to say that obligation is correspondingly transformed to practical *necessity*. When an obligation is regimented by forcing the actual behaviour of the system to conform to that obligation, this amounts to designing the system in such a way that *no* configuration of state changes in the system will lead to violation of the obligation: in other words, it becomes practically necessary, or unavoidable, that the system behaves in the ideal way; this notion of practical necessity is the dual of the notion of practical possibility.

Just as with obligations in the library regulations of Part I, so also here, it is not reasonable to suppose that all permissions could be regimented.

Consider another example from Ting:

> "A patient has the right to transfer his medical record to another physician and have that physician become the primary physician."

It is surely not realistic to imagine that patients in a mental hospital would be given use of a computer system that provides some mechanism by which the patient can transfer his records to another physician and make that physician his primary physician. There are no technical obstacles to providing such a mechanism, but it just does not seem likely that the hospital would want it.

In discussing the limits of regimentation in Part I, we mentioned the fact that faults may still arise, either because of defects in the operation of the system itself, or because of the influence of extraneous factors. This point of course applies not only in regard to the regimentation of obligations but also to the regimentation of permissions. Fault tolerance, we have suggested, pertains to the distinction between actuality and ideality; we are now in a position to refine this distinction. Consider the task of specifying requirements for a fault-tolerant computer system which is itself intended either to regiment, to some degree, a set of existing regulations, or to

operate as one of a collection of agents whose interactions are to be governed by a set of regulations. The discussion we have just gone through points in the direction of a three-fold distinction which this kind of requirements specification will have to take into account. There is the question of how the agents ought and are permitted to behave if the regulations are followed; there is the question of how they must or can behave, as a matter of practical necessity and possibility, if the system operates in a fault-free fashion; and there is the question of how the system in fact behaves.

Each of these three dimensions has a role to play in the specification of computer systems. The detailed characterisation of these dimensions, and of their interplay in systems specification, is a task of considerable complexity which we plan to address in future work.

Our final remark concerns the approach to the logic of action we have employed in this paper. As already observed, the 'brings it about' operator abstracts away details of the specific actions performed by the agent, changes of state, and the temporal dimension generally; we have indicated that for certain purposes this abstraction is appropriate. But in the context of computer systems, a specification employing this operator would be a formal specification at an unusually high level of abstraction. Now, it would be interesting to compare the practical value of a specification at this level of abstraction with the more detailed specifications that are constructed using more standard formal specification languages in computer science. This is an exercise that we are planning to undertake. Whatever the outcome, it is clear that some aspects of access control mechanisms and some of the behaviour of distributed computer systems need to be modelled at a finer grain of detail. In these cases, it will be necessary to replace or augment use of the 'brings it about' operator with more standard approaches to action and time in computer science (dynamic logic, say, or one of the knowledge representation formalisms developed in artificial intelligence). Some work along these lines has already been conducted (see, e.g. [Elg92]); further development is again part of our current plans.

ACKNOWLEDGEMENTS

This work was begun while Andrew Jones was supported by a Visiting Fellowship at Imperial College for 1990/91 from the United Kingdom's Science and Engineering Research Council, and completed while he was supported by a Visiting Fellowship from Région Rhône-Alpes at LIFIA-INPG, Grenoble, France for 1991/92. We are grateful to Philip Morris for valuable discussions concerning the applicability of deontic logic to aspects of computer security, and for identifying Ting's discussion paper as a source of examples. We should also like to thank Kluwer Academic Publishers and Springer-Verlag for granting permission to reprint extracts from our earlier papers.

REFERENCES

[AS86] L.E. Allen and C.S. Saxon. Analysis of the logical structure of legal rules by a modernized and formalized version of Hohfeld fundamental legal conceptions. In A.A Martino and F. Socci, editors, *Automated Analysis of Legal Texts*, pages 385–451. North-Holland, Amsterdam, 1986.

[Ben89] T.J.M. Bench-Capon. Deep models, normative reasoning and legal expert systems. In *Proc. Second International Conference on Artificial Intelligence and Law*, Vancouver, 1987, pages 37–45. ACM Press, 1989.

[BRRS87] T.J.M. Bench-Capon, G.O. Robinson, T.W. Routen, M.J. Sergot. Logic programming for large scale applications in law: A formalisation of Supplementary Benefit legislation. In *Proc. First International Conference on Artificial Intelligence and Law*, Boston, 1987, pages 190–98. ACM Press 1987.

[BMT87] C. Biagioli, P. Mariani, P. and D. Tiscornia. ESPLEX: A rule and conceptual based model for representing statutes. In *Proc. First International Conference on Artificial Intelligence and Law*, Boston, 1987, pages 240–51. ACM Press 1987.

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

[Chi63] R.M. Chisholm. Contrary-to-duty imperatives and deontic logic. *Analysis*, 24, 1963.

[Coe91] J. Coenen. Specifying Fault Tolerant Programs in Deontic Logic. In J.-J. Ch. Meyer and R.J. Wieringa, editors, *Proc. First International Workshop on Deontic Logic in Computer Science*, Amsterdam, December 1991.

[Elg92] D. Elgesem. *Action Theory and Modal Logic*. Doctoral Thesis, University of Oslo, Department of Philosophy, 1992.

[FM91] J. Fiadeiro and T.S.E. Maibaum. Temporal reasoning over deontic specifications. *Logic and Computation.* (to appear).

[Hil81] R. Hilpinen, editor. *New Studies in Deontic Logic*. Synthese Library 152, D.Reidel, Dordrecht, 1981.

[Jon90] A.J.I. Jones. Deontic logic and legal knowledge representation. *Ratio Juris*, 3 (2):237–44, July 1990.

[JP85] A.J.I. Jones and I. Pörn. Ideality, sub-ideality and deontic logic. *Synthese*, 65, 1985.

[JP91] A.J.I. Jones and I. Pörn. On the logic of deontic conditionals. In J.-J. Ch. Meyer and R.J. Wieringa, editors, *Proc. First International Workshop on Deontic Logic in Computer Science*, Amsterdam, December 1991.

[JS92a] A.J.I. Jones and M.J. Sergot. Deontic logic in the representation of law: Towards a methodology. *Artificial Intelligence and Law*1:45–64, 1992.

[JS92b] A.J.I. Jones and M.J. Sergot. Formal Specification of Security Requirements using the Theory of Normative Positions. In Y. Deswarte, G. Eizenberg and J.-J. Quisquater, editors, *Computer Security—ESORICS 92*, Lecture Notes in Computer Science 648, pages 103–121. Springer-Verlag, Berlin Heidelberg, 1992.

[Kan72] S. Kanger. Law and Logic. *Theoria*, 38, 1972.

[Kan85] S. Kanger. On Realization of Human Rights. In G. Holmström and A.J.I. Jones, editors, *Action, Logic and Social Theory*. Acta Philosophica Fennica, Vol. 38. 1985.

[KK66] S. Kanger and H. Kanger. Rights and Parliamentarism. *Theoria*, 32, 1966.

[KM87] S. Khosla and T.S.E. Maibaum. The prescription and description of state based systems. In B. Banieqbal et al., editors, *Temporal Logic in Specification*, Lecture Notes in Computer Science 398, Springer-Verlag, Berlin, 1987.

[Lin77] L. Lindahl. *Position and Change – A Study in Law and Logic*. Synthese Library 112, D.Reidel, Dordrecht, 1977.

[LM83] B. Loewer and M. Belzer. Dyadic deontic detachment. *Synthese*, 54:295–319, 1983.

[Mac90] D.N. MacCormick. Paper presented to the ESPRIT Working Group on Foundations of Legal Reasoning, Cork, 1990.

[McC86] L.T. McCarty. Permissions and obligations. In *Proc. Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, 1983, pages 287–94.

[McC83] L.T. McCarty. Permissions and obligations: An informal introduction. In A.A Martino and F. Socci, editors, *Automated Analysis of Legal Texts*, pages 307–37. North-Holland, Amsterdam, 1986.

[Mey88] J.-J. Ch. Meyer. A different approach to deontic logic: Deontic logic viewed as a

variant of dynamic logic. *Notre Dame Journal of Formal Logic*, 29, 1988.

[ML85] N.H. Minsky and A. Lockman. Ensuring integrity by adding obligations to privileges. In *Proc. Eighth International Conference on Software Engineering*, August 1985, pages 92–102.

[MM91] P. Morris and J. McDermid. Security and Normative Rights. In J.-J. Ch. Meyer and R.J. Wieringa, editors, *Proc. First International Workshop on Deontic Logic in Computer Science*, Amsterdam, December 1991.

[Pör70] I. Pörn. *The Logic of Power.* Blackwells, Oxford, 1970.

[Pör77] I. Pörn. *Action Theory and Social Science: Some Formal Models.* Synthese Library 120, D. Reidel, Dordrecht, 1977.

[Ser82] M.J. Sergot. Prospects for representing the law as logic programs. In K.L. Clark and S-Å. Tarnlund, editors, *Logic Programming.* Academic Press, London, 1982.

[Ser88] M.J. Sergot. Representing legislation as logic programs. In J.E. Hayes, D. Michie and J. Richards, editors, *Machine Intelligence 11*, pages 209–60. Oxford University Press, 1988.

[Ser90] M.J. Sergot. The representation of law in computer programs: A survey and comparison. In T.J.M. Bench-Capon, editor, *Knowledge Based Systems and Legal Applications.* Academic Press, 1990.

[SSK+86] M.J. Sergot, F. Sadri, R.A. Kowalski, F. Kriwaczek, P. Hammond and H.T. Cory. The British Nationality Act as a logic program. *Communications of the ACM*, 29(5):370–86, May 1986.

[Sus87] R.E. Susskind. *Expert Systems in Law: A Jurisprudential Inquiry.* Oxford University Press, 1987.

[Tin90] T.C. Ting. Application Information Security Semantics: A Case of Mental Health Delivery. In D.L. Spooner and C.E. Landwehr, editors, *Database Security: Status and Prospects III.* North Holland, Amsterdam, 1990.

[TM82] W.L. Twining and D. Miers. *How To Do Things With Rules.* Wiedenfeld and Nicolson, London, second edition, 1982.

[WMW89] R.J. Wieringa, J.-J. Ch. Meyer and H. Weigand. Specifying dynamic and deontic integrity constraints. *Data & Knowledge Engineering*, 4, 1989.

[ÅH81] L. Åqvist and J. Hoepelman. Some theorems about a "tree" system of deontic tense logic. In R. Hilpinen, editor, *New Studies in Deontic Logic.* Synthese Library 152, D.Reidel, Dordrecht, 1981.