Logic-Based Tools for the Analysis and Representation of Legal Contracts

Aspassia-Kaliopi Daskalopulu

Submitted in fulfilment of the requirements for the award of the degree of Doctor of Philosophy of the University of London

Imperial College London

1999

ABSTRACT

There has been a significant amount of research in Artificial Intelligence focusing on the representation of legislation and regulations. The motivation for this has been twofold: on the one hand there have been opportunities for developing advisory systems for legal practitioners; on the other hand the law is a complex domain in which diverse modes of reasoning are employed, offering ample opportunity to test existing Artificial Intelligence techniques as well as to develop new ones. The general aim of the thesis is to explore the potential for developing logic-based tools for the analysis and representation of legal contracts, by considering the following two questions:

(a) To what extent can techniques developed for the representation of legislation and regulations be transferred and applied usefully in the domain of legal contracts?

(b) What features are specific to legal contracts and what techniques can be developed to address them?

The intended applications include both the drafting of new contracts and the management and administration of existing ones, that is to say, the general problem of storing and retrieving information from large contractual documents, and more specific tasks such as monitoring compliance or establishing parties' duties/rights under a given agreement when it is in force. Experimental material is drawn mostly from engineering contracts, which are typically large and complex and contain a multitude of interrelated provisions.

The term 'contract' is commonly used to refer both to a legally binding *agreement* between (usually) two parties and to the *document*, that records such an agreement. The first part of the thesis is concerned with *documents* and the representation of contracts at the *macro-level*: the emphasis is on issues relevant to the design of structurally coherent documents. The thesis presents a document assembly tool designed to be applicable, where contract drafting is based on model-form contracts or existing examples of a given type. The main features of the approach are: (i) the representation addresses the structure and interrelationships between the constituent parts of contracts but not the text of the document itself; (ii) the representation of documents is separated from the mechanisms that manipulate it; and (iii) the drafting process is subject to a collection of explicitly represented constraints that govern the structure of documents.

The second part of the thesis deals with the contents of *agreements* and representations at the *micro-level*. Micro-level drafting is the source of a host of issues ranging from representing the detailed wording of individual sections, to representing the nature of provisions (obligations, powers, reparations, procedures), to representing their "fitness" or effectiveness in securing some party's best interests. Various techniques are available to assist in aspects of this task, such as disambiguating contractual provisions and in detecting inconsistency or incompleteness. The second part of the thesis comprises three discussions. The first is on contractual obligations as the result of promissory exchanges between parties and draws upon work by Kimbrough and his associates. The second concentrates on contractual obligations and common patterns encountered in contracts. The third is concerned with temporal verification of contracts and shows how the techniques employed in model checking for hardware specification can be transferred to the domain of contracts.

ACKNOWLEDGEMENTS

The research that led to this dissertation and the write-up have been a long (and at times turbulent) process, during which many people offered me intellectual, financial and moral support.

I owe a huge debt of gratitude to my supervisor, Marek Sergot, which is difficult to express in words without sounding trite. Since the beginning, Marek has guided me patiently and continuously. He taught me how to read and how to write, what questions to ask and where to look for answers, generously sharing his experience, ideas and insights with me. He encouraged me to develop my ideas, challenged me to improve them, reassured me when I was despairing and inspired me to forge ahead, without ever complaining or losing his patience. If this dissertation says anything useful or interesting it is very much due to Marek—however, I am solely liable for any mistakes and misunderstandings. He has been the best supervisor I could possibly have hoped for and I am privileged to have been his student.

I am grateful to British Gas Plc. for the financial support it provided me during 1992–1995. I would especially like to thank Paul Cartledge, Philip Colby, Tony Fincham, Mark Green, Patrick Leonard and John Piggott for the opportunity to work on this topic, their helpful comments during the development of MODELLER, their generosity in providing me with sample contracts and their hospitality during my research visits to British Gas Research and Development at Loughborough.

I have had the privilege to conduct the main part of my doctoral studies within the Logic Programming Section of the Department of Computing at Imperial College. I am grateful to all the members of the department for creating an exciting and stimulating environment for me to work in and for giving me opportunities to assist them with teaching activities and supplement my scholarship stipend. Special thanks are due to Krysia Broda, Derek Brough, Keith Clark, Chris Hogger, Bob Kowalski, Chris Moss, Fariba Sadri, and Silvana Zappacosta. While at Imperial, I shared lots of coffee and lots of conversations about common difficulties and achievements with fellow PhD candidates (Ber, David, Kosta, Kosti, Maria, Niko, Paulo, Priscilla, Sonia, Zac—and those of you I forget inadvertently—thank you).

I am grateful to colleagues at Brunel University and my students and personal tutees for being so understanding when faced with the “do not disturb” notice on my office door during the last months of the write-up (I would have been even more grateful if they had observed it!). Thank you Chris Reade, Chris Reed, John Salt, Simon Taylor and Robert Zimmer for putting up with me when I was going on about my thesis, for exciting and heated discussions in BRAINIER seminars and the pub, for stimulating ideas and plans for collaborative work (some of which have happily materialised) and for being such great friends.

I am grateful to many researchers who have generously provided me with their papers, explained their ideas, taken an interest in my own work and commented on it, at one time or another, in the course of my doctoral studies: I would like to thank Layman Allen, Andrew Jones, Steven Kimbrough, Marc Lauritsen, David Makinson, Tim Norman, Henry Prakken, Giovanni Sartor, Richard Susskind, and Daniela Tiscornia.

Last but by no means least, I am grateful to my parents and brother for their endless patience, optimism, support and love.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1

INTRODUCTION

1.1 Overview

The aim of this research is to explore the potential for developing logic-based tools for the analysis and representation of legal contracts. Over the last twenty years or so, a growing body of research in Artificial Intelligence has focused on the representation of legislation and regulations. The motivation for this has been twofold. On the one hand, there have been opportunities for deploying computing technology in the field of Law to provide useful applications that facilitate aspects of legal practice. Consequently, legislation formulation and drafting, legal information retrieval, preparation of briefs, execution of procedural law and so on have been areas of legal and para-legal practice for which tools and systems have been developed. On the other hand, the Law is a complex field in which diverse modes of reasoning are employed and a variety of processes are in place, offering ample opportunity to test existing Artificial Intelligence techniques and to develop new ones. Consequently deductive reasoning, argumentation, case-based reasoning, default reasoning, open textured concepts, legal language clarification and so on have been at the centre of various research projects[1].

Just as the primary function of legislation is to regulate human behaviour in a social context, the primary function of legal contracts is to regulate the behaviour of typically two parties (or more in the case of multi-party agreements) in a professional context. The main activities associated with legislation, which centre on its formulation and its application, have their parallels in the narrower domain of legal contracts. Legal contracts themselves, as putative entities that are governed by legislation, inherit many of the general features associated with Law. They are concerned with legal concepts, they stipulate mechanisms for their application and when violated they admit to resolution through a court of law, albeit this is not the only mechanism for resolving contractual violations. Where legal contracts are recorded in writing, the contractual document is expressed in similar stylistically structured language as that used for legislation, it is typically non-linear with cross-references between provisions, and often contains open-textured concepts.

Given this similarity in nature, purpose and associated activity between legislation and legal contracts, the idea of transferring techniques developed for the analysis and representation of the former to the domain of the latter has come up from time to time. The topic however has not been explored further. A possible exception is Gardner's work (1987), though that concentrates on contract formation rules rather than the content of contracts. Some authors have even seemed to suggest that the development of electronic tools to support contractual activity is uninteresting, either because legal contracts themselves as putative entities are relatively simple, or because the activity associated with them is straightforward and automation would be an unnecessary luxury, or worse a complication.

The ESPRIT-funded ALDUS[2] project (ALDUS 1992), for instance, conducted a preliminary investigation into the potential for developing drafting tools for sales contracts. Its conclusion, though ambivalent in parts, was that that there were no real opportunities for developing economically useful tools in this area. While the ALDUS conclusion might be true for relatively simple legal contracts such as the ones typically found in the area of sale of goods, might not be true of legal contracts in the large. The research described in this dissertation was conducted in collaboration with British Gas Plc., one of the biggest—in size and in profits—organizations in the U.K. to investigate the potential for the development of useful tools for *engineering* contracts. The approach adopted for the conduct of this investigation relied heavily on the examination and analysis of sample contracts, informed by the general principles of Contract Law and by the comments provided by the industrial collaborators.

[1] For recent developments in this area, see for example the proceedings of the International Conference on Artificial Intelligence and Law (especially 1993; 1995; 1997; 1999).

[2] **A**rtificial **L**egal **D**raftsman for **U**se in **S**ales.

The initial context of the collaborative project with British Gas was to explore the possibilities for developing tools to support the drafting, administration and management of legal contracts. The term "contract" is commonly understood as referring sometimes to a legally binding agreement, and sometimes to the document in which this agreement is recorded. In this dissertation, where it is important to distinguish between these two meanings the terms "agreement" and "document" are used accordingly. Where such distinction is not necessary, the term "contract" is used.

The types of contracts that were used as our main experimental material concern the purchase of natural gas[3]. In common with contracts in many engineering fields, they are recorded in writing and the associated documents are large and complex. They cover details of pricing and payment, supply schedules, quality assurance, maintenance of equipment, *force majeure* provisions, assignment and subcontracting, arbitration of disputes and so on. Some of these contents are typical of contracts in general—such as the specification of the period of the agreement, delivery quantities, prices, billing and payment arrangements and so on. Some are particular to engineering contracts in general—such as arrangements for the provision of technical services, the conduct of technical reviews, the appointment of experts, insurance arrangements, warranties, indemnities and so on. And some are specific to the particular kind of engineering—extraction of natural gas from a hydrocarbon field cannot be turned on and off at will and consequently there are many complex provisions dealing with shutdown procedures, adjustments for over- and under-delivery, monitoring of quality and so on.

A typical document runs to 200–300 pages, not including detailed drawings and technical appendices. Apart from their size, contractual documents of this kind are complex in two senses. They consist of a number of closely interrelated sub-agreements, each dealing with some aspect of the overall agreement, and some of the individual provisions contain complicated details, which are difficult both to follow and to apply in specific cases. A typical contract has a life span of 20–25 years with a review every five years or so. It is difficult for parties to foresee all eventualities that might arise, caused by factors external to them (such as a change in legislation or in interest rates) or caused by changes within their respective organizations (such as changes in organizational policy, structure and broader business goals). Consequently contractual documents leave many details only vaguely

[3] Besides sample contracts provided by British Gas, other contractual documents from the broader engineering sector were considered, such as model form contracts issued by the Institution of Electrical Engineers which are often used as the basis for British Gas contracts. Appendix C lists the main documents that were consulted.

specified and contain provisions that determine how the applicability of other provisions might be adjusted in the future.

Contract formation is typically undertaken by a team rather than one individual. During negotiation, which is typically a lengthy process, several drafts of contractual documents are produced and the associated costs often account for a significant proportion of the cost of the project as a whole. This is because contractual documents are sensitive and an omission or a mistake might have significant financial and legal consequences—especially in the light of the *parol evidence rule* under English Law, whereby contractual documents are typically regarded as definitive in establishing all that the parties agreed upon and only that.

While an agreement is in force, the associated documents are frequently consulted (parts of them even on a daily basis, for example to establish the required delivery quantity at the time) by staff with varying degrees of familiarity with their overall contents and legal effects.

Our investigation painted a very different picture from that of ALDUS. Three broad areas where electronic support for contractual activity would be of practical value were highlighted:

(i) Management of contracts: This refers to the general problem of storing and retrieving whole or parts of contractual documents.

(ii) Administration of contracts: This refers to systems that perform specific tasks during the operationalisation of an agreement, such as monitoring the compliance of contracting parties, or advising on the effects of individual provisions or on the procedures to be followed under certain circumstances.

(iii) Drafting of new contracts: This refers to tools that assist drafters in the creation of new contractual documents.

The research reported here focuses mainly on the third area. This choice was motivated by two factors. First, of the three areas of contractual activity that were outlined above, contract drafting is in practice the one associated with the highest costs (in time and manpower). Second, a framework for representing contracts for the purpose of drafting naturally lends itself as the basis on which to add extensions to support contract management and contract administration.

Prior to this research, some work already existed in the area of legal document drafting, although it did not specifically address legal contracts. The first system for legal document drafting reported in the literature is the ABF processor, developed by Sprowl (1980) as part of a project within the American Bar Foundation. The ABF processor dealt with the drafting of standardised legal documents and was based on Allen's (1957) proposal for a normalised language for legal drafting. The processor effectively provided a special-purpose imperative language for writing programs that can generate certain types of documents (most of the documented examples provided concern the drafting of wills). Sprowl's approach is procedural and relies on analysing legal documents so that all possible variations in them are identified and encoded algorithmically. This may be feasible for short and not very complex documents but not for the kinds of documents that we are interested in. Moreover, it requires that the drafter (the user of the processor) be familiar with the special-purpose language if he wishes to draft a document of a different structure and contents from the ones already provided by the processor.

The other major computer system for legal document assembly is Scrivener (marketed by the Dianoetic Development Company since 1989 and first developed by Evans (Evans 1990)). It is more flexible than the ABF processor; it contains an explicit tree-like representation of the syntactic structure of a document and the drafting session is conducted in interactive fashion with the drafter providing answers to questions presented to him by the system. From the user's perspective, the system behaves like an expert system shell with query-the-user facilities. However, as is the case with the ABF processor, the text of a document and the mechanisms that manipulate it (such as procedures for eliciting parameter values from the user, or conditions for the inclusion of certain document components) are amalgamated. References to other work and subsequent developments are provided in chapter 4.

The central position of this dissertation is that, exploiting the dual meaning of the term "contract" the drafting problem can be addressed at two levels. At the *macro-level*, the emphasis is on deciding what components to include in a contractual document and how they relate to each other so that the whole is structurally coherent. At the *micro-level*, the emphasis is on deciding the contents of the agreement in terms of detailed provisions so that the whole is consistent and complete, that is, it covers all intended cases without giving rise to conflicting provisions.

Contract drafting at the macro-level is viewed here as a variant of computer-aided design activity. The idea is not novel and had been suggested by other researchers (such as Fiedler (1985) and Gordon (1992)) but it had not been developed into a practical application. Based on this idea and drawing upon research into syntactic document models from the Electronic

Publishing community and the functional organisation of text from the Computational Linguistics community, we developed MODELLER, a prototype document assembly system for contract drafting.

The main features of MODELLER are:

(i) Generic documents, that is, document *types* rather than individual instances, are represented as collections of re-usable components. Each component corresponds to portions of text and may have multiple *versions*, that is, it may correspond to multiple variants of portions of text. Generic documents are *dynamic*. They are initially defined based on model forms or previous examples for their corresponding document type but new versions of components, created during the drafting session, are incorporated in them (see (iii) below).

(ii) Syntactic relations between document components (such as containment relations) and functional dependencies between them (such as necessary inclusion, exclusion and so on) are represented explicitly and constrain the drafting process.

(iii) Individual documents of a given type are created by instantiating a generic document for that type, interactively with the drafter, who chooses what components to include in the document instance and what version of such components. In this sense, the drafting process is precedent-based. The drafter's choices are constrained by the syntactic relations and functional dependencies between document components. In other words, document instances are negotiated between the system and the drafter with the system *critiquing* the drafter's choices in similar spirit to Hammond *et al.* (1993). The drafter however is not limited to choosing one of the available versions for each component. If none of the available versions is appropriate for his purposes at the time, he may create new ones, which become incorporated in the generic document and are available for future users.

(iv) As document instances for a given type may share versions of components, they are not represented and stored explicitly as text. Rather they are represented and stored as collections of indices to the text files of their components. Apart from avoiding unnecessary replication of data in memory,

> this allows for greater flexibility in document instance maintenance and administration. For example, should a change in organizational policy or external legislation require the re-wording of a portion of contractual text, the corresponding component is the only thing to be altered and all document instances using that component are ensured to incorporate that change. This form of storing and representing document instances is sufficient for constructing their whole text in the desired form (plain text, SGML or HTML for example) through separate mechanisms that manipulate the representation. The text of a given document is consequently separated from the mechanisms that manipulate it.

These features put the MODELLER approach in a different category from that of previous systems. The framework for representing contracts at the macro-level is easily extended to support their management. Document instances, or parts of them, can be retrieved not only by keyword but also, to some extent, by content as they are already indexed along their components. As regards contract administration, the framework can in principle be extended with executable micro-level representations of document components.

The MODELLER approach has influenced to some extent recent work in Artificial Intelligence and Law, as witnessed by dedicated sessions on automated drafting in subsequent ICAIL[4] conferences and workshops and by developments, most notably by Karl Branting and his associates. This work is discussed in chapter 4.

Contract drafting at the micro-level, that is, the design of detailed provisions of a given agreement, raises a host of diverse issues, ranging from representing the detailed wording of individual sections, to representing the nature of provisions (obligations, powers, reparations, procedures and so on), to representing their “fitness” or effectiveness in securing some party’s best interests. Various techniques are available to assist in aspects of this task, such as disambiguating contractual provisions and detecting inconsistency or incompleteness—such as Logic Programming, Allen’s normalised legal language, the Kanger-Lindahl theory of normative positions, recently substantially extended by Sergot, and Alchourrón & Bulygin’s theoretical framework for the formal properties of normative systems. The dissertation discusses these techniques and seeks to identify the extent of their applicability.

[4] **I**nternational **C**onference on **A**rtificial **I**ntelligence and **L**aw.

Our investigation into the detailed contents of engineering contracts highlighted the diversity in nature and purpose of their contents. Although contractual obligations are perhaps the first notion that springs to mind when discussing contractual content, we found that apart from prescriptions contracts contain many other types of provisions deserving attention. For example, all contracts contain definitional provisions that define the intended meaning of certain terms. Some contracts, particularly engineering ones, contain specifications of procedures to be followed under given circumstances in order to establish certain states of affairs (for example to have an expert appointed, or to terminate the agreement earlier than normally expected, or to process *force majeure* claims). Engineering contracts also contain algebraic provisions that provide formulae for the calculation of varying parameters (such as pricing adjustments or quantity adjustments where over- or under-delivery has occurred). Clearly not all parts of such contracts are useful to represent in detail, nor is a single representation scheme appropriate for all kinds of provisions. Definitional provisions for example lend themselves naturally to logic programming formulations. But they need to be sufficiently complex to warrant the effort of constructing such representation, otherwise the enterprise is laden by questions of vagueness and open texture. Since they constitute executable specifications, logic programming representations of the complex provisions are useful for the formulation of provisions and also for their management and administration as evidenced in the relevant literature on applications of logic programming to the representation of legislation and regulations (for example (Sergot 1988; 1991), (Bench-Capon 1987), Kowalski (1995)). The last part of the dissertation identifies and discusses the representation of some common patterns of contractual obligations encountered in contracts, such as obligations directed from bearers to counter-parties, contrary-to-duty structures and temporally qualified obligations. The dissertation takes a process view of contracts, where obligations are entities that are created, persist over some time and are extinguished (naturally through fulfilment or unnaturally through violation). From the perspective of contract drafting, such a view is useful as it allows drafters to establish the effects of provisions during the course of an agreement. Possible violation of norms, the sub-ideal state it leads the agreement into and possible reparatory courses of action to return it to a normative state are represented explicitly.

Our investigation into micro-level representation of contractual content gave rise to another observation. When considering the detailed provisions of contracts, it is difficult to demarcate precisely between issues pertaining to the design of documents and issues pertaining to the design of agreements. Following a similarity between software and legislation—which has been observed by many researchers—whereby the formulation of legislation is akin to program specification and legislative drafting is akin to program implementation, it is natural to liken agreement design to specification and document drafting to implementation. In practice however, contract creation is not carried out in two distinct phases, one in which the agreement is designed and agreed (negotiation) and one in which the corresponding document is drafted. In engineering domains and generally where contracts are not formed on one party's terms alone on a "take it or leave it" basis, negotiation itself is typically conducted on the basis of draft documents and often is *about* document content itself, for example in terms of the particular wording used for some provision. The way in which the final agreement (which becomes the contract) stipulates that the business exchange will be realised comes about as a result of various exchanges between the parties. During such exchanges, offers are made, promises are given, facts are asserted and questions are asked to clarify various points and to determine specific values for certain parameters.

Researchers in electronic commerce have recently become interested in the nature and outcome of such exchanges, both in order to construct appropriate tools to support electronic contracting and in order to explore the essential notions of contract formation discourse. A promising strand of such research, noting the limitations of current EDI[5] standards, seeks to develop a formal language for business communication based on speech act theory (Kimbrough 1998). Speech act theory, originating in the work of Austin (1962) and Searle (1969), addresses performative statements, that is, statements that are not true or false of the world but rather achieve certain states of affairs when uttered. For example, the existence of a promise made by a person to some other is merely described truly or falsely by a simple proposition. The existence of such promise however creates an obligation on the part of the promissor to keep it, to see to it that what is promised becomes the case. Chapter 5 discusses Kimbrough's work on the formalization of performative statements for the purposes of business communication and, focusing on statements of a promissory nature, extends his proposal so that contractual obligations can be explicitly represented as the effects of such statements. The representation of obligations can then be integrated with a formal language for business communication such as that proposed by Kimbrough, so that what parties negotiate about in the course of agreement design becomes explicit.

Finally, the dissertation turns to temporal aspects of contracts. It explores the view that questions about the consistency and completeness of temporally qualified norms in the context of micro-level contract design are comparable to verification questions that arise in the context of software and hardware specification. A number of formal techniques have proved effective in establishing whether a hardware system specification has the appropriate safety ("nothing bad will happen") and liveness ("something good will happen") properties. Some recent work in this area is known as *model checking* (Holzmann 1997) and employs temporal logic for the expression and verification of safety and liveness properties of a system. The dissertation examines how such techniques can be usefully applied to contracts (viewing the latter as specifications of business exchanges) and demonstrates this via a worked example of a simple contractual scenario. The underlying representation of our specification for model checking is a Petri net and the dissertation shows how Petri nets can be constructed for contractual settings drawing upon previous work (Bons *et al.* 1995) on the specification of trade procedures.

1.2 Organization of Dissertation

The main body of this dissertation is organized as follows:

Chapter 2 aims to familiarize the reader with the domain of investigation and to clarify the concepts mentioned subsequently in the dissertation. To this end, it discusses contracts and contractual obligations from the point of view of Contract Law, it presents briefly the explication of legal concepts (such as "duty", "right", "power" and so on) often associated with Hohfeld (1913), which are adopted here and offers an account of the characteristics of written agreements and the kind of engineering contracts that were used as experimental material.

Chapter 3 aims to familiarize the reader with background techniques and theories that have been applied to legislation and regulations, mainly from the domains of Artificial Intelligence and Deontic Logic. It briefly discusses such techniques and illustrates how they could be applied to legal contracts and also discusses in more detail the limits of their applicability.

Chapter 4 is concerned with macro-level document drafting. It presents the rationale and the approach for MODELLER and discusses its relation to other researchers' work before and after its development. Appendix A contains the complete representation of a generic document in MODELLER, as a full example that complements the discussion in chapter 4. Appendix B notes all the documents that MODELLER was tried on and all the documents that were

[5] **E**lectronic **D**ata **I**nterchange.

consulted to determine whether they would lend themselves to MODELLER without constructing explicit representations for them.

Chapter 5 turns to the micro-level representation of contracts. It comprises three discussions. The first is on contractual obligations as the result of promissory exchanges between parties and draws upon work by Kimbrough and his associates. Some notes on the classification and structure of performative statements in speech act theory terms facilitate this discussion and are included in Appendix C. The second concentrates on contractual obligations and common patterns encountered in contracts. The third is concerned with temporal verification of contracts and shows how the techniques employed in model checking for hardware specification can be transferred to the domain of contracts.

Finally chapter 6 summarises the findings of this research and discusses directions for future work.

2

THE DOMAIN OF INVESTIGATION: CONTRACTS AND CONTRACT LAW

2.1 Introduction

The aim of this research is to explore the potential for developing logic-based tools for the analysis and representation of legal contracts. This chapter discusses legal contracts from the general perspective of Contract Law, and explains the notions of the legal concepts that they involve. Finally, it discusses in more detail the particular kinds of legal contracts that were considered in the course of this research.

A reader's intuition about contracts is more or less that they are agreements entered into freely by a party with at least one other, to deliver goods or services, or to do something in return for some consideration (usually financial), on mutually agreed and binding terms, often in writing[6]. In cases where an agreement is put in writing, the term 'contract' is commonly used to refer both to a legally binding agreement and to the document that records it. The Law views contracts (agreements and their associated documents, where they exist) as entities that are created at a given point in time, persist over some specified period and then are extinguished (naturally by fulfilment, or unnaturally by early termination as we shall see later). Should litigation arise, at any point during the lifetime of a contract, courts of law are called to resolve the dispute between the parties by establishing what the parties agreed to do. This task often calls for interpretation, on the part of courts of law, of the terms of the contract, which in turn may require interpretation of the parties' intentions at the time of entering contractual relations. The task is far from easy but where agreements are recorded in writing the procedure is facilitated, as it is generally held that written documents offer better evidence of the intentions of the parties than verbal claims. However, it is noted later in section 2.2 that apart from terms that parties explicitly incorporate in their agreements, additional terms are viewed as belonging to them, especially where contracts concern transactions that are regulated by statutory legislation.

Section 2.2 also shows that the Law essentially regards contracts as obligations of a specific character. As we shall see, there are a number of legal notions that are encountered in contracts, such as ‘duty’, ‘right’, ‘power’ and so on. Such concepts are often associated with the legal scholar Hohfeld, who attempted to provide a systematic account of them and their interrelationships. section 2.3 presents briefly a semi-formal account of Hohfeld’s (1913) explication of such notions to provide the framework within which such terms are used in the context of this research.

Given the long and established record of research that sought to apply Artificial Intelligence techniques to legislation (see Sergot 1991 for an overview) the idea of applying similar techniques to the representation of contracts is not new, and has in fact been emerging from time to time, as contracts serve a function similar to that of legislation: they are meant to regulate the actions of (usually) two parties while they interact (usually in a professional context). The topic however has not been explored in depth. Some authors have even seemed to suggest that the development of electronic tools to support contractual activity is uninteresting, either because the domain of investigation—contractual content—is comparatively trivial, or because the tasks associated with the domain—contractual activity—are straightforward and do not require automation. The ESPRIT-funded ALDUS project (ALDUS, 1992) investigated the potential for developing systems to assist with the drafting of contracts. It concluded that there were no real opportunities for developing economically useful tools. Our view, however, is that such projects have looked at the wrong kind of contracts. ALDUS concerned itself almost exclusively with the Sale of Goods, where contracts do tend to be very simple. However, not all contracts are as simple as that. In other areas, both contractual content and contractual activity can be extremely complex, and automated support can be time-saving and cost-effective. The development of appropriate tools is challenging: knowledge elicitation and representation require the integration of many

[6] This is covered by standard dictionary definitions of the term ‘contract’ as a noun or verb, for example (Collins 1994).

paradigms from diverse areas of Artificial Intelligence and confront a number of fundamental representational problems (these are also discussed in Daskalopulu and Sergot 1997).

Even in the case of Sales of Goods contracts however, contrary to the conclusions reached by the ALDUS project there are interesting questions to pursue, as shown by recent work on the representation of the United Nations Convention on Contracts for the International Sale of Goods. Such work has been at the core of a series of workshops held in conjunction with the International Conference on Artificial Intelligence and Law since 1995, and a major Japanese project is reported in (Yoshino 1997; 1998).

In section 2.4 we turn our attention to engineering contracts, which were the main type used as a sample for this research. Engineering contracts are typically large and complex agreements ranging over long periods of time and offering a number of opportunities for the development of electronic tools to support activities associated with them. These are outlined in order to place the discussion in chapter 3 in context.

2.2 Contracts as the subject matter of Contract Law

The Law of Contract is defined by some legal theorists as the law of obligations "which people incur to others as a result of the relations and transactions in which they become involved" (Atiyah 1989). Such obligations are incurred through promises that people make to others with respect to some action they will perform or a state of affairs that they will attain to. In this light, a contract as a whole (whether oral or written) can be regarded as an obligation taken up by both parties involved to perform as they agree. Such performance is specified in more detail in the terms of a contract, where parties are ascribed particular obligations; hence, the Law regards contracts as *collections* of obligations.

Contractual obligations are distinct from other kinds of obligations (such as social, political and so on) in the following respects, according to legal theorists (cf. Atiyah 1989):

(i) they are of a *private* character as opposed to political or constitutional obligations, that is, they only apply to those parties that have assumed them;

(ii) they are owed by some person to some other person specifically, rather than the public in general;

(iii) they are *enforceable* only by the people to whom they are owed, that is, a public authority will not intervene to see to it that they are carried out unless required to do so by the people to whom they are owed;

(iv) they are *self-imposed* in that people assume them of their own free will and usually after negotiation[7].

Contractual obligations are usually classified into two kinds: Contractual duties *stricto sensu*, which are imposed by the contract into which parties have entered and are relevant to its substantive aims, its performance. To illustrate such contractual duties consider the example of a sales contract between a buyer and a seller of certain goods. The substantive aim of the contract is that the buyer acquires the goods and that the seller is rewarded financially for them. Hence the buyer's obligation to pay the agreed price and the seller's obligation to deliver the specified goods by a certain date are examples of *stricto sensu* contractual duties—without them the substantive aim of the contract is not realised. A second kind of contractual duty that legal theorists identify are those imposed by the law independently of any particular contract; these relate to contract formation and the procedure by which a contract is concluded. Such duties are relevant to the notions of fairness and justice in forming and concluding contracts. For example the Law regulates the way that contracts are negotiated so that unfairness is avoided and appropriate penalties are imposed for non-performance. Such regulation implicitly imposes on contracting parties the duty to behave in the prescribed way in forming and concluding their agreements. Such duties imposed by the Law independently of any particular contact centre around the notion of misrepresentation, duress and undue influence, unfair contracts and illegal contracts (cf. Atiyah 1989). The main penalty imposed, if parties depart from prescribed conduct in the making or the conclusion of the contract, is that the latter is deemed void, that is, it is not binding and a case for it cannot be made in a court of law in order for it to be enforced. In the case of duress and illegal contracts there may arise other associated penalties, for example parties that entered an agreement whereby one kills in return for a reward from the other may be liable to be prosecuted for intention to kill. However, these do not concern us any further here, since it is the content of legally formed contracts that we are examining. Hence, here we are concerned with contractual duties *stricto sensu*. As mentioned earlier their central notion is that of (legal) obligation or duty.

[7] Having said that, for transactions that are governed by statutes or acts of parliament, e.g. sales that fall within the scope of the Sale of Goods Act, parties are expected to assume a number of obligations (*implied terms*, see also section 2.2.4) so the 'freedom of contract' is not altogether unrestricted. As Atiyah (1989) points out the idea of 'freedom of contract' involves two closely

Despite the differences in character (constitutional, legal, contractual and so on) it is possible to identify a common underlying structure for all kinds of obligations. Such a structure comprises:

(i) The subject, or bearer, of the obligation, that is, the person that assumes it.

(ii) The person(s) to whom the obligation is owed, or the beneficiary.

(iii) The object or content of the obligation, that is, the action that is meant to be performed or the state of affairs that is meant to be achieved through some action on the part of the bearer. An interesting ontological question arises as to whether obligations concern actions or states of affairs. Sometimes specific actions are determined through which a state of affairs is achieved (hence the content of an obligation concerns the desirable means to achieve an end); sometimes only the end, a desirable state of affairs, is specified without an explicit reference to actions that bring it about. One might regard these two views as imperative and declarative forms of specifying obligations.

(iv) The conditions under which the obligation comes into force. Some obligations hold unconditionally, for example the obligation for a buyer to pay for goods holds in all purchase situations. Some obligations however are imposed on subjects that meet certain criteria, or upon the occurrence of an event or the existence of a specific state of affairs. For example, an obligation to pay for parking in a private garage is imposed on a driver only if he enters the garage and leaves his car there. An obligation to pay rent is imposed on a subject following an event through which he becomes a tenant of certain premises.

(v) The temporal bounds of the obligation, that is, the period of time during which it is in force. Some obligations can be viewed as holding at 'all times' (after they come into force, that is, after commencement); the obligation to pay salary to an employee comes into force once the latter commences employment and is an instance of this. Others have definite temporal bounds; for example a tenant is obliged to pay rent from the point in time at which he starts occupying certain premises and until such occupation is terminated.

related but distinct concepts: that contracts are based on mutual agreement, and that the creation of a contract is the result of a free choice "unhampered by external control such as government or legislative interference". See also (Atiyah 1979).

(vi) The penalty that is imposed to the bearer of the obligation if he does not satisfy it. Or alternatively the repair available to a party that suffers as a result of a subject's violation of an obligation. For example, the repair available to a property owner whose tenant does not pay rent is eviction of the latter. This notion of penalty, or repair can also be referred to as sanction or punishment.

The notion of sanction (or punishment, or penalty or repair) is an interesting one because it creates new legal relations between the bearer of an obligation and the person(s) to whom it is owed. Hence if the bearer of an obligation violates it, he is obliged to comply with the penalty, and the person that suffered a loss through the violation of the original obligation is entitled to repair (and hence has the right to claim performance of the penalty from the bearer). This interestingly shows that the violation of contractual duties *stricto sensu* may bring about other duties *stricto sensu* or even duties independent of any particular contract. The notion of sanction is relevant to the notion of violation or non-fulfilment of an obligation, and this in turn may have a temporal dimension, that is, the sanction may be applicable within or over a specified time period. Frequently the sanction imposed on a party that fails to fulfil an obligation is an obligation itself (to pay damages, or to perform some other remedial act). Such secondary obligations have been discussed extensively within the broad area of deontic logic, as we shall see in chapter 3.

Contractual obligations follow the general structure outlined above, which is illustrated in Figure 2–1.

There are other conceptual models for contractual obligations that are similar to this one. For example, Santos & Carmo (1993) examined the Portuguese Law of Contracts and made the following observations about the general model of obligations; their points are applicable to contractual obligations under English Law as well (and in fact any legal system to the best of my knowledge). Contractual obligations are regarded as *credit* relations, where the bearer is the *debtor* and the person to whom the obligation is owed is the *creditor*. They define jural relations where the right assigned to one of the subjects corresponds to a duty to perform an act or attain to a state of affairs specifically imposed on the other subject of the relation. A contractual obligation is an entity that is created, can be modified and persists for a definite period of time, at the end of which it is extinguished.

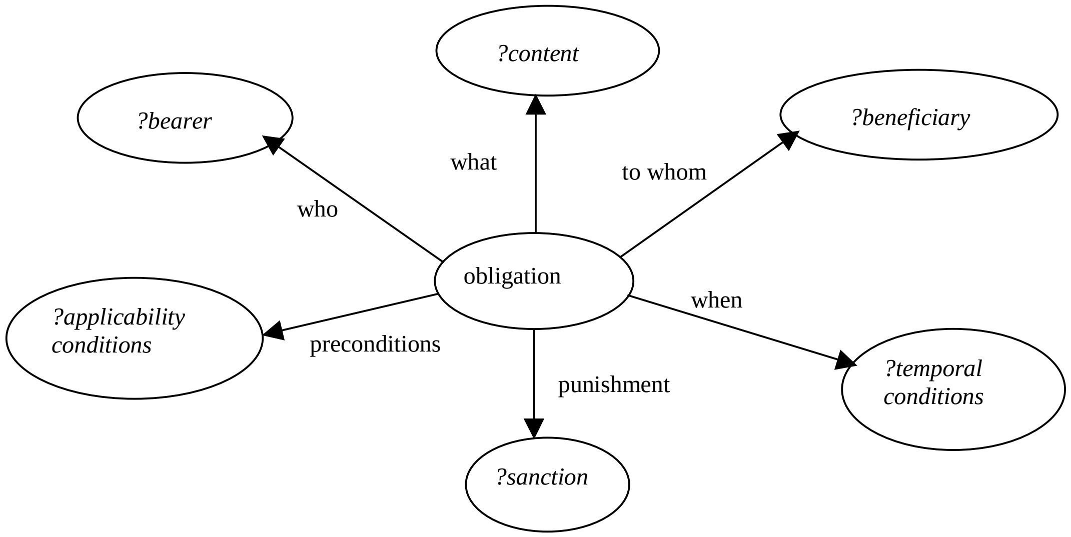


Figure 2–1 The structure of contractual obligations

The object of the obligation, that is, the performance of an act, or the achievement of a state of affairs, can be instantaneous (for example the payment for goods bought over the counter), continuous (for example the execution of one's employment duties) or periodic (for example the payment of rent that takes place at specified recurring points of time, such as the beginning of a month, or a week). Creation of a contractual obligation occurs when an agreement between parties comes into force (and the latter may itself be subject to certain conditions obtaining). Modification of a contractual obligation may arise as a result of changes in circumstances, both within and outside the parties' control, during the performance of the agreement. Extinction can come about naturally as a result of fulfilment, or unnaturally as a result of breach of the obligation. This life cycle of a contractual obligation raises therefore questions that we may be interested in addressing: For example, when does an obligation come into force, when is it regarded as fulfilled, how is it fulfilled, what happens when it is not fulfilled. Answering such questions involves reasoning with and about obligations and it is useful from two perspectives: During the making of the contract, where negotiation takes place (and drafting of the relevant document, if the agreement is in writing) it is helpful to establish what the obligations of parties need to be in order for the contract to work. Specific concerns of contracting parties are for example whether any conditions attached to obligations are appropriate and likely to obtain in practice, whether there are conflicting obligations imposed on one or both parties, or whether there is vagueness, that is, whether no specific conduct is prescribed for the parties in certain circumstances. This can be seen as a task of *a priori specification* of obligations as Santos & Carmo (1993) point out. During the performance of the contract, it is useful to establish whether the actual behaviour of the parties coincides with the desired one specified by the contract. This is what Santos & Carmo (1993) call *a posteriori verification* of contractual obligations.

Some general remarks from the point of view of Contract Law need to be made, in order to inform and facilitate further discussion about these questions. These points are relevant to the conditions under which an obligation comes into force, its temporal bounds and the consequences that its violation entails. The first point concerns the classification of contractual obligations depending on the consequences that their violation entails. In the eyes of the Law, breach of an obligation may result in a contract being dissolved and both parties being discharged of any further obligations towards each other, or it may invoke new obligations/entitlements for one or the other party. This point is explored in section 2.2.2. The second point is relevant to the temporal bounds of an obligation and the consequences invoked if the obligation is not satisfied within the specified time; this is discussed briefly in section 2.2.3. Finally, section 2.2.4 discusses the way written agreements are perceived by Contract Law.

2.2.1 Classification of Contracts

As Atiyah (1989) notes, definitions of contracts typically rest on the notions of promise or agreement. For example, he notes that in English Law a contract is “an agreement which is legally enforceable or legally recognized as creating a duty”. The American Restatement of Contracts, on the other hand, defines them in terms of promises: “A contract is a promise or a set of promises for the breach of which the law gives a remedy, or the performance of which the law in some way recognises as a duty”. However Atiyah argues that due to problems inherent in such definitions one should refer to the “law of contracts” (covering Contract Law, Tort and Restitution) rather than to Contract Law alone. The problems he identifies enlighten the ensuing discussion about the nature and enforcement of contractual obligations, so they are worth mentioning here:

(i) Such definitions of contract assume that the law enforces them. However Atiyah notes that the law does not compel performance of a contract; rather it provides a remedy, usually by awarding damages, for breach[8]. We shall see in the next section the different kinds of remedy available to a party depending on the nature of the violated obligation.

(ii) Such definitions assume that agreements and promises are easily identifiable entities that exist outside the law. However, Atiyah notes that though words such as ‘promise’ or ‘agreement’ have recognized meanings in common speech, it is actually difficult in some cases to identify contractual relations without analysing the issue in legal terms. Indeed a lot of litigation is concerned with whether a contractual relation actually exists between two parties.

(iii) The definition provided by the American Restatement ignores negotiation and consideration aspects of contractual relations, in that it does not explicitly refer to a contract as a two-sided exchange, where something is promised or done by one party in return for something promised or done by the other party. However, Atiyah notes that not all contracts involve bargain elements. Other doctrines, which are very close to contract, such as *estoppel*, (where a promise is enforceable because it was relied upon by the promisee, and this is demonstrated by the promisee’s conduct) give rise to contractual relations, even though no specific agreement involving a bargain exists. Such situations that admit to *estoppel*, give rise to the Law of Tort and the Law of Restitution, the other parts of the law of obligations. The former seeks to protect a party by giving it a right to damages, where the party suffered a loss through reasonably relying upon the other party, even though no contract may exist between them (or even though, where a contract does exist, it does not make such provision). The latter imposes obligations on a party to reward (usually financially) another for benefits received, even where there is no such promise for such reward in an agreement or any contract between the parties in the first place.

(iv) Such definitions seem to assume that people enter contractual relations in a well-planned manner: first they make agreements or promises, then they become bound by their contract. Atiyah notes a number of situations where people enter contractual relations without any agreement of promise made

[8] As Atiyah puts it “[i]n practice this means that what a breaching party is ordered to do by a court is rarely what he actually

explicitly, for instance in simultaneous exchanges involving sale of goods over the counter, when a person boards a bus and buys a ticket, or when someone drives into a car park, collects a ticket and is expected to pay on exit. Atiyah notes that in many cases a transaction becomes legally binding because one party has *behaved* in a certain manner.

Atiyah concludes that though promises and agreements play a central role in defining the concept of contract, there are two other ideas that are very close to it. First, that "a person who induces another party to rely upon him and change his position, ought not to let that person down"; second, that "a person who does a service to another or renders him some benefit, ought generally to be recompensed for his trouble".

Atiyah's classification of contracts is done along several dimensions:

(i) By terminology used, alone, one can distinguish between *contracts of record*, *contracts under seal* and *simple* (or parol) contracts. Atiyah stresses that the first two bear little or no resemblance to ordinary contracts. This is because contracts of record are just obligations incurred by a court of law; they do not involve any of the notions of ordinary contracts, and are only called so because they are enforceable by the same type of action as is used for genuine contracts. Contracts under seal (also called deeds or covenants) involve written promises but they do not require the existence of consideration, or acceptance of the promissor's offer by the promisee (which is considered a necessity for the creation of ordinary contracts).

(ii) By considering formation and the binding on the parties, one can distinguish between *bilateral* and *unilateral* contracts. In the former a promise (or a set of promises) on one party is exchanged for a promise (or a set of promises) on the other party. An example is a sale situation where the buyer promises to pay a price for certain goods while the seller promises to deliver them. In the latter a promise (or a set of promises) on one party is exchanged for an act on the other party. An example is a promise to pay commission to an agency if it finds a buyer for the promissor's house. In bilateral contracts both parties are equally bound to performance of their promises, whereas in unilateral contracts only

promised or agreed to do […]".

the promissor is bound to perform (in this example, the agency may do nothing at all, but if it does find a buyer the promissor has to pay the commission).

(iii) By considering the legal categories in which they belong, one can distinguish between *express*, *implied* and *quasi contracts*. The first two belong to the category of ordinary contract (Contract Law), while the last one falls within the scope of the Law of Restitution. Express and implied contracts differ only in how the intentions of the parties have been expressed (explicitly in the former, implicitly in the latter and hence inferred from their conduct and general circumstances of the exchange). Quasi-contracts are at the core of the Law of Restitution, where it is held that a party that benefits due to another party ought to offer some reward. Despite the classificatory distinction, Contract Law and the Law of Restitution are closely linked and perform similar functions when remedies are decided by courts.

(iv) By considering their legal effects, one can distinguish between *valid*, *void*, *voidable*, *illegal* and *unenforceable* contracts. A valid contract is an ordinary contract in full force. A void contract is a contract rendered non-valid because of two main reasons: either the procedures followed for its creation are in contrast with normal requirements (no offer/acceptance, for instance or no consideration rendered) or the law disapproves of its objective or of the means it stipulates (the terms) for the achievement of its objective. Contracts where the acceptance of an offer has not been communicated, or where a promise is given with nothing in exchange, or where the terms are prejudicial to family relations are instances of void contracts. Atiyah stresses that a void contract is not strictly speaking devoid of effect. Where parties have conducted themselves as though they had made a valid contract, courts of law may, instead of dismissing it unfairly, adjust it so that it becomes valid. Similarly, a contract may be void on both sides, but where it is void on one side only, there may be legal rights created on the other side that courts still have to decide instead of dismissing the case. A voidable contract is a valid contract which may be avoided by one or both of the parties if this is desired (for example if a person enters a contract as a result of fraud, the contract is voidable at his option, but if he does not exercise this option the contract has full legal effect as though it were valid). An illegal contract is one whose objective contravenes criminal law (a contract to kill someone for example) or encourages something contrary to public interest (for example prostitution). An illegal contract is often void while the converse is obviously not true. An unenforceable contract is closely related to a void contract and usually arises when it is formed contravening a statutory requirement for a transaction to be evidenced in writing[9] but it cannot be litigated as easily.

(v) By considering the performance of contracts, one can distinguish between *executory* and *executed* ones. The former are totally unperformed, while the latter are performed or partly performed. The difference is important typically when the question of breach arises and damages are to be decided.

Finally, Atiyah notes that one can distinguish between contracts that are *transactions* and contracts that are *relations*. A typical transaction is regarded as a discrete event, which is formed and executed almost instantaneously or on a 'one-off' basis. Relations on the other hand, refer to exchanges that span a long period (such as employment or long-standing supply arrangements between companies). In these cases, the obligations of the parties cannot as easily be tracked back to a promise or an agreement, for they may have been modified in time because of day-to-day adjustments. The following passage from Atiyah (1989, pp. 56–59) illustrates some of the features of such relations, and is given here because it is precisely such contracts that this research focuses on, as we shall see in section 2.4.

> "In such relations norms often seem to emerge about the way obligations are created which do not conform to the classical paradigm of promise and agreement. For example, in relations agreement is often reached gradually over a period of time, rather than by a more or less instantaneous process. The original contract may need to be amended in ways which do not conform to the classical procedures for making and changing contracts. The parties may imperceptibly slide into a situation from which they would feel it impossible to withdraw. A concept of 'good faith', or of fidelity to the relationship becomes important. Dispute-settlement procedures may come to be needed which are not so adversary as those involved in litigation, because the parties may want to settle an argument amicably, while continuing their relations. […] For instance, any complex or engineering work will these days be carried out under a detailed contract which allows for all sorts of changes to be made to the work as things progress. […] The need for contractual relations of this kind to be flexible and to permit of change means that it often becomes necessary for such contracts to contain detailed arrangements as to *how* the changes are to be made. And when things reach a certain degree of complexity, the contract needs to have some kind of institutional arrangements built into it. For instance, a large building or engineering contract will vest many powers of decision-making in the architect or engineer in charge—he becomes a sort of arbitrator with power to adjust the rights and duties of the parties, within the terms of the contract."

[9] Contracts of guarantee and contracts for sale of land are such transactions that require written evidence, for example.

Having discussed the notion of contract and the differences in treating contractual obligations, we now turn to these in more detail.

2.2.2 Classification of Contractual Obligations

According to the Law of Contract contractual obligations *stricto sensu* vary in nature and importance. Their main difference lies in the remedies available to one party should the other party violate an obligation. Breach may arise in two ways: A party obliged to perform an act may fail to do so; or that party may fail to perform a temporally bound obligatory act *within the time period specified*. Where a contractual obligation requires a party to refrain from performing a certain act, it may be violated by the party's performance of such act; similarly, where a contractual obligation requires a party to refrain from performing a certain act within a certain time period, it is violated if that party performs such act within that time period. In both cases such breach gives rise to entitlement on the part of the innocent party to some remedy and to an obligation on the part of the party that violated the obligation to comply with the remedy. The precise kind of remedy available to the other party is sometimes pre-determined by the contract, if the parties had thought in advance, at the time of making it, about the consequences of possible violations. In practice, most contracts do not specify explicitly what the consequences of breach are in all possible cases of violation, as it is hard for parties to foresee all possible circumstances that arise. Also, as parties usually tend to anticipate performance they may not wish to specify in detail such penalties in advance; rather they may prefer to defer such decisions till breach arises in practice and assess its impact and hence determine the penalty in practice. Should litigation arise as a result of breach the courts try to establish what the parties had agreed on the issue. To assist in this process the courts regard contract clauses as being one of the following kinds:

(i) representations about states of facts;

(ii) promises that an action will or will not happen in the future (and sometimes such action is not in the control of the promissor); and

(iii) conditional clauses.

Legal theorists (cf. Atiyah 1989) regard (i) and (ii) as promises or "undertakings". As Atiyah points out, one may debate whether a representation about a state of facts is something that can be an "undertaking", but in practice many such representations of states of facts are taken as being of a promissory nature. To clarify this point Atiyah (1989) presents the example where a sale of goods contract is formed between a buyer and a seller: The buyer promises that he will pay for the goods and that he will accept delivery of them, and the seller promises that he will deliver the goods and that they are in accordance with some description and fit for the purpose for which the buyer intends them. Promises referring explicitly to actions are of type (ii); statements concerning fitness for purpose and description of the goods are of type (i) and of a promissory nature. Research in interpersonal reasoning also indicates that representations of states of facts are best viewed as undertakings in a dialogical context. Walton & Krabbe (1995) regard agents' statements about representations of states of facts as commitments: "...whoever makes a declarative statement thereby commits himself to some course of action, depending upon the context. In this respect, asserting (...) is rather like making a promise: it puts something on the speaker's agenda. (And, usually, it puts something on the listener's agenda as well)". In this sense, a seller that asserts that the goods are fit for the purpose for they are sold is committed to a number of things: For example holding that the goods are fit for purpose, not denying that the goods are fit for purpose, giving evidence if required that the goods are fit for purpose, proving or establishing that the goods are fit for purpose and so on[10].

Undertakings themselves are distinguished by English Law into *promissory conditions*, *warranties* and *intermediate terms.* The distinction between promissory conditions and warranties dates back to the Sale of Goods Act 1893 and it is important, as it is relevant to the kinds of remedies available to a party in cases of breach. Breach of a promissory condition by one party gives the innocent party the right to treat the whole contract as terminated and absolves him from performing any of his obligations from that point in time. Hence "the right to the other party's performance is conditional upon [promissory] conditions being performed" (Atiyah 1989). In contrast, breach of a *warranty* merely gives the innocent party the right to claim damages (and creates the obligation for the party in breach to see to it that such damages are properly awarded), but does not discharge him from performing his own obligations under the contract. Characterising contractual obligations appropriately as promissory conditions or warranties is therefore essential as the sanctions entailed in cases of violations have different legal effects.

As Atiyah (1989) points out, this classification of undertakings into promissory conditions and warranties is rigid, because it requires that contract terms are pre-labelled, as being of one or

[10] Searle (1969) also views assertions as undertakings (cf. Appendix C). Some researchers would disagree with this view that regards representations about states of facts as promissory. We do not expand on this issue here, as debates in speech-act theory

the other type, by specifying in detail the consequences of breach in advance, before the breach actually takes place and its effect is assessed in practice. Hence, courts recognise a third kind of undertaking, namely intermediate terms. Such terms are neither *a priori* promissory conditions nor warranties, and breach of them does not have pre-specified consequences agreed by the parties at the time of forming the contract. If an intermediate term is violated, then the courts assess the extent of the loss for the innocent party and decide whether to treat the term as a warranty or a promissory condition. In a sense, intermediate terms are analogous to open-textured concepts in Law, that is, concepts whose precise meaning is not pre-determined but rather decided in the context of a particular situation as it arises. In a given context, intermediate terms can therefore be deemed by a court of law to be one of either warranties or conditions.

As mentioned earlier there is a third kind of contract term, conditional clauses. These again are of various kinds, but Atiyah (1989) distinguishes two main types, namely *condition precedents* and *condition subsequents*. A condition precedent is “a clause on which the entire operation of the contract depends” (Atiyah 1989). Contracts that are subject to a condition precedent are called *conditional contracts* because if the condition precedent fails, then the contract is not operative. For example, A agrees to buy certain goods from B subject to their being approved by a third party. If such approval is not obtained, then the contract need not be performed at all; that is, A is not under any obligation to accept and pay for the goods, and B is not under any obligation to supply goods that comply with any description at any particular time. A condition subsequent is “a condition on the happening of which an obligation (or sometimes the whole contract) is terminated or dissolved”. By this definition, it appears that a condition subsequent is very similar to a promissory condition. Atiyah (1989) however points out that this is not so because a condition subsequent does not always involve parties in any liability. For example, two parties entering a sales contract may agree that in case of war, the contract is dissolved and neither bears any obligations towards the other.

Such a classification of contract clauses is based on the consequences of their violation. Distinguishing between them is not merely interesting from a philosophical and legal-theoretic perspective but also useful in practice since the violation of a particular obligation may dissolve all legal relations between parties, or create new ones. The distinction of contract clauses is not always made in a straightforward manner as they are not always clearly labelled by the parties as being of one kind or another. Moreover as Atiyah (1989) points out,

fall outside the scope of this research.

even in the cases where parties explicitly agree on the status of each contract clause, “the terminology used by the parties is not necessarily decisive, for they may have been using the words incorrectly, without full appreciation of their proper legal significance”. In the light of this, should litigation arise as a result of breach of a clause whose status is unclear the courts assess the impact of the breach in order to determine the consequences that it invokes depending on “whether justice and convenience are best served” one way or another.

2.2.3 Stipulations about Time

As mentioned earlier, an obligation to perform a particular action can be violated if the obligation is temporally bound and the bearer fails to perform within the specified time period. The consequences of such violation may again range from dissolving the contract to creating new legal relations between the parties; the precise form of the consequence depends on whether time was ‘of the essence’, that is, whether performance of the act in question within the specified time period was a promissory condition. Where a contract explicitly specifies that a party’s failure to perform a certain obligation within a specified time period gives the other party the right to terminate the contract, then the corresponding clause is taken to be a promissory condition. In many cases however contracts do not explicitly state whether time is ‘of the essence’. As Atiyah (1989) points out, in commercial contexts the traditional English rule applies: “stipulations as to time are prima facie to be treated as of the essence”. In some contexts however this rule is not followed; instead the general rule of equity applies whereby “time is not ‘of the essence’ of contracts in the absence of a contrary intention”. Assessing the intention of the parties is a question of construing the contract and courts proceed depending on how they see justice and convenience best served. However written evidence plays a decisive role, as we shall see in the next section.

2.2.4 Written Agreements

The general points about contractual obligations made in the previous section concern both oral and written agreements. English Law does not formally require that an agreement be put in writing in order for it to be legally binding[11]. As Atiyah (1989) points out this absence of formal requirements reflects the fact that intention plays a central role in classical contract

[11] With very few exceptions, a contract can be created by writing, by word of mouth, by conduct or a combination of two or three of these methods. The exceptions are contracts regulated by the Law of Property Act (1925), the Marine Insurance Act (1906), the Consumer Credit Act (1974) and the Employment Protection (Consolidation) Act (1978), for which written documents are required.

theory: "If a person is liable on a contract because of what he promised or agreed to do, then insistence on writing may be a hindrance to giving effect to that person's intentions. To require writing is thus a paternalistic device which may protect people from the consequences of hasty or ill-thought-out promises or agreements". Legal theorists note that the anti-formality approach relies on confidence in the ability of the courts always to discover the truth, where the existence or the precise terms of a contract are challenged.

However, there is an increasing tendency to put agreements in writing for a number of reasons:

(i) Courts may not always be in a position to discover the truth and litigation incurs high costs. Written agreements reduce such costs because courts can apply the *parol evidence rule* whereby "evidence is not admissible to contradict or qualify a complete written contract" (Atiyah 1989). Hence it is generally held that a written document is exclusive evidence of the parties' intentions and the courts will not order that other terms be incorporated in it unless it is established that the written document departs from the intentions of *both* parties.

(ii) Written agreements reduce the risk of error. Courts may sometimes make mistakes but they are more likely to make mistakes about the existence of an oral agreement, rather than about a written agreement, again due to the *parol evidence rule* mentioned earlier. This is because although courts of law may have to exercise a fair amount of interpretation in order to establish the parties' intentions and the precise terms of a contract, a written document provides a more reliable frame of reference than the parties' (potentially different) accounts of their oral agreement.

(iii) Contracts are often entered into by large organizations and they are performed by the members of such organizations. Oral agreements would entail that everyone involved in aspects of contract performance be familiar and have the same understanding as everybody else of the terms of the agreement; that could be cost incurring, and prone to errors. Written agreements provide the common basis for co-ordinating everyone's actions.

Despite the strong legal presumption that signed contracts reflect accurately the intention of the parties written contracts do not necessarily contain explicitly *all* the contractual obligations assumed by the parties. Legal theorists draw a distinction between *express* and *implied* terms of an agreement.

2.2.4.1 Express and Implied Terms

Express terms are those terms negotiated, agreed and undertaken by the parties of their own free will. In written agreements such are the terms recorded in contractual documents. In case of litigation, the function of the court is to give effect to the terms that the parties have chosen to impose upon themselves. However, in practice the Law holds that a number of additional terms are implied in an agreement. This is largely due to concerns about fairness, justice and reasonableness. Parties anticipate performance rather than breach when forming agreements. Consequently, they do not always specify remedies for each and every possible breach that may arise. Generally it is difficult for parties to foresee all possible circumstances that may arise in the course of their transaction and consequently they do not specify in detail all of their obligations. So many contracting parties "simply agree on the bare essentials and leave everything else unexpressed" (Atiyah 1989).

Certain classes of contracts, such as sale of goods, agency, employment, insurance and so on, have a corresponding body of statutory legislation in the form of Acts of Parliament. If parties do not expressly contract outside such legislation (by including for example a statement to that effect in their agreement), then courts hold that the terms of such Acts are incorporated—or implied—in the agreement. In some cases parties cannot contract outside certain Acts, even if their intention is so and they state it expressly in their agreement. For example, parties cannot contract outside the Unfair Contract Terms Act; to be more precise, they are free to express their intention for the agreement not to be subject to it at the risk of the contract being deemed void.

One common case of terms being implied in an agreement without necessarily the parties' negotiation and subsequent consent (signalled by the parties' signing of relevant contractual documents) is standard-form contracts. Many contractual situations arise today not as a result of detailed negotiation but rather on the basis of pre-drafted standard forms that are offered by one party to another on a "take it or leave it" basis (cf. Atiyah 1989). For example, the purchase of tickets from a rail company or airline, the purchase of electricity or gas from respective organizations, or the undertaking of a mortgage or adoption of a credit card with banks and building societies, or the issuing of a software licence, are all governed by standard-form contracts. These are sometimes not offered explicitly by the issuing party to the consumer for inspection and signature, let alone negotiation. As Atiyah (1989) points out such standard-form contracts are natural in this age of mass production and they offer considerable benefits as they save time, effort and expense that individual detailed negotiation of terms would require. They can also facilitate the resolution of disputes in case of litigation, as a precedent ruling for one case offers guidance for future cases. However these are typically drafted by one party and thus represent only that party's intentions; sometimes they are drafted by neither party but rather a professional body, such as the Institution of Electrical Engineers, the Institution of Civil Engineers, or the Royal Institute of British Architects. One implication of standard-form contracts is that at least one of the contracting parties has restricted choice (the choice of either 'taking it' or 'leaving it'). Where the standard terms are drafted by one party they invariably tend to be favourable for it. As Atiyah (1989) notes "[o]ne extremely common and troublesome feature of standard-form contracts was the presence of an 'exemption clause', which often provided that the organization was not to be liable in virtually any circumstances whatsoever". Moreover, anyone that has entered such standard agreements with banks or building societies has experience of clauses that stipulate that the organization reserves the right to alter the terms of the agreement without prior consultation with the consumer.

In addition to implied terms as a result of Acts of Parliament or standard-form contracts it is sometimes the case that additional terms are deemed as belonging to an agreement through the Law of Tort and the Law of Restitution which are other parts of the law of obligations.

The implication of such distinction between express and implied terms is important in the context of this research: The primary source of information about the contents of contracts has been contractual documents, which potentially include terms not explicitly stated in them. As we shall see in section 2.4 many engineering contracts are based on standard-forms and may refer to Acts that regulate certain aspects of them. Any attempt to represent the contents of contracts in their entirety would need to represent relevant legislation (Acts, aspects of the Law of Tort and the Law of Restitution and so on), where it is appropriate.

2.3 Hohfeld's Fundamental Legal Notions

In 1913 Hohfeld noted that "[o]ne of the greatest hindrances to the clear understanding, the incisive statement, and the true solution of legal problems frequently arises from the express or tacit assumptions that all legal relations may be reduced to "rights" and "duties", and that these latter categories are therefore adequate for the purpose of analyzing even the most complex legal interests […]". He identified eight legal concepts as "the common denominators of the law" and suggested that these were essential in characterising unambiguously legal relations. Hohfeld's proposal is important in two respects, from the point of view of this research:

(i) Legal contracts are about defining legal relations and the common perception of their contents is that they are about duties and rights. Indeed Contract Law itself is characterised as the law of obligations. But there are other notions, besides that of 'obligation' that play a central part in them, such as the notion of 'power' (the ability to create legal relations by certain conduct), which were hinted at by the foregone discussion in section 2.2. It is therefore both interesting and useful to provide Hohfeld's explication of such notions, as a system of reference for their use in this dissertation.

(ii) These are notions that have been studied in legal theory (for example, (Kanger 1985), (Kanger & Kanger 1966), (Lindahl 1977)) and have provided the basis for much of the work of Artificial Intelligence researchers interested in modelling and developing applications for Law that employ such concepts (for example (Allen & Saxon 1993), (Jones & Sergot 1992; 1993; 1996), (Jones 1990)). Their work is part of the context for this research and to understand it we need to trace it back to Hohfeldian concepts.

Hohfeld's eight fundamental legal concepts were organised in opposite and correlative relations as illustrated in Figure 2–2. The exact choice of terms for the concepts varies but the figure illustrates one commonly referred to set.

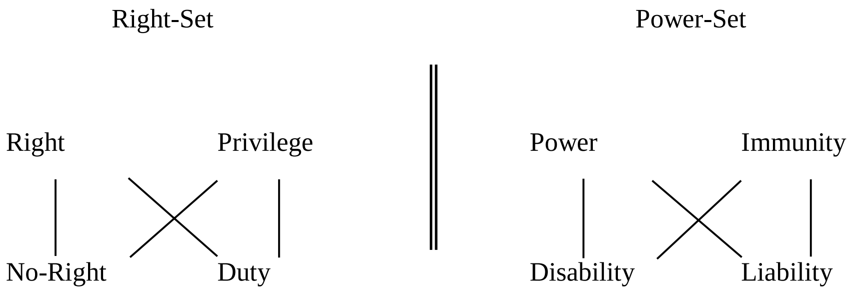


Figure 2–2 Hohfeld's Fundamental Legal Conceptions

Concepts appearing on the left belong to the right-set, whereas concepts on the right belong to the power-set. Concepts in the right-set are about rights, obligations and privileges that pertain between two parties. Concepts in the power-set on the other hand concern a person's ability to change the legal relations that obtain between himself and other people or between other people. For example, by entering contractual relations parties create legal relations between them, by virtue of which they have duties and rights with respect to one another.

Within each set, concepts appearing in the same column are, according to Hohfeld, jural opposites and concepts appearing along the same diagonal are jural correlatives. Hohfeld noted that concepts along the same row of both sets are often conflated: one often uses the terms 'right', 'privilege', 'power', 'immunity' interchangeably, and this also happens for terms such as 'duty', 'disability' and 'liability'. Each of those terms is also used interchangeably with other synonyms; for example, 'liberty', 'freedom' and 'permission' are frequently used in place of 'right' or 'privilege', 'exemption' in place of 'immunity', 'competence' and 'authority' in place of 'power', 'obligation' and 'responsibility' in place of 'duty' or 'liability' and so on. The term 'no-right' is not actually in the vocabulary one finds in legal (or even normal, everyday) discourse. Instead, it was coined by Hohfeld to denote 'absence of right'.

Hohfeld provided explications for these notions in his (1913) account through a series of examples of legal decisions that referred to them and example scenarios that illustrate the legal relations that they denote. His account can be summarised using formal notation and is discussed in what follows. The main example he used to illustrate his concepts is here summarised as follows:

> Consider two agents p_1 and p_2 where p_1 owns some land. If p_1 has a right towards p_2 that the latter shall stay off the former's land, then the correlative (or equivalent) is that p_2 has a duty towards p_1 to stay off his land. As regards p_1 himself, he has the privilege of entering his own land; in other words, p_1 does not have a duty towards p_2 to stay off his own land. Equivalently p_2 has no right towards p_1 to the effect that p_1 stays off his own land. The owner of the land, p_1, has the power to extinguish his own legal interest and to change that of p_2 if he sells his land to him. Then through this action of his, p_1 no longer has a right towards p_2 that the latter shall stay off the land, p_2 no longer has a duty towards p_1 to stay off the land, p_1 may no longer have the privilege to enter the land and p_2 may now have the right towards p_1 that the latter shall stay off the land and the privilege to enter it as it is now his. By selling his land, p_1 creates a new state of legal affairs to which p_2 is liable. Agent p_1 has certain immunities towards p_2; for example p_2 is under a disability (i.e. a lack of power) to take on the legal interest of the land formerly belonging to p_1 without p_1's action.

2.3.1 Legal Concepts of the Right-Set

Lindahl (1977) offers a formalisation of concepts in the right-set. The one presented here is based on Lindahl's but the content of each legal relation has been modified to denote explicitly the acting agent, following (Pörn 1977) and (Jones & Sergot 1993).

Let P be a set of legal persons or agents $\{p_i\}$ and A be a set of state of affairs $\{\alpha_j\}$. Expressions of the form $E_{p_i}\alpha_j$ denote that agent p_i sees to it (or brings it about) that α_j is the case. If A refers to actions rather than states of affairs, then $E_{p_i}\alpha_j$ denotes that agent p_i does action α_j. Expressions of the form $E_{p_i}\alpha_j$ are in other words agent-act descriptions of a set E. The set of legal relations between two legal persons is defined as $L \subseteq P \times P \times E$. As we are interested in legal relations between distinct legal persons[12], it is more accurate to say that $L \subseteq B \times C \times E$, where $B \subseteq P$ denotes the source of the relation (or the bearer) and $C \subseteq P$ denotes the destination of the relation (or the beneficiary). Relations of the right-set are then the members of L. The following opposite and correlative relations hold between them, reconstructed from Hohfeld's example:

(H1) $right(p_1, p_2, E_{p_2}\alpha) \equiv duty(p_2, p_1, E_{p_2}\alpha)$

(H2) $right(p_1, p_2, E_{p_2}\alpha) \equiv \neg no - right(p_1, p_2, E_{p_2}\alpha)$

(H3) $privilege(p_1, p_2, \neg E_{p_1}\alpha) \equiv no - right(p_2, p_1, E_{p_1}\alpha)$

(H4) $privilege(p_1, p_2, \neg E_{p_1}\alpha) \equiv \neg duty(p_1, p_2, E_{p_1}\alpha)$

Re-casting them in terms of Hohfeld's example, where $E_{p_i}\alpha$ denotes that agent p_i sees to it that he does not enter p_1's land, (H1) states that p_2 has a duty towards p_1 to refrain from entering his land, which corresponds to a right on the part of p_1 that p_2 refrains from entering his land. In similar manner, (H2) states that this right of p_1's is equivalent to a non-absence of this right on his part; (H3) states that p_1 has a privilege of not refraining from entering his own land, which corresponds to an absence of right on p_2's part to expect p_1 to refrain from entering his land; and (H4) states that p_1's privilege of not refraining from entering his land is equivalent to his lack of a duty to refrain from entering his land.

[12] There seems to be an implicit assumption on Hohfeld's part that legal relations are not reflexive. This is in accordance with our intuitive understanding of such concepts, as for example it is meaningless to speak of one's legal rights towards oneself, or one's legal privileges towards oneself and so on. There are of course linguistic expressions that refer to claims made from oneself, such as "I owe it to myself to complete my doctoral dissertation", but these are not referring to *legal* rights or duties. Perhaps it is useful to distinguish between legal relations between individuals and legal relations between roles. A legal relation can then be reflexive at the level of individuals but irreflexive at the level of roles.

2.3.2 Legal concepts of the Power-Set

In providing explications for the notions of ‘power’, ‘immunity’, ‘disability’ and ‘liability’, Hohfeld refers to the notion of *change* in given legal relations between two parties. He notes that a change in a given legal relation is brought about by either some facts beyond the volitional control of an agent (or both agents) or some facts which are under the volitional control of one or both agents. In the latter case “the person (or persons) whose volitional control is paramount may be said to have the legal power to effect the particular change of legal relations that is involved in the problem”. There are therefore three important dimensions in the notion of ‘power’: The agent (or agents) who have it; the action that the agent takes to manifest and exercise his power; and the transformation of the legal relations that is effected. This transformation may entail the extinction of existing rights and the creation of new ones, that is, the transformation affects relations of the right-set. However, it may also entail the extinction or creation of powers, disabilities, immunities and liabilities; that is, the transformation may affect relations of the power-set.

Let P, A, E and L be defined as in section 2.3.1. Let T be the set of power, immunity, disability and liability relations that may obtain between two parties and whose content rather than being an agent-act description is the transformation of legal relation. Then $T \subseteq P \times (P^n - \{\varnothing\}) \times E \times R^n \times R^n$, where the superscript n denotes powersets (in the mathematical sense), and the set R contains both relations from the right-set and relations from the power-set. For example

$$power(p_1,\{p_j\},E_{p_1}\alpha,\{duty(p_2,p_1,E_{p_2}\beta), privilege(p_1,p_2,\neg E_{p_1}\beta)\},\{privilege(p_2,p_1,\neg E_{p_2}\beta)\})$$

denotes that agent p_1 has power towards himself and agent p_2 (denoted by the set $\{p_j\}$), through performing action α to change the legal state of affairs from being one where p_2 has a duty towards p_1 to bring it about that β and p_1 has a privilege not to bring it about that β, to one where p_2 has a privilege not to bring it about that β. Restating the expression in terms of Hohfeld’s example, the former state of legal relations is one where p_2 has a duty to stay off p_1’s land while p_1 has the privilege not to stay off his own land, while the latter state is one where p_2 has the privilege not to stay off the land; the transformation is effected through p_1’s selling his land (denoted by α). The set of agents $\{p_j\}$ essentially comprises all agents whose legal relations are affected by the empowered agent’s actions. If it is desirable to have explicit relations between any two agents at a time, then expressions such as the above can be replaced by the equivalent conjunction of relations between pairs of agents.

Relations of the power-set are members of T. The following opposite and correlative relations hold between them, reconstructed from Hohfeld's example:

(H7) $power(p_1,\{p_2\},E_{p_1}\alpha,S_1,S_2) \equiv \neg disability(p_1,\{p_2\},E_{p_1}\alpha,S_1,S_2)$

(H8) $power(p_1,\{p_2\},E_{p_1}\alpha,S_1,S_2) \equiv liability(p_2,\{p_1\},E_{p_1}\alpha,S_1,S_2)$

(H9) $immunity(p_2,\{p_1\},E_{p_1}\alpha,S_1,S_2) \equiv disability(p_1,\{p_2\},E_{p_1}\alpha,S_1,S_2)$

(H10) $immunity(p_2,\{p_1\},E_{p_1}\alpha,S_1,S_2) \equiv \neg liability(p_2,\{p_1\},E_{p_1}\alpha,S_1,S_2)$

Recasting them in terms of Hohfeld's examples, (H7) states that p_1's power to transform legal states of affairs denoted by S_1 into S_2 by virtue of bringing about α is equivalent to a lack of disability on his part to effect such change. Similarly (H8) states that p_1's power to effect a change in legal affairs is equivalent to p_2 being liable to this change and thus to the new state of affairs created. By (H9) and (H10) it follows that lack of power, disability in other terms, on p_1's part to effect a change in legal affairs means that p_2 is immune (and hence not liable) to the change. Lindahl (1977) notes that Hohfeld's explication of these legal relations can take a different form, depending on how we interpret their corresponding linguistic expressions. Allen & Saxon (1993) in their formalization of relations in the power-set regard them as relationships between a legal person and a legal relation, rather than relationships between legal persons. chapter 3 revisits such considerations in more detail.

2.4 Engineering Contracts and Intended Applications

This section discusses the contents of engineering contracts in more detail along with the representational requirements that they give rise to for a number of potentially useful applications.

As mentioned earlier in this chapter, in common usage the term ‘contract’ refers both to a legally binding *agreement* between (usually) two parties and to the *document* that records such an agreement, if it is put in writing. In this dissertation the terms *agreement* and *document* are used when such a distinction needs to be made explicit, and they should be understood as referring to *contractual* agreement and document respectively. Where no distinction needs to be made the term ‘contract’ is used.

The common perception of contractual activity is that it can roughly be regarded as comprising two phases: Contract *formation*, where the parties involved specify their requirements of each other, negotiate on the various aspects of the exchange which will take place and come to some agreement. And contract *performance*, where the agreement is in force and the business exchange between the parties actually takes place. Consequently, there are two broad classes of electronic tools that one could consider, one for each phase of contractual activity. Contract formation tools would support *a priori specification* of contracts, whereas contract performance tools would support *a posteriori verification* of contracts (in (Santos & Carmo 1993) terms)

Contract formation tools include those that

(i) determine whether a given agreement is legally binding (whether a legally valid offer and acceptance exist);

(ii) enable parties to specify their requirements and check whether these are compatible or suggest adjustments in order to make them so (one could choose to call these *negotiation* tools); and, in the case of written agreements

(iii) assist drafters in putting the final product of the negotiation, the document, in written form (one could choose to call these *drafting* tools).

Contract performance tools are those that, given a specific agreement

(i) advise parties about their behaviour during the business exchange, reminding them of what needs to be done and when (one could call these *diary* tools);

(ii) monitor the parties’ compliance with the agreement and, in the case of violations, suggest available remedies or advise on the possible consequences.

Not all of these tools are useful for all kinds of contracts. According to legal theorists a contractual situation arises when (usually) two parties enter voluntarily into an agreement, assuming obligations towards each other, for the purpose of exchanging some product or service for a (usually) financial reward (cf. (Atiyah 1989); (Stone 1994)). Hence contractual situations can be identified in business exchanges ranging from the relatively straightforward (the purchase of a ticket for a bus journey, a simple Sale of Goods, standardised tenancy agreements) to the complex (the establishment of a long-term trading agreement between organisations or a complex trading procedure involving third parties). For contracts at the simple end of this scale, electronic support is likely to be unwanted. Where contracts are based on standard terms and conditions parties form and execute them without any apparent difficulty, when it comes to monitoring compliance the question is often whether goods were delivered on time and whether the required payment was made. Therefore, projects such as ALDUS (ALDUS 1992) are right in concluding that contractual activity is hardly in need of electronic support—*insofar as* one focuses on contracts at the simplest end of the scale.

The research reported here however has been addressing the representation of contracts at the other end of the scale, with particular attention to contracts that govern long-term exchanges between parties. Such agreements are frequently encountered in engineering. Most of the sample documents used as experimental material were provided through a collaborative project supported by British Gas. They concern the supply of natural gas from hydrocarbon field owners. The sample documents that were examined run to 200-300 pages each and are often accompanied by drawings, and various technical appendices. These are not one-off exchanges. The ‘life span’ of an agreement can be up to 20-25 years with a review every five years or so. Consequently, the contracts cover a large number of aspects. Some are typical of contracts in general—such as a specification of the period of the agreement, delivery quantities, prices, billing and payment arrangements, and so on. Some are *particular* to engineering contracts—such as provision of technical services, arrangements for technical reviews, the appointment of experts, the arbitration of disputes, the resolution of *force majeure* claims, insurance arrangements, warranties, indemnities, and so on. Some are specific to the particular kind of engineering—extraction of natural gas from a hydrocarbon field cannot be turned on and off at will, so there are many complex provisions dealing with shutdown procedures, adjustments for over- and under-delivery, monitoring of quality and so on.

The size and complexity of contracts in these and other areas highlights the need for electronic support in all aspects of contractual activity. As regards contract formation, the process of negotiating and establishing a new agreement is long and requires careful preparation. It is typically undertaken by a team rather than one individual (and this raises problems of co-authorship and co-ordination). The associated costs often account for a significant proportion of the cost of the project as a whole. Though there is no formal requirement in English Law for an agreement to be put in writing in order for it to be legally binding (cf. (Atiyah 1989); (Stone 1994)), complex business and engineering agreements *are* recorded in written form. It is essential to provide an agreed point of reference, especially where there are large amounts of detail or where the agreement is to remain in force for a considerable length of time, during which adjustments need to be made (for example to pricing information because of inflation or to other details if some of the circumstances of the organisations involved change). During the negotiation, several drafts of the contractual documents are produced. This is because contractual documents are sensitive and an omission or a mistake might have significant financial and legal consequences; moreover as the agreement covers a multitude of interrelated aspects, changes in some parts often propagate to other parts of the document. There is a need both for negotiation tools and for drafting tools.

It is often the case that any large organisation will have a large number of related agreements ongoing at any time. The associated documents are frequently consulted—some parts of them even on a daily basis—both by experienced and junior staff with varying degrees of understanding of the legal contents of the agreement. During the performance of the contract, *force majeure* procedures or even litigation are not unusual and the associated costs are very high. Tools that enable retrieval of contract content (both in terms of values for certain parameters and in terms of the legal implications of the agreement for the parties involved) are therefore desirable.

2.4.1 Contractual Content and Knowledge Representation Implications

This section discusses the contents of engineering contracts as these emerge from an examination of sample documents that was conducted as part of this research. In the discussion that follows, we consider only express contract terms. Implied contract terms that are specified in statutory legislation are also part of an agreement as mentioned in the previous section. In the sample documents that were examined such terms are incorporated typically by reference to the relevant Act. The documents that were examined reveal that a major part of them is dedicated to recording and explaining *stricto sensu* contractual duties of the parties. Many examples were found of promissory conditions, warranties and intermediate terms, though the latter were more frequent. In many cases the status of a clause as a promissory condition or warranty was signalled by a linguistic expression to that effect, or by explicit mention of the consequences of violation that enabled us to classify it, bearing in mind the criterion for the distinction held by Contract Law (though as mentioned in section 2.2.2, one should also bear in mind that parties may use terms inappropriately without full understanding of their legal meaning). We also encountered many examples of provisions that involve stipulations about time, typically associated with contractual duties and defining their temporal bounds. Apart from contract clauses that describe obligations (and consequently rights) of contracting parties written agreements contain clauses of a different nature. The purpose of presenting them is not to provide a formal classification of contractual clauses; rather it is to illustrate the different forms and functions that a clause may take in a given contract and consequently the different levels at which contractual content may require to be represented for different purposes. These different functions that contractual clauses may serve are:

(i) Provisions that define the meaning of terms that are to be used in a certain sense in the context of the agreement. These might include common terms that are not to be understood with their usual meaning; for example a "Year" may be intended to be understood as a time period not necessarily coincident with the common notion of a calendar year, or as a time period with specific bounds: An example of the latter is the following[13]:

> "Contract Year shall mean a period beginning at six o'clock am on the first day of October in any year and ending at the same hour of the first day of October in the next succeeding year".

In addition, certain terms that may refer to technical terminology specific to the domain of business that the transaction involves are also defined. An example of this type of definition is the following[13]:

> "Gross Calorific Value shall be computed "real" and shall mean that number of Megajoules produced by the complete combustion at a constant pressure of Atmospheric Pressure of one Cubic Metre of Natural Gas at a temperature of fifteen degrees Celsius with excess air at the same temperature and pressure when the products of combustion are cooled to fifteen degrees Celsius and when the water formed by combustion is condensed to the liquid state and the products of combustion contain the same total mass of water vapour as the Natural Gas and air before combustion".

It is also customary to define labels for entities that are referred to in the document in order to avoid repeating details. For example[14]:

> "Property shall mean in this agreement the property at the address shown in box A".

Finally, some terms are defined by reference to other provisions. For example[13]:

> "Annual Reconciliation Statement shall have the meaning specified in clause 9.4"[15].

[13] Taken from a sample contract on Gas Purchase.

[14] Taken from the author's own tenancy agreement.

Such definitions are agreed between the parties. Although they do not immediately concern the parties' conduct for the purpose of the transaction, their role is auxiliary: They facilitate understanding of the rest of the contractual provisions and may be referred to if the contract needs to be construed by a court of law in case litigation arises.

(ii) Provisions that describe procedures that must be followed in order for certain states of affairs to be achieved. For example[13] clause 19.2 of a sample contract specifies how contracting parties can go about having an expert appointed in order to resolve a dispute:

> "The Party wishing the appointment to be made shall give notice to that effect to the other party...The parties shall meet in an endeavour to agree upon a single Expert...If within 14 days from the service of the said notice the Parties have either failed to meet or failed to agree then the matter may forthwith be referred to the President of the Law Society of England and Wales, who shall be requested to select an Expert within 21 days... Upon the Expert being selected the Parties (or either Party) shall forthwith notify the Expert of his selection and...shall request him to confirm within 14 days whether or not he is willing and able to accept the appointment on the terms proposed....".

We shall return to the discussion of procedures later in this section.

(iii) Provisions that specify the conditions under which other provisions apply. Such conditions may apply to a variety of clauses: For example[16], the following clause determines when other prescriptive clauses are suspended:

> "If either party is prevented or delayed from or in performing any of his obligations under the Contract by Force Majeure, then he may notify the other party of the circumstances constituting the Force Majeure and of the obligations performance of which is thereby delayed or prevented, and the party giving the notice shall thereupon be excused the performance or punctual performance, as the case may be, of such obligation for so long as the circumstances of prevention or delay may continue".

This provision specifies conditions (namely *force majeure* circumstances and notification for them) that suspend any provisions that prescribe obligations for the party that is subject to *force majeure* circumstances. Similarly, other provisions may specify conditions under which price is to be calculated one way, and conditions under which it is to be calculated in a different way. One may regard such provisions as applicability conditions attached to obligations.

[15] Clause 9.4 contains a lengthy description of the contents of a statement.

[16] Model Form of General Conditions of Contract: Home or Overseas Contracts—with Erection. Institution of Electrical Engineers, 1988.

(iv) Provisions that specify how variant quantities are to be calculated, for example prices or quantity of goods, where the contract is not ‘fixed’ on these aspects[17], or various dates. An example of the latter is the following[13]:

> “The Start Date of this Agreement shall be the later of (a) the first day of October 199* and (b) completion of the Run-In Period”

One may regard such specifications as a special kind of definitions.

Contractual clauses such as (ii), (iii), or (iv) above are part of the agreement, in that they are themselves agreed by the parties. In the same way that definitions are useful in construing the contract, procedures, provisions that specify applicability conditions for other provisions and clauses that specify how variant quantities are to be calculated are useful for the contract to be performed. Of these procedures deserve further discussion for two reasons:

(i) In their specification they contain linguistic modalities that range over actions that are meant to be performed by the parties, often within specified time limits. Such modalities may be erroneously perceived to be of a deontic character, thus leading to erroneously conflating procedures with contractual duties *stricto sensu*, as explained below.

(ii) Their effect is that a state of affairs is achieved (in the example given here an expert becomes appointed if the procedure is carried out) and this may entail further legal relations for contracting parties. They are therefore associated with the notion of legal power, in the Hohfeldian sense, as mentioned in section 2.3.

As regards the first point, consider again the procedure for the appointment of an expert who is to resolve a dispute, where linguistic modalities have been underlined:

> “The Party wishing the appointment to be made shall give notice to that effect to the other party...The parties shall meet in an endeavour to agree upon a single Expert...If within 14 days from the service of the said notice the Parties have either failed to meet or failed to agree then the matter may forthwith be referred to the President of the Law Society of England and Wales, who shall be requested to select an Expert within 21 days... Upon the Expert being selected the Parties (or either Party) shall forthwith notify the Expert of his selection and...shall request him to confirm within 14 days whether or not he is willing and able to accept the appointment on the terms proposed....”.

[17] Cf. section 2.2.1 on engineering contracts and generally contracts that are relations.

Are the linguistic modalities in this procedure to be understood as obligations/rights of the parties? Some of them seem to be of a deontic nature, such as the first occurrence of “may” and the third occurrence of “shall”, but not all can be usefully seen as being of a deontic character. To illustrate this point consider the following: Is a party wishing the appointment of an expert obliged to give notice to that effect to the other party? At first sight it might appear that this is the case as words such as “shall”, “may”, “shall not” are often the hallmarks of deontic statements. However, it would not make sense to treat such modalities as having *deontic* standing. A party wishing to appoint an expert who does not serve a notice to the other party is not violating the Law or the contract. The violation of contractual obligations by one party, as we saw in section 2.2.2, entitles the other party to claim damages or to terminate the contract. A party wishing to appoint an expert who fails to serve notice to that effect to the other party is not liable to have the contract terminated or to pay damages; the only consequence that might be invoked is that no expert be appointed since the procedure was not initiated, and therefore the party’s goal will not be satisfied. Similarly, it does not seem to be (legally) obligatory for the parties to meet, despite the modality, for failure to do so does not yield a legal penalty, merely failure to promote the appointment of an expert.

Such modalities in procedures have therefore no deontic character. Rather the implication of their non-fulfilment is that progress towards a certain goal is not made. In this respect they are very much like operating instructions, for example of the sort “if you want to make an international call from the UK, you must first dial 00, then you must dial the country code, followed by the area code, but you must omit the ‘0’ of the area code, and finally you must dial the subscriber’s number”.

As regards the temporal bounds on such instructions, they are important in that not performing the specified action within the time limits may result in the action not being effective. Put alternatively, if certain actions are performed within the specified time limits they are effective in that they bring about some state of affairs. Perhaps one should take a stronger reading: Certain actions will be effective only if performed within the specified time limits (assuming no other clause allows an alternative procedure). In the example procedure above, it is perhaps implied that if the matter is forwarded to the president earlier than 14 days he may decide not to act on it (at all, or not to act before 14 days are completed and the parties renew their request to him); similarly failure on the part of the president to select an expert within 21 days may render the whole procedure ineffective and the party wishing such an appointment to be made may have to initiate it afresh; or it may not, in that the parties may accept the appointment of an expert who was selected by the President after the 21 days had elapsed. There are interesting points that can be raised in relation to the effect of a procedure that is associated with the exercise of power, such as, what the effect is of an ill-performed procedure. Austin’s theory of performative utterances (1962) offers some useful insights into such questions and is presented briefly in the next section.

As regards the second point, the effect of the procedure for the appointment of an expert is that such appointment is made once the procedure has been followed. However, further legal relations may be implied for the parties. For example, the intention might be that they are expected to comply with the expert’s ruling over the dispute. If such ruling entails obligations for one or both of the parties, then essentially following the procedure seems to commit the parties to assume such obligations. Such implied obligations should perhaps be regarded as part of the original agreement between the parties, since they agreed on the means (the procedures) by which they might come about, in the same manner as implied terms that arise as a result of legislation. In other words, the conduct of parties is sometimes treated by the Law as indicative of their undertaking of obligations[18]. The fact that parties have subscribed to some mechanism for appointing an expert may be construed as indicative of their undertaking of any obligations that might come about as a result of such expert’s ruling.

The list outlining the different forms that a contractual clause may take and the different functions it may serve motivates the ensuing discussion about the representation of contracts in general.

A contract can be regarded as a collection of different conceptual models that are interrelated, and various paradigms can be employed to represent them. At one level of abstraction, a contract is an organized collection of concepts. At another level, a contract is a collection of obligations, permissions, entitlements, powers and so on. At a third level, a contract can be regarded as a collection of procedures (protocols) that specify its operational aspects (how the business transaction is to be conducted in practice). These have temporal and action aspects that are at the core of much current research in Artificial Intelligence, Computer Science and Philosophical Logic. And from another standpoint still, a contract may be represented as a collection of parameters (the parties, the product in question, the price, the delivery quantities, the delivery date and so on). Contractual activities are not all concerned with all aspects of a contract. Each focuses on some particular parts. Alternative views of contracts need to be represented and sometimes integrated into a single system to support a variety of functions.

However, as mentioned earlier, information that is not contained in contractual documents is also required to support some aspects of contractual activity. What follows expands on this with reference to the tools outlined earlier.

Those tools that are intended to identify legally binding agreements need to establish whether a valid offer and acceptance exists. The Law (whether in England or elsewhere) specifies explicitly circumstances under which an agreement would *not* be legally binding (for example when it is formed under duress or when it involves unlawful activity, or when one of the parties is a minor) (cf. Atiyah 1989). Such tools need therefore to operate on a representation of the notions of ‘offer’, ‘acceptance’, ‘validity’, ‘duress’, ‘minor’ and ‘unlawful activity’, amongst others. This suggests that the general problem of identifying legally binding agreements is one of *classification*. Fundamental concepts must be represented and organized in a way that enables other concepts to be defined in terms of them or as instances of them. This is the kind of representation task that accounts for much of Artificial Intelligence and Law research over the past 20 years. Logic programs have been employed for this purpose (Sergot *et al.* 1986) as well as a variety of other formalisms (for an overview see (Sergot 1991)). Gardner's work (1987) for example concentrated on offer/acceptance representation using augmented transition networks and a special rule-based language. The central problem in the representation of classification norms is the treatment of open-textured concepts, that is, concepts whose meaning is not provided by a legislative definition but rather through example and decisions of courts of law (cf. (Sanders 1991); (Bench-Capon & Sergot 1988); (Gardner 1987); **(**McCarty, 1980)).

Tools that are intended to support contract negotiation, on the other hand, have very different representation requirements. One way to view contracts is as a collection of separate but interrelated sub-agreements. The parties involved have a common goal, to realise the business exchange, to co-operate, but each wants this to happen under the ‘best’ terms for them. What makes a particular arrangement good for a party is relevant to how it affects their broader business goals. Often the goals of the two parties are not mutually satisfiable as they stand, and revision (some mutual compromise) is required. A negotiation tool would therefore be useful if it allowed parties to specify their goals and determined whether these are satisfiable, or would be satisfiable, if certain terms were agreed; if resolution of some conflict were required, then it would be useful for the tool to indicate alternative terms (that entail change in

[18] Cf. section 2.2.1 especially contracts which are relations. See also section 2.4.2 for some useful insights on this matter from Austin’s doctrine of Infelicities.

the original set of goals). Obviously in its full generality, this is a huge problem raising a whole range of issues to which various techniques could be applied. This research has not addressed negotiation issues specifically but some of the discussion in chapter 5 is related to them.

Drafting tools raise similar concerns as negotiation tools, and this is no coincidence. The drafters' quest is for 'well-formed' documents, and good form entails, among other things, requirements of consistency and completeness—that contractual provisions are not contradictory and that they cover all cases that they are intended to cover. There are theoretical proposals for defining consistency and completeness for a set of norms (Alchourrón & Bulygin 1971) but application of these methods is not a practical possibility. It would require an exhaustive generation of all the possible factual circumstances (the 'Universe of Cases' in Alchourrón and Bulygin's terms) which is quite unrealistic except in some very special cases. The methods can be applied if we restrict attention to some very narrow, very specific part of the contract.

There is also the question of designing a set of norms, that is, deciding what obligations, permission, entitlements, rights and so on should be included in the contract. Some automated support for this question can also be provided. Sergot (1997; 1998b) presents a generalised and extended version of the Kanger-Lindahl theory of normative positions (Lindahl, 1977). This is a theory, which attempts to apply a combination of deontic logic (the logic of obligation and permission) and the logic of action/agency to the formalisation of Hohfeld's (1913) "fundamental legal conceptions": duty, right, and other complex normative concepts that were discussed in section 2.3. The generalised theory includes automated inference methods which have been implemented in computer programs intended to facilitate application of the theory to the analysis of practical problems, either for the purpose of interpretation and disambiguation of legal texts, or in the design and specification of a new set of norms. The objective is to clarify and expand an incomplete and imprecise statement of requirements into a precise formal specification at some desired level of detail. The role of the system is to guide this process, ensuring overall consistency and identifying any possibilities that remain to be explored. Chapter 3 gives a more detailed account of these issues.

It is difficult to demarcate precisely between the design of the agreement (and hence the object of negotiation tools) and the design of the document (and hence the object of drafting tools). Though the document records the agreement, it is often the case that what is being negotiated *is* the document itself, for example in terms of the wording used to express certain provisions. As contractual documents are taken to contain all that certain parties agreed by Law, it is important for the text to express as closely as possible what the parties' intended. Chapter 4 considers the drafting problem in detail and presents an approach that concentrates on the design of documents. Chapter 5 considers the detailed representation of contractual content.

Contract performance tools aim to advise parties on the effects of individual provisions, once an agreement is in force, to assist in planning the daily business exchange (in terms of what actions need to be performed and when) and to monitor the parties' compliance with the contract. We want to be able to extract parameter values, formulae for pricing or delivery times and detailed procedures as they apply in changing circumstances. We might want to monitor the parties' *compliance* with provisions of the contract. This is not a straightforward matter. It touches on fundamental problems in the field of deontic logic—contrary-to-duty obligations (for example (Prakken & Sergot 1996); (Prakken 1997)), the interplay between time and obligation (for example (van Eck 1982), the proper treatment of legal competence or power.

2.4.2 Austin's Doctrine of Infelicities

Part of Austin's work stemmed from the observation that in many cases "to say something is to do something", rather than merely asserting a truth-functional statement. For example the groom that says 'I do (take this woman to be my lawful wedded wife)' essentially *does* wed his bride, rather than merely asserting that he does. Austin called such operative utterances that bring about action *performatives* and, questioning the conditions under which they are successful, he developed his doctrine of *Infelicities*. Austin's account of performative utterances can inform the earlier discussion on contractual procedures (and indeed the discussion on contracts in general, since a whole contract can be regarded as one complex performative utterance made by both parties). It comprises some of the necessary conditions for the smooth or "happy" functioning of a performative and is reproduced below from his (1962):

> (A. 1) There must exist an accepted conventional procedure having a certain conventional effect, that procedure to include the uttering of certain words by certain persons in certain circumstances, and further,
> (A. 2) the particular persons and circumstances in a given case must be appropriate for the invocation of the particular procedure invoked.
> (B. 1) The procedure must be executed by all participants both correctly and
> (B. 2) completely.

> (Γ. 1) Where, as often, the procedure is designed for use by persons having certain thoughts or feelings, or for the inauguration of certain consequential conduct on the part of any participant, then a person participating in and so invoking the procedure must in fact have those thoughts and feelings, and the participants must intend so to conduct themselves, and further
>
> (Γ. 2) must actually so conduct themselves subsequently.

Austin notes that violating any of the six conditions results in the performative utterance being "unhappy", but in different ways. If conditions A and B are violated, that is, if the procedure is executed incorrectly, or if the person acting is not appropriate, then the act in question is not achieved. If on the other hand condition Γ is violated, say by the person acting being insincere or not conducting himself appropriately subsequently, then the act *is* achieved, albeit to achieve it in such circumstances is "an abuse of the procedure". Austin calls the first kind of infelicity, where through violation of A or B the act is not achieved, *misfires*, and the second kind of infelicity, where despite the violation of Γ the act is achieved, *abuses*.

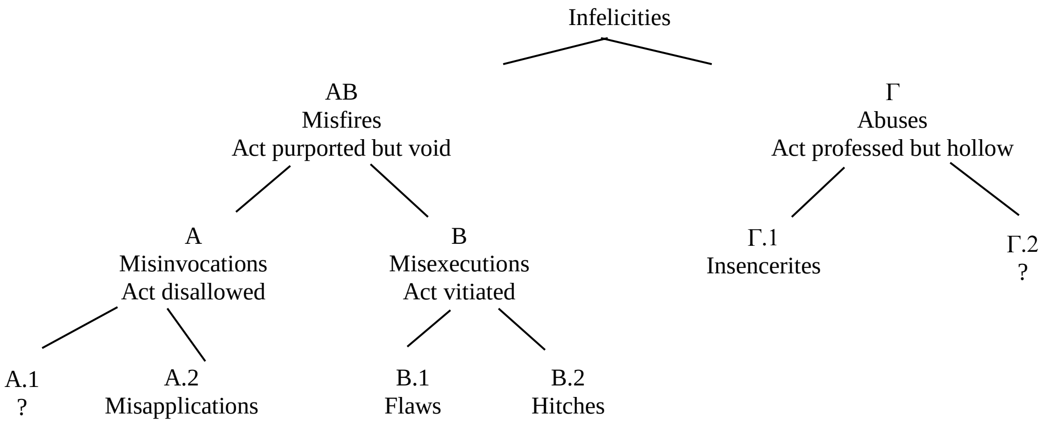


Figure 2–3 Austin's classification of Infelicities

Misinvocation of a procedure refers to the situations where there is no procedure to follow ((A.1), for which Austin provided no special name) or where the procedure in question cannot be made to apply in the way attempted because either the person invoking it is inappropriate or the circumstances in which it is invoked are inappropriate (A.2). *Misexecution* of a procedure refers to the situations where although the procedure is appropriate and is invoked by appropriate persons in appropriate circumstances, the purported act is vitiated by a flaw (B.1) or a hitch (B.2) in the conduct of the procedure.

Austin's account of performatives brings to mind the earlier discussion on contractual procedures in section 2.4.1. As mentioned then, contractual procedures are associated with the notion of legal power. It seems that Austin's "appropriate persons" in (A. 2) correspond directly to empowered agents invoking contractual procedures. Re-visiting the questions raised in section 2.4.1, Austin's account seems to confirm that modalities appearing in contractual procedures have no deontic character and departures from them do not yield legal penalties. Rather, in Austin's terms, they yield misfires or abuses. Hence, recalling that the example procedure concerned the appointment of an expert, if the party wishing the appointment does not serve notice to the other party, then the procedure is flawed; if it serves notice to the other party but not within the time limits specified (B.2) is violated, that is, there is a hitch in the execution of the procedure. If following correct execution of the procedure the parties do not intend to accept the appointed expert's ruling and conduct themselves accordingly subsequently (Γ), then the procedure has been abused.

As regards contracts in general, Austin's account agrees with the classification presented in section 2.2.1. Void contracts for example, (formed either in inappropriate circumstances or by inappropriate persons) seem to conform to what Austin calls misfires, whereas illegal or voidable contracts seem to correspond to what he calls abuses.

2.5 Conclusions

This chapter focused on discussing the contents of contracts at a general level (that is, independently of any specific agreement). Essentially contracts are understood as undertakings that parties assume of their own free will. From the point of view of Contract Law such undertakings imply obligations imposed on parties to behave as they promise to do when forming an agreement. Promises may refer to the performance of specific actions or the attainment of specific states of affairs. However, they also refer to representations of states of facts. At a general level contractual obligations have an underlying structure that is common to other kinds of obligations as well, and are entities with a life cycle: They are imposed on a subject, they are owed to someone specifically and they concern the performance of an action or the attainment of a state of affairs; they are initiated, possibly as a result of certain conditions being satisfied, they persist over some time and they become extinguished either when fulfilled or when their temporal bounds are exhausted. In the latter case, if they are not fulfilled they give rise to penalties for the party that did not satisfy them (or alternatively remedies for the party that suffered as a result of such violation). Such penalties depend on the importance attached to the obligation by the parties, or by a court of law if parties have not specified it, and effectively create new legal relations between parties: obligations to pay damages, or the right to terminate an agreement for example.

As the Law holds that written agreements contain the terms on which parties contract, sample contractual documents can be used as an additional source in determining the contents of contracts. Our examination of such documents showed that they contain provisions that do not directly concern obligations. These provisions concern agreed means of achieving certain goals (procedures and calculation of variant quantities), agreed meaning for certain terms (definitions), and agreed conditions under which other provisions apply or are suspended. Representing therefore the contents of contracts entails representing obligations (and their associated temporal bounds, conditions, sanctions), but also representing other information such as procedures, definitions and so on. Apart from obligations agreed between contracting parties, additional obligations are imposed on them by Law if the transaction they are involved in is regulated by statutory legislation and they do not contract outside it. Hence, to attempt a complete representation of the contents of contracts such statutory legislation needs also to be included. This chapter discussed the representation requirements that are raised by different kinds of tools aimed at supporting contractual activity.

The research reported here focuses on engineering contracts but it is worthwhile to mention that the results of the study apply to contracts in other areas as well (though perhaps with limited practical benefits). With an increasing number of electronic quasi-contractual transactions (cf. various issues of the *EDI Law Review*) taking place between organizations, in lieu of traditional exchanges in person, by representatives or by exchange of forms and notices, there is a growing demand for tools that analyse and represent contracts even if those concern exchanges at the simple end of the scale of contractual relations, such as the sale of goods. Recent work in the field of electronic commerce, for instance, focuses on formal (and hence machine processable) ways in which to represent trade procedures (cf. (Bons *et al.* 1995)) and on the development of a formal language for business communication that enables pre-contractual messages exchanged between two parties to be analysed so as to determine the contents of their agreement (cf. Kimbrough 1998)). Chapter 5 returns to these issues.

The next chapter provides the background to this research from the Artificial Intelligence perspective, in order to establish what available techniques can be applied to the representation of legal contracts.

3

Background Techniques and Theories

3.1 Introduction

This chapter reviews existing Artificial Intelligence techniques, which have been applied to legislation in general, for purposes of analysis and representation. Some authors trace the beginning of research into Artificial Intelligence and Law to Buchanan & Headrick (1970), while others trace the application of Computer Science, in general, to Law to Mehl (1958). In his (1958), Mehl described two kinds of "law machines", the *documentary* and the *consultation* machine. The former could be useful for retrieving relevant legal texts; the latter could be useful for providing legal advice. In their (1970) Buchanan & Headrick identified four processes used by lawyers to solve legal problems, namely the establishment of a goal and a method of measuring progress towards it, the recognition of relevant facts, the selection of rules to apply to facts and the discovery of analogies by generalisation of facts or of rules. Since then, a large amount of research has been conducted towards both kinds of machine identified by Mehl and their combination. Research towards documentary machines concentrated on information storage and retrieval, whereas research towards consultation machines focused on modelling legal reasoning. Much work was concerned with the interplay of both kinds of machines, that is, the deployment of retrieval in the consultation process and the use of reasoning to facilitate retrieval. Central issues in all this research were the development and application of appropriate representation formalisms for legislation and the development of appropriate inference methods.

Sergot (1991) provides a comprehensive review of projects that represent aspects of legislation and reason with such representations. His survey concentrates on what McCarty

(1983) called legal analysis and planning systems, as opposed to general legal practice automation systems and systems modelling the cognitive processes involved in legal reasoning. Legal analysis and planning systems match a representation of facts of a real or hypothetical case against a representation of relevant legislation and determine the legal concepts and relationships that hold, along with the legal consequences that follow. Susskind (1998) notes that legal analysis systems assist in inferring legal consequences from a given representation of facts and a given representation of relevant legislation, that is, they regard the law as an axiomatic system; hence one might infer that they are associated mostly with data-driven modes of reasoning. Planning systems on the other hand assist in establishing facts that would be appropriate in order for given legal consequences to hold, in a given representation of relevant legislation; hence one might infer that they are associated mostly with goal-driven modes of reasoning. Sergot (1991) argues convincingly that particular projects or approaches could be classified along several dimensions:

(i) By the aims and underlying assumptions of a project. Some projects aim to provide practical support for legislative activities. Others focus on generic techniques, such as logic programming or reasoning by analogy and use the law as a domain in which to experiment with them (e.g. Sergot (1985); (Rissland *et al.* 1984); (Rissland & Ashley 1987)). A third kind of projects seeks to explore the nature of legislation and the formalisation of modes of legal reasoning (e.g. (McCarty 1980; 1989); (McCarty & Sridharan 1980; 1981; 1982); (McCarty *et al.* 1979)).

(ii) By the type of activity that a project or approach seeks to support. Such activities include litigation and the resolution of legal disputes, the design and drafting of legislation or other legal documents, administrative tasks such as the processing of applications, and general advice provision (e.g. (Michaelsen 1984); (Capper & Susskind 1988)).

(iii) By the computational formalism employed by a given project. This includes the particular implementation language used in developing a system, but also the higher-level distinction between imperative and declarative representations, or algorithmic and non-algorithmic views of legislation (e.g. (Waterman & Peterson 1980); (Hellawell 1980; 1981; 1982), (Greenleaf *et al.* 1987); (Stamper 1980); (Stamper *et al.* 1982)).

(iv) By the nature of sources that are being represented in a given system. Some systems model the expertise of legal practitioners (e.g. (Waterman & Peterson 1980); (Capper & Susskind 1988)). Others represent legislation and amongst them, some represent statute law and other written rules and regulations, while others represent previous cases and a third kind attempts to combine both statute and case law (e.g. (Sergot *et al.* 1986); (Hafner 1987); (Gardner 1987)).

Sergot (1991) points out that no single classification dimension is appropriate. Categorising projects by aims and underlying assumptions seems to imply that a given project contributes towards one particular aim only. Yet approaches that originally aimed at experimenting with specific techniques produced practical applications and delved into the nature of legislation and legal reasoning (for example cf. (Sergot *et al.* 1986); (Bench-Capon *et al.* 1988)). Similarly, categorising projects by activity imposes an unnatural separation of techniques, for the principles and techniques of many projects that primarily aim at supporting one type of activity can be employed for another type of activity. As regards classification by computational formalism, Sergot (1991) points out that this can be misleading, as many approaches can be re-implemented in different computational formalisms. Finally, classifying projects by the nature of legislation that they represent seems to force an unnatural separation between statute law and case law. Though reasoning from previous cases is more pertinent in legal systems with a strong common law tradition, precedents play an important role in legal systems that are predominantly characterised by statute law. This is because determining whether a statute applies in a given case is usually a matter informed by previous cases, for the sake of consistency and fairness. The meaning of a statute sometimes calls for interpretation and reference to past decisions about it becomes an integral part of legal reasoning. Moreover, as Sergot (1991) puts it, such a distinction between systems that represent statutes and systems that represent case law can be very misleading, for "[i]t can give the false impression that there exists some essential difference between reasoning with statutes and reasoning with past cases. In particular, it can give the false impression that reasoning with statutes is mechanical and certain in its outcome, in a way that reasoning with cases is not".

The problem of classification of existing techniques arises in the context of this research as well. At the end of chapter 2, when discussing possible tools that could be developed to support contractual activity, we distinguished between contract formation and contract performance tools. It would therefore make sense to attempt a discussion of existing techniques, classifying them by activity, but this could be misleading because as we shall see in a moment, many could in principle be employed to support aspects of both contract formation and performance.

To my knowledge there is very little by way of past research concerned specifically with legal contracts. The most immediate precedent is perhaps Anne Gardner's (1987) work which used contractual offer and acceptance structures to illustrate her framework. Gardner's approach is prominent in the field of Artificial Intelligence and Law, and so it is reviewed briefly in what follows. A significant proportion of existing techniques employs Logic as their main representation formalism and inference mechanism. The relationship between Law and Logic has been explored in detail in recent years, both by Artificial Intelligence researchers (for example (Kowalski & Sergot 1990); (Sergot 1988)) and legal theorists (for example (Sartor 1998)). On the one hand, the Law is a rich and diverse field raising challenging issues which have stimulated Artificial Intelligence research, for example in knowledge representation, the analysis of natural language discourse, and the modelling of legal reasoning processes. On the other hand, logic as a tool for formalizing rational thinking has appealed to legal theorists concerned with improving legal reasoning and communication. The axiomatic view of the Law, first proposed by Leibniz, whereby a legal system is expressed by propositions from which legal conclusions are derived, has found many followers and has prompted much work on the formalization of legislation in logic and the simulation of legal reasoning by deduction. However, the axiomatic view of the Law and the analogy between legal reasoning and deduction has also been heavily opposed (cf. (Sartor 1998) for an overview). Nonetheless logic-based representations of legislation have a considerable following and they constitute much of the background for this research.

3.2 Contractual Offer and Acceptance

Gardner (1987) aimed to "create a model for the legal reasoning process that makes sense from both jurisprudential and AI perspectives". To this end, she proposed a system which not only aimed to solve legal problems, but also "to recognise the issues a problem raises and to distinguish between those it has enough information to resolve and those on which competent human judgements might differ". Gardner concentrated on contractual offer and acceptance, in order to illustrate her findings, an area which is both documented in legal sources and for which many precedents exist. Her framework, drawing ideas from natural language processing and frame-based representation, comprises four levels:

The top level holds knowledge about legal states, such as “a contract exists”, and events, such as “an offer is made”. These are represented by means of augmented transition networks. The second level holds definitional knowledge expressed by means of ‘if-then’ rules, relating conditions sufficient for legal events and legal states, such as the conditions under which an offer can be said to have been made. The third level holds examples that are and are not covered by the predicates used in the rules of the second level. The fourth level, which was left unimplemented at the time, was intended to hold knowledge sufficient to solve “questions that lawyers could reasonably disagree on”. Gardener calls the cases where experts genuinely disagree, “hard” and she articulates heuristics that could determine whether a given case is “hard” or “easy” (and consequently not solvable or solvable by the system).

The inferencing at the first two levels is primarily deductive in nature. The first level aims to classify a given situation, characterised in terms of events, as a legal state, by matching it against the augmented transition network for that legal state. If such classification cannot be made, the definitional rules of the second level are applied to determine whether a certain arc of the network at the first level can be followed. Where rules conflict or “run out”, typically due to open texture, level three is activated. Gardner developed a frame-like representation language in which to express problems for the system to solve. Her language was intended to accommodate actions, events, states, objects, speech acts and the notions of time and space. Appropriate representation formalisms and inference mechanisms for these have been sought by other researchers in the field. A natural starting point for these representations is symbolic logic.

3.3 Symbolic Logic

As early as 1957 and consistently through a series of papers (cf. (Allen 1968; 1978; 1980; 1982); (Allen & Saxon 1984)), Allen has been advocating the use of symbolic logic as a tool for analysing and interpreting legal text. Allen’s research programme concentrated on the use of logic to improve the language of the Law, by considering inadvertent ambiguity that arises in written legislative text. In his (1982) he claimed that a large amount of litigation based on written instruments could be traced to the drafter’s failure to convey his meaning clearly. Allen noted that ambiguity in written legislative text was of two kinds: intentional, when the drafter deliberately leaves the text open to various interpretations for political or social reasons; and unintentional, arising from imprecision which is inherent in natural language. He also stressed that ambiguity is often the result of what is written but also the result of what is omitted. Allen’s taxonomy of imprecision is summarised in Figure 3–1.

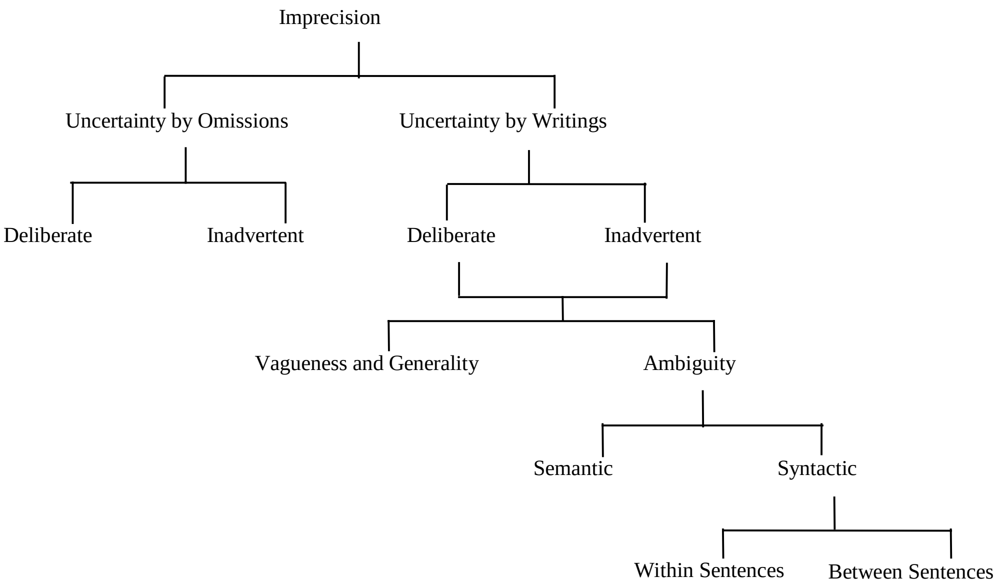


Figure 3–1 Allen's Taxonomy of Imprecision in Legal Text

Allen's early work concentrated on syntactic ambiguity that arises from relations between sentences. He proposed a normalised language in which the nuances of the meaning of terms (see Table 3–1) are reflected by the use of appropriately marked words.

Notation	**Meaning**
DEFINED TERM	Intended in the defined sense.
DEFINED TERM$_o$	Intended in the ordinary language sense.
DEFINED TERM$_a$	Intended to be ambiguous about whether occurrence is in defined sense or ordinary language sense.
defined term	(a) Drafter thought about problem, intended to be ambiguous about whether this occurrence is in defined sense or in ordinary language sense, but does not wish to tell audience about this deliberate choice. OR (b) Drafter did not think about the problem of whether this occurrence is in the defined sense or ordinary language sense.

Table 3-1 Allen's (1982) classification of defined terms

Using the conventions introduced for defined terms, Allen also provided definitions for logical connectives (AND, OR, NOT, IF…THEN, IF AND ONLY IF…THEN). His definitions for conjunction, disjunction and negation are the standard ones of propositional logic, but his definitions for the implication and the biconditional are different as Table 3–2 shows.

Connective	Meaning
if X then Y	True when both X and Y are true. False when X is true and Y is false. Sometimes true and sometimes false when X is false.
if and only if X then Y	True when both X and Y are true. False when either X or Y (but not both) are false. Sometimes true and sometimes false when both X and Y are false.

Table 3-2 Allen's definition of implication and biconditional

Based on the definitions for the primitive logical connectives, Allen provided definitions for other structural terms that relate sentences to each other, making explicit all the possible interpretations that could arise. Table 3–3 shows two interpretations for IF (one close to relevant implication, the other as a biconditional), two interpretations for ONLY IF and four (of "the more than 4000 senses") of UNLESS.

Connective	Notation	Meaning
IF	a IF$_1$ b	IF b THEN a
	a IF$_2$ b	IF b THEN a AND IF a THEN b
ONLY IF	a ONLY IF$_1$ b	IF NOT b THEN NOT a

	a ONLY IF$_2$ b	IF NOT b THEN NOT a
		AND
		IF NOT a THEN NOT b

UNLESS	a UNLESS$_1$ b	IF NOT b THEN a

	a UNLESS$_2$ b	IF NOT b THEN a
		AND
		IF b THEN NOT a

	X shall Y UNLESS$_3$ b	IF NOT b THEN X shall Y
		AND
		IF b THEN X shall NOT Y

	X shall Y UNLESS$_4$ b	IF NOT b THEN X shall Y
		AND
		IF b THEN IT IS NOT SO THAT X shall Y

Table 3-3 Some of Allen's definitions of logical connectives

Allen extended his language by producing definitions for statements and a diagrammatic notation ("arrow diagrams") to express the structure of legal norms. Table 3–4 summarises his definitions for statements. He then provided definitions for characterising a document as clearly or unclearly stated, clearly or unclearly worded, clearly or unclearly expressed, that he used to characterise well-drafted documents and ill-drafted ones.

In Allen's framework, all natural language statements are normalized ("systematically pulverized"). The process of normalization assists both in the interpretation of legislative provisions and the drafting of new ones. In the case of the former, this is because implicit relations between propositions become explicit by ascribing precise meaning to connectives and terms. In the case of the latter, this is because the drafter becomes aware of the different possible interpretations that might be ascribed to his text and has the option to remove inadvertent ambiguity by reformulating his text so that the only possible interpretation is the intended one. The following example illustrates the motivation for and the result of his proposal. It concerns the University of Michigan lease termination clause:

Statement Type	**Definition**
SINGULAR	It consists of just one constituent sentence.
CONJUNCTION	It consists of two constituent sentences joined by AND.
DISJUNCTION	It consists of two constituent sentences joined by OR.
CONDITIONAL	It consists of two constituent sentences joined by IF…THEN. (ANTECEDENT and CONSEQUENT of a CONDITIONAL have their usual meaning).
NORMATIVE	It includes some DEONTIC terminology.
DEONTIC TERMINOLOGY	Some form of the term 'obligated', some form of the term 'permitted', or some form of the term 'forbidden'.
RESULT	It is a NORMATIVE statement in the CONSEQUENT
CONDITIONAL NORMAL FORM	It is an IF…THEN statement, AND Its ANTECEDENT is either SINGULAR or a CONJUNCTION or a DISJUNCTION, AND Its CONSEQUENT is either SINGULAR, or a CONJUNCTION or in CONDITIONAL NORMAL FORM.
NORMALIZED	It is in CONDITIONAL NORMAL FORM AND Its CONSEQUENT is NORMATIVE (i.e. it is a RESULT) OR It is a CONJUNCTION of NORMALIZED statements.
LEGAL NORM	It is a SIMPLE LEGAL NORM or a COMPLEX LEGAL NORM.
SIMPLE LEGAL NORM	It is NORMALIZED AND Its ANTECEDENT is either SINGULAR or a CONJUNCTION AND Its RESULT is SINGULAR.
COMPLEX LEGAL NORM	It is a SINGLE-RESULT COMPLEX LEGAL NORM or a MULTIPLE-RESULT COMPLEX LEGAL NORM.
SINGLE-RESULT COMPLEX LEGAL NORM	It is a CONJUNCTION of SIMPLE LEGAL NORMS each of which has the same RESULT.
MULTIPLE-RESULT COMPLEX LEGAL NORM	It is a CONJUNCTION of SIMPLE LEGAL NORMS such that at least two of them have different RESULTS.

Table 3-4 Allen's definitions of statements

> The University may terminate this lease when the Lessee, having made application and executed this lease in advance of enrolment, is not eligible to enrol or fails to enrol in the University or leaves the University at any time prior to the expiration of this lease, or for violation of any provisions of this lease, or for violation of any University regulation relative to Resident Halls or for health reasons, by providing the student with written notice of this termination 30 days prior to the effective time of termination, unless life, limb, or property would be jeopardized, the Lessee engages in the sales or purchase of controlled substances in violation of federal, state or local law, or the Lessee is no longer enrolled as a student or the Lessee engages in the use or possession of firearms, explosives, inflammable liquids, fireworks, or other dangerous weapons within the building or turns in a false alarm in which cases a maximum of 24 hours notice would be sufficient. [Emphasis added for our present discussion].

The clause consists of a single sentence with the ambiguous form:

```
A if A1 and A2 or A3 or A4 or A5 or A6 or A7 unless B1 or B2 or B3 or B4 or
B5 in which cases B.
```

Depending on how this is bracketed different interpretations arise. Allen & Saxon (1984) identify approximately 80 clarification questions that may be required in order to disambiguate between all possible interpretations and as a result of such a process they conclude that the intended interpretation is:

```
((A IF ((A1 AND (A2 OR A3)) OR A4 OR A5 OR A6 OR A7))
IF NOT (B1 OR B2 OR B3 OR B4 OR B5))
AND (IF (B1 OR B2 OR B3 OR B4 OR B5) THEN B).
AND (IF NOT (B1 OR B2 OR B3 OR B4 OR B5) THEN NOT B
```

In this interpretation, ‘when’ is taken to mean IF, ‘unless’ is taken to mean IF NOT and ‘in which cases’ is taken to mean IF AND ONLY IF. Had both ‘when’ and ‘in which cases’ been taken to mean IF the resulting interpretation would be:

```
((A IF (A1 AND (A2 OR A3)) OR A4 OR A5 OR A6 OR A7))
IF NOT (B1 OR B2 OR B3 OR B4 OR B5)
AND IF (B1 OR B2 OR B3 OR B4 OR B5) THEN B)
```

Kowalski (1995) argues that logic programming representations are superior because they render many of the possible interpretations that might arise logically implausible. Perhaps it is more appropriate to say that logic programming representations yield better computational results precisely because the space of possible interpretations is smaller. But this is at the cost of expressive power, since ‘if’ and ‘only if’ are indistinguishable. Kowalski’s formulation of the University of Michigan lease termination clause is:

```
A if A1 and A2 and not B.
A if A1 and A3 and not B.
A if A4 and not B.
A if A5 and not B.
A if A6 and not B.
A if A7 and not B.
B if B1.
B if B2.
B if B3.
B if B4.
B if B5.
```

In this the condition `not (B1 or B2 or B3 or B4 or B5)` has been replaced by the simpler condition `not B` (under the assumption that `B1–B5` are the only conditions under which `B` holds). Kowalski (1995) points out however, that in logic programming there is no difference between the representation of ‘if’ and the representation of ‘if and only if’ as the closed world assumption operates and the logical consequences of a program include those of its completion. Therefore, the two interpretations produced by Allen & Saxon (1984) correspond to the same logic program. If we were interested in establishing the effects of a given set of provisions, the logic programming formulation above would account for two of Allen & Saxon's interpretations. Moreover, the logic programming representation can easily be turned into a natural language expression that is much clearer than the original clause, and more concise than the natural language expression corresponding to Allen & Saxon's representation. Given the fact that logic programs provide executable specifications, there does not seem to be any significant advantage to Allen & Saxon's approach over Logic Programming.

More recently, (Allen & Saxon 1993) the original framework was extended with explications of Hohfeld's (1913) "fundamental legal conceptions" to cater for normative statements. As we shall see in section 3.5.2 their explication of relations in the power-set is different from the one presented in the previous chapter, which was reconstructed from Hohfeld's examples.

Allen & Saxon's extended language (A-Hohfeld) for normalised legal drafting is the core of the MINT system in which a lawyer-user is asked to determine his intended interpretation of a given statement when presented by a set of alternatives produced by the system. It is not altogether clear whether the user's choices at one point of the consultation session progressively narrow the space of available interpretations, that is, whether the user's commitment to some interpretation for part of the legislative or regulatory text entails commitments to particular interpretations for other parts of the text, by restricting choices to those that would be consistent with previous ones. This is the kind of functionality offered by Sergot's (1998a; 1998b) automated analysis of normative positions, which is discussed in section 3.5.3.

3.4 Logic Programming

A detailed representation of contractual provisions could follow in the same steps of a popular approach for the representation of legislation or regulations, where provisions are represented as logic programs. The most celebrated instance of this approach is the formalisation of the British Nationality Act 1981 (Sergot *et al.* 1986). Social security regulations (Hammond 1983) and pension rules (Sergot *et al.* 1991) are also amongst the areas where the approach was tried successfully.

The following example, reproduced from Kowalski (1995) illustrates the approach on a subsection of the British Nationality Act 1981:

> 1(1) A person born in the United Kingdom after commencement shall be a British citizen
> if at the time of the birth his father or mother is—
> (a) A British citizen; or
> (b) settled in the United Kingdom.

This can be paraphrased in logic programming form as follows, where predicates are in infix notation and variables in uppercase:

```
X acquires british citizenship by section 1.1
if     X is born in the uk at time T
and    T is after commencement
and    Y is parent of X
and    Y is a british citizen at T
or     Y is settled in the uk at T
```

The main feature of such logic programming formulations is their close resemblance to the original legislative and regulatory text. Hence, the representation maintains a clarity that renders its inspection, verification, modification and extension easier to perform than on algorithmic representations.

Legislation or large sets of regulations typically contain heavy amounts of cross-references, with special cases or exceptions to rules appearing at different parts of a document or even in different documents. Constructing algorithmic representations is therefore a cumbersome task. Moreover, legislative and regulatory text may be ambiguous as it is expressed in natural language. Such ambiguity (as we saw earlier in Allen’s classification) may be semantic or syntactic. Sergot (1988) cites the example of regulations under which a woman could receive a ‘Housewives Non-Contributory Invalidity Pension’ (HNCIP), where one provision states that a woman would be entitled to HNCIP “if she is incapable of performing her normal household duties to a substantial extent”. The scope of “to a substantial extent” is not clearly defined (that is, it may refer to “performing” or to “incapable”). Moreover, the term “substantial extent” itself is vague as it might be perceived differently by different people. Open-textured concepts, that is, concepts whose meaning is not fixed, are abundant in legislative and regulatory text. An advantage of the logic programming approach is that multiple interpretations may be supported within the same representation as different logic programs.

The main benefit of logic programming representations of legislative or regulatory text is that they constitute executable specifications. Hence, drafters or policy-makers can ascertain whether the provisions they are creating have the desired effect or any unwanted side effects by ‘executing’ the model.

Having reviewed briefly the main features of the approach, let us now see to what extent it can be usefully applied to the representation of contractual provisions. As Sergot (1988) points out provisions that are essentially descriptive in nature admit logic programming formulation in a straightforward manner. That is, for provisions, which specify conditions under which an entity X is to be classified as being of type Y, we can construct corresponding logic programs with relative ease. Existing applications of logic programming to legislation and regulations show however that such an exercise is not useful unless the legislation or regulations that are represented are reasonably complex. Where such complexity is lacking questions of vagueness and open texture dominate. To illustrate this point consider the following extract from a sample contract (IEE 1988), which describes what circumstances are considered to be *force majeure*:

> (46) Force Majeure means:-
> —war, hostilities (whether war be declared or not), invasion, act of foreign enemies;
> —ionising radiations, or contamination by radio-activity from any nuclear fuel, or from nuclear waste from the combustion of nuclear fuel, radio-active toxic explosive, or other hazardous properties of any explosive nuclear assembly or nuclear component thereof;
> —pressure waves caused by aircraft of other aerial devices travelling at sonic or supersonic speeds;
> —rebellion, revolution, insurrection, military or usurped power or civil war;
> —riot, civil commotion or disorder;
> —any circumstances beyond the reasonable control of either of the parties.

It is not technically difficult to construct a logic program for classifying an incident as *force majeure*. Such a representation may take various syntactic forms, one of which is shown below:

```
fm_circumstance(war).
fm_circumstance(hostility).
fm_circumstance(invasion).
:
:
force_majeure(Episode) :-
     fm_circumstance(Episode).
force_majeure(Episode) :-
    beyond_parties_reasonable_control(Episode).
```

Such a representation, although possible, is not very useful however. If one were interested in identifying circumstances that are regarded as *force majeure*, the text of the contract itself is good enough to provide an overall idea[19]. Experience with logic programming applications to legislation and regulations suggests that the legislative text or rules that are represented must be sufficiently complex to warrant the effort involved in analysing them, interpreting them and representing them. The *force majeure* example above is not complicated enough to yield a useful logic program. The open-textured “reasonable control” is the most interesting of its contents and it would be useful if precedents existed that indicated what circumstances had counted as *force majeure* previously in the context of the agreement at hand, and also in relation to other contracts. Unfortunately, such precedents are not available for analysis and representation. As our industrial collaborators from British Gas indicated, when such circumstances arise they are resolved through negotiation and sometimes litigation on a context-dependent basis.

Logic programs can also be constructed for prescriptive provisions if we regard them as qualification norms, that is, as statements specifying legal relations for the parties in given conditions. For example, the following is a provision from the section on Contractor's Obligations (IEE 1988):

> (14.4) The Contractor shall not without the Engineer's consent make any material alteration to the approved Programme.

This might be regarded as defining an obligation for the contractor to refrain from performing a certain action in given conditions. A logic programming formulation of this clause might go along the lines:

```
The contractor is obliged by section 14.4 to not alter the programme P
if     there does not exist consent by the engineer to alter the programme P.
```

Or in prefix form, by choosing appropriate predicates:

```
obliged(section('14.4'), contractor, ~a):-
       not fact(consent(engineer, a)).
```

where '`a`' is the proposition corresponding to 'altering programme P'. Alternatively, taking an obligation not to do something as equivalent to a prohibition to do it, the formulation could be:

```
prohibited(section('14.4'), contractor, a):-
       not fact(consent(engineer, a)).
```

Consider also the following example from the same sample contract (IEE 1988):

> (14.1) Within the time stated in the Contract or, if no time is stated within 30 days after the Letter of Acceptance, the Contractor shall submit to the Engineer for his approval the Programme for the execution of the Works showing: -
> (a) the sequence and timing of the activities by which the Contractor proposes to carry out the Works (including design, manufacture, delivery to site, erection and testing),
> (b) the anticipated numbers of skilled and unskilled labour and supervisory staff required for the various activities when the Contractor is working on Site,

[19] Though, an examination of the text of that provision shows that any circumstance is pretty much covered, so long as it is deemed to be beyond the reasonable control of the parties, and what is "reasonable" control is open-textured.

(c) the respective times for submission by the Contractor of drawings and operating and maintenance instructions for the approval thereof by the Engineer,

(d) the times by which the Contractor requires the Purchaser

(i) to furnish any drawings or information,

(ii) to provide access to Site,

(iii) to have completed any necessary civil engineering or building work (including foundations for the Plant),

(iv) to have obtained any import licences, consents, wayleaves, and approvals necessary for the purposes of the Works,

(v) to provide electricity, water, gas, and air on the Site or any equipment, materials or services which are to be provided by the Purchaser.

This section specifies an obligation on the part of the contractor to submit a programme of a specified description to the engineer within a specified time. A logic programming representation of such an obligation might go along the lines:

```
The contractor is obliged by section (14.1) to submit an acceptable programme
P at time T
if     T is the time specified in the contract.

The contractor is obliged by section (14.1) to submit an acceptable programme
P at time T
if     there exist a letter of acceptance at time T1
and    T is later than T1 by 30 days.
```

The representation could be extended by an appropriate definition for what makes an acceptable programme according to section (14.1). This is certainly neither the only nor the best way to represent the prescriptions above. The purpose of the example is to illustrate a point made by Jones and Sergot (1992): Prescriptive statements *can* be regarded as definitional and be represented as logic programs if one wants to retrieve such information or to construct a system that operates as they specify. Where it is important to maintain the distinction between what *ought* to be the case and what actually *is* the case, however, such representations fail, as they do not accommodate the possibility that norms might in practice be violated. The representation of section (14.1) above, for instance, is useful if all we want to establish is what obligation the contractor has with respect to this section, what action on his part is characterised by this obligation and when such obligation is to be carried out. There are no means of establishing what follows from such an obligation. For example what happens if the contractor violates it and, relevant to that, what constitutes violation—the non-performance of the act within the specified time period, the non-conformance of the programme to the standards set out in the contract or both? Jones and Sergot (1992) conclude that in cases where we want to reason with the distinction between the ideal and the actual some form of deontic logic is required. Moreover, means of analysing sets of norms in terms of consistency and completeness are also required, if we are interested for instance in establishing whether a party bears conflicting obligations. Such conflict is perhaps apparent in cases where a party is obliged to do a and obliged to do ~a. The case of temporally qualified obligations is even more interesting as to detect conflict we also need to reason about time. Returning to the example above, can the same set of rules consistently state that the contractor is obliged to submit a programme before some time T and after the same time T? Chapter 5 discusses how formal techniques developed for the temporal verification of hardware specifications may be applied for the verification of temporally qualified norms.

3.5 Deontic Logic

As legislative and regulatory statements primarily aim to direct human behaviour by specifying permissible, obligatory or forbidden actions, deontic logic, a branch of modal logic (introduced by von Wright 1951) that is concerned with norms and normative behaviour, is a natural candidate for representing and reasoning with such statements. Deontic Logic finds its origins in Ethics and Legal Philosophy, but it has more recently found applications in computer science and Artificial Intelligence, for example as a means of specifying constraints or security policies (Meyer & Wieringa 1993). Since 1926, when Mally first tried to account formally for deontic notions, the field has been abundant with paradoxes that alternative systems seek to avoid. At the core of current developments in Deontic Logic lies von Wright's *standard system of deontic logic*, or a logic of type KD in Chellas's (1980) classification, whose axioms and rules are summarised below (from (Meyer & Wieringa 1993), where O stands for 'obligatory', P stands for 'permitted' and F stands for 'forbidden':

(KD0)	All (or enough) tautologies of Propositional Calculus	
(KD1)	$O(p \to q) \to (Op \to Oq)$	the so-called K-axiom
(KD2)	$Op \to Pp$	"Obligatory implies permitted", the so-called D axiom
(KD3)	$Pp \equiv \neg O \neg p$	"Permission is the dual of Obligation"
(KD4)	$Fp \equiv \neg Pp$	"Forbidden is not permitted"
(KD5)	$\dfrac{p, p \to q}{q}$	Modus Ponens
(KD6)	$\dfrac{p}{Op}$	O-necessitation

Table 3-5 The standard system of deontic logic

Chapter 2 introduced a logical formulation of the legal notions that Hohfeld (1913) regarded as fundamental in the analysis of any given legal problem. In what follows we revisit

Hohfeldian concepts, discuss the formulation in more detail and present some of the advances made by legal theorists and Artificial Intelligence researchers.

3.5.1 Hohfeldian Legal Relations of the Right-Set

As we saw in chapter 2, a possible formulation of Hohfeld's relations of the right-set is the following:

(H1) $right(p_1, p_2, E_{p_2}\alpha) \equiv duty(p_2, p_1, E_{p_2}\alpha)$

(H2) $right(p_1, p_2, E_{p_2}\alpha) \equiv \neg no - right(p_1, p_2, E_{p_2}\alpha)$

(H3) $privilege(p_1, p_2, \neg E_{p_1}\alpha) \equiv no - right(p_2, p_1, E_{p_1}\alpha)$

(H4) $privilege(p_1, p_2, \neg E_{p_1}\alpha) \equiv \neg duty(p_1, p_2, E_{p_1}\alpha)$

The first two arguments in this formulation are the bearer and counter-party of the relation respectively. The third argument is an agent-act component of the form $E_a X$ denoting "agent *a* brings about state of affairs *X*" or alternatively "agent *a* sees to it that *X* is the case". This is intended to be more general than "agent *a* performs action *X*" or "agent *a* does *X*", which was captured by Kangers' *Do* operator (Kanger & Kanger 1966). The agent-relativized action operator *E* captures therefore both direct and indirect action and can be used to express both specified actions and specified states of affairs (where the end is specified rather than the means to achieve it).

From a syntactic point of view, the parties of a legal relation remain the same in opposite relations and change roles in correlatives. Therefore, the person bearing a duty is the person on whom a claim is made through a 'right' relation. It can also be noted that in this formulation, when an agent bears a duty or a privilege towards another, that duty concerns his own action, whereas when an agent has a right or the absence of it towards another, such right or the lack of it concerns the other agent's action.

Another point is relevant to the agent-act description component of legal relations: Hohfeld makes the following remark about the content of legal privilege and legal duty: "some caution is necessary at this point; for, always, when it is said that a given privilege is the mere negation of *duty*, what is meant, of course, is a duty having a content or tenor precisely *opposite* to that of the privilege in question". Given an agent p_i and some state of affairs or action α there are four possible agent-act descriptions, captured by the general schema $\pm E_{p_i} \pm a$:

$$E_{p_i}\alpha \qquad E_{p_i}\neg\alpha \qquad \neg E_{p_i}\alpha \qquad \neg E_{p_i}\neg\alpha$$

The first refers to an agent performing some positive action; the second refers to an agent performing some negative action; the third refers to an agent refraining from performing a positive action and the fourth refers to an agent refraining from performing a negative action (what is positive or negative of course depends on how one fixes the interpretation of action α). One might regard the first as equivalent to the fourth and the second as equivalent to the third. However this is not always the case unless operating within some 'closed world', where not acting in a certain manner to achieve a specific state of affairs is taken as implicit action that achieves the opposite state of affairs. For example, if α stands from setting a switch to 'on' and $\neg\alpha$ stands for setting it to 'off', then saying that an agent does not set the switch to 'on' cannot be conflated with saying that an agent sets the switch to 'off', for the agent may simply be idle, that is, he may refrain from any action that has anything to do with the switch settings.

A final point is worth noting concerning relations in the right-set. Reconstructing jural opposite and correlative relations from Hohfeld's explication gave rise to (H1)–(H4) as we saw. By equivalence substitution, it is possible to derive relations between the pairs of relations that lie along the same row in the right-set:

(H5) $right(p_1, p_2, E_{p_2}\alpha) \equiv \neg privilege(p_2, p_1, \neg E_{p_2}\alpha)$

(H6) $duty(p_1, p_2, E_{p_1}\alpha) \equiv \neg no - right(p_2, p_1, E_{p_1}\alpha)$

Clauses (H5) and (H6) describe what one might call 'correlatively opposite' concepts and complete the relations amongst concepts of the right-set. 'Duty', 'privilege' and 'no-right' can now be all defined in terms of 'right':

(R1) $duty(p_1, p_2, E_{p_1}\alpha) \equiv right(p_2, p_1, E_{p_1}\alpha)$

(R2) $no - right(p_1, p_2, E_{p_2}\alpha) \equiv \neg right(p_1, p_2, E_{p_2}\alpha)$

(R3) $privilege(p_1, p_2, E_{p_1}\alpha) \equiv \neg right(p_2, p_{1,} \neg E_{p_1}\alpha)$

It should be noted that this is just one way of summarising Hohfeld's explication of legal relations of the right-set. Kanger & Kanger (1966) and following them Lindahl (1977) note that this schema of opposites and correlatives may be misleading as it does not appear to be entirely what Hohfeld had in mind. Hence, 'right' and 'no-right' are not opposites in the same sense that 'duty' and 'privilege' are, since in the latter case apart from the external negation, the agent-act component is negated as well.

Lindahl (1977) notes that this occurs in the case of correlative relations as well. Hence, 'right' and 'duty' are not correlative in the same sense that 'no-right' and 'privilege' are, since in the latter case there is an internal negation on the agent-act component. In his critique of such a formulation of Hohfeldian concepts, Lindahl makes further observations. He notes for instance that the interdefinability schema allows the following statement to be valid:

$$\neg(duty(p_1, p_2, E_{p_1}\alpha) \wedge duty(p_1, p_2, \neg E_{p_1}\alpha))$$

That is, it is not possible for an agent to have conflicting obligations towards another agent. Unless the set of norms that apply to two agents is consistent, it is however possible for such a situation to arise in practice. A conflict between two norms must be detectable before it can be resolved.

Lindahl also notes that this formulation of Hohfeldian concepts in the right-set, as legal relations between two parties does not account for linguistic expressions where no counter-party is specified, such as "X has a duty to perform action A", that is, what Austin called *absolute obligations*[20]. A third criticism raised by Lindahl is in relation to third parties. Lindahl notes that the interdefinability schema allows the following statement to be derived:

$$duty(p_1, p_2, \neg E_{p_1}\alpha) \wedge privilege(p_1, p_3, E_{p_1}\alpha)$$

This entails that the same agent has a duty towards another not to perform some action and at the same time a privilege towards a third party to perform the same action. Assuming that p_2 and p_3 are different agents, there does not seem to be any reason for such a statement to be regarded as problematic, for each of the conjunct relation applies in a different context.

Kanger provided a generalised theory of rights, based on Hohfeldian concepts, which was further extended by Lindahl (1977) and more recently by Sergot (1998b). Jones & Sergot (1992; 1993) have shown how the extended theory of normative positions can be applied to the disambiguation of norms in a practical context.

[20] Austin J. (1863). *Lectures in Jurisprudence*, vols II-III, London. Cited in Lindahl (1977).

3.5.2 Hohfeldian Legal Relations of the Power-Set

In chapter 2, we saw that a possible formulation of Hohfeld's relations of the power-set is the following:

(H7) $$\mathit{power}(p_1, p_2, E_{p_1}\alpha, S_1, S_2) \equiv \neg \mathit{disability}(p_1, p_2, E_{p_1}\alpha, S_1, S_2)$$

(H8) $$\mathit{power}(p_1, p_2, E_{p_1}\alpha, S_1, S_2) \equiv \mathit{liability}(p_2, p_1, E_{p_1}\alpha, S_1, S_2)$$

(H9) $$\mathit{immunity}(p_2, p_1, E_{p_1}\alpha, S_1, S_2) \equiv \mathit{disability}(p_1, p_2, E_{p_1}\alpha, S_1, S_2)$$

(H10) $$\mathit{immunity}(p_2, p_1, E_{p_1}\alpha, S_1, S_2) \equiv \neg \mathit{liability}(p_2, p_1, E_{p_1}\alpha, S_1, S_2)$$

In this formulation, relations of the power-set are regarded as relations between two parties. The first argument is the party bearing the relation and the second is the counter-party. The third argument describes what the empowered agent brings about in order to effect a change from legal relations contained in the set S_1 to legal relations of the set S_2. The sets S_1 and S_2 comprise legal relations of the right-set and the power-set.

Lindahl's formulation of Hohfeldian concepts of the power-set is different. Where F stands for the changed legal relation in expressions of the form "p has versus q power to the effect that legal relation F is created", Lindahl (1977) notes that there are two possible ways of expressing Hohfeld's opposite and correlative relations. One is the following:

(L1) $$\mathit{power}(p, q, F) \equiv \neg \mathit{disability}(p, q, F)$$

(L2) $$\mathit{power}(p, q, F) \equiv \mathit{liability}(q, p, F)$$

(L3) $$\mathit{immunity}(p, q, F) \equiv \mathit{disability}(q, p, F)$$

(L4) $$\mathit{immunity}(p, q, F) \equiv \neg \mathit{liability}(p, q, F)$$

This is close to (H7)–(H10). The difference is that the agent-act component that brings about a change in legal relations, as well as the previous state of legal relations are hidden in (L1)–(L4).

The other possibility for expressing Hohfeld's opposite and correlative relations as Lindahl notes is following Kanger (1971):

(K1) $power(p,q,F) \equiv \neg disability(p,q,\neg F)$

(K2) $power(p,q,F) \equiv liability(q,p,F)$

(K3) $immunity(p,q,F) \equiv disability(q,p,F)$

(K4) $immunity(p,q,F) \equiv \neg liability(p,q,\neg F)$

(K1) and (K4) are where the difference from the set (L1)–(L4) lies. As Lindahl notes the choice between schemata L and K depends on how they are read. L schemata correspond to statements of the form '…has versus…a…with respect to…' whereas K schemata correspond to statements of the form '…has versus…a…to the effect that…'.

As was the case for legal relations of the right-set, from a syntactic point of view, the bearer and counter-party of a relation remain the same for opposite concepts, whereas they change roles in correlative relations. Hence the person against whom an agent has power is liable to the change effected, and is not immune to this change. It is also worth noting that the action that brings about the change of legal affairs is that of the empowered (or not disabled) agent, who exercises such power. An agent may acquire power to change legal relations, through some other agent's exercise of power, however. For example, a head of department is empowered by signing off purchase forms for equipment to create a state of affairs whereby contractual relations are created between his department and the supplier of such equipment. By exercising his power to delegate, however, say by signing appropriate forms or issuing appropriate instructions, his secretary may acquire the power to sign off purchase forms for equipment (and so S_2 in (H7)–(H10) in the power relation borne by the head of department would contain the power relation borne by the secretary).

Another point to note is that an empowered agent may also find himself liable to the change in legal affairs that he effects through exercising his power. So returning to Hohfeld's example, when p_1 sells his own land to p_2 (because he has the power to do so) he creates a duty for himself to refrain from entering it.

Allen & Saxon (1993) mention that in their formalization of relations in the power-set these are taken to be relationships between a legal person and a legal relation, rather than relationships between legal persons. Hence in A-Hohfeld their definitions of relations in the power-set are:

(i) p_1 has POWER to create legal relation $R =_{def}$ legal relation R is NOT so, AND it is naturally possible for p_1 to do a_1, AND IF p_1 does a_1 THEN legal relation R is created.

(ii) Legal relation R has LIABILITY of being created by $p_1 =_{def} p_1$ has POWER to create legal relation R.

(iii) p_1 has DISABILITY to create legal relation $R =_{def} p_1$ lacks POWER to create legal relation R.

(iv) Legal relation R has IMMUNITY of being created by $p_1 =_{def} p_1$ lacks POWER to create legal relation R.

The first of these A-Hohfeld definitions merits some discussion. In their definition of power, Allen & Saxon include the agent's physical ability to perform the action a_1 that creates the legal relation R. Jones & Sergot (1996) argue persuasively against this formulation. Their point is that there is a distinction between legal power, the physical ability required in order to exercise it and permission to exercise it.

Makinson (1986, p. 409) presents the following example which illustrates the difference between power and permission to exercise power:

> "…consider the case of a priest of a certain religion who does not have permission, according to instructions issued by the ecclesiastical authorities, to marry two people, only one of whom is of that religion, unless they both promise to bring up the children in that religion. He may nevertheless have the *power* to marry the couple even in the absence of such a promise, in the sense that if he goes ahead and performs the ceremony, it still counts as a valid act of marriage under the rules of the same church even though the priest may be subject to reprimand or more severe penalty for having performed it."

Jones & Sergot (1996, p. 431) present the following example, following on from Makinson's, which illustrates the difference between physical ability and legal power:

> "… one may imagine circumstances in which it is not *practically possible* for the priest to marry the couple (because, say, he is sick or otherwise incapacitated), although he is still empowered to do so".

Jones & Sergot (1996) also present what they call the "transfer problem":

> "…an agent may have the practical possibility to see to it that some couple are married (by, e.g., getting an authorised priest to conduct the ceremony) without himself being empowered to marry them".

Therefore, according to Jones & Sergot an agent's physical (practical) ability to perform certain acts, which establish legal relations, is neither a necessary nor a sufficient condition for the agent to be empowered.

Another condition in Allen & Saxon's definition for power is that the legal relation *R* does not exist before the exercise of a specified action by the empowered agent. This also seems counter-intuitive, since if a particular legal relation *R* does exist, the empowered agent seems merely to lack reason for exercising his power, or in other words, he is still empowered but exercising his power is redundant as it does not achieve a change. For example, a priest's legal power to effect marriage does not become obsolete because two particular individuals happen to be married already. What the priest lacks in such circumstances is not power, but rather permission to exercise it. Similarly, when a Head of Department has power to enter into contractual relations with a computing company in order to purchase equipment (whereby, by signing a purchase form he creates legal relations between his Department and such company, in which the Department is obliged to pay for equipment and the Company is obliged to deliver it), such power is not removed once there is already a contractual relation with this company, say because of a previous purchase order. What the Head of Department may lack in such circumstances is opportunity or reason to exercise his power (say because the existing contractual arrangement covers the Department's needs for equipment).

The content of legal relations in the power-set has two dimensions:

(i) The action performed by an empowered agent that brings about a change in the state of legal affairs.

(ii) The actual transformation of a state of legal affairs S_1 into a new state of legal affairs S_2.

Hohfeld's examples imply that a change in legal states of affairs is effected through the performance of some *specific* action. This is made explicit by Jones & Sergot (1996):

> "It is a commonplace feature of legal systems, and other norm-governed organisations, that particular agents are empowered to create certain types of states, by means of performance of specified types of acts. Typically the states created will have a normative character […]. The performance by means of which these states are established will often be of a clearly prescribed, perhaps ritualised nature, involving the utterance of a particular form of words, […], or the production of a formal document, or the issuing of a pass, perhaps in a particular context (e.g., in the presence of witnesses".

Where we might want to distinguish the cases where a legal state of affairs is not effected because the specific action was not performed, from the cases where a legal state of affairs is not effected because the agent is not empowered (in Austin's spirit, cf. section 2.4.2), then a representation along the lines of (H7)–(H10) is preferable to a representation along the lines of (L1)–(L4) or (K1)–(K4). Allen & Saxon's (1993) formulation also shows explicitly the action that the empowered agent must take in order to effect a legal state of affairs.

As regards the action performed by an empowered agent, a point worth considering is whether it is always positive. In other words, are expressions such as the following meaningful?

(i) $power(p_1, p_2, \neg E_{p_1}\alpha, S_1, S_2)$

(ii) $power(p_1, p_2, E_{p_1}\neg\alpha, S_1, S_2)$

(iii) $power(p_1, p_2, \neg E_{p_1}\neg\alpha, S_1, S_2)$

These correspond to saying that an agent has power to change legal relations S_1 into S_2 by (i) refraining from performing a positive action; or (ii) performing a negative action; or (iii) refraining from performing a negative action. What counts as positive or negative action depends largely on how one chooses to fix the meaning of propositions such as α. As we saw in the previous section, there are situations where we might want to retain the distinction between performance and non-performance of an action, rather than operate in a 'closed world'. One can conceive of situations where it is meaningful for an agent to create legal states of affairs by refraining from action: a mail-order book club sends unsolicited books to an agent along with an offer for him to become a member and instructions that if he wishes to refuse the membership he must return the items within a specified period of time, otherwise, if he does nothing, he becomes a member. The agent is therefore empowered to create legal relations between himself and the book club (transforming them from being essentially empty, to some situation that involves duties for him to pay or order more books etc.) through refraining from any action.

Another interesting point that arises in relation to the empowered agent's action is whether such action is qualified in any way, in particular deontically. In other words, the question is whether an empowered agent is permitted or obliged to exercise his power to effect a change in legal affairs. As we saw earlier, in Makinson's (1986) example, an agent's power may indeed be deontically qualified. There are indeed many examples in contractual documents of permissions and obligations ranging over the exercise of power.

One might also consider how the transformation of states of affairs is actually specified. In other words, is it appropriate to regard the exercise of an agent's power as an event that initiates or terminates properties which in this case are legal relations (in situation calculus terms ((McCarthy 1968), (McCarthy & Hayes 1969)) or as an event with an explicit temporal dimension in event calculus terms (Kowalski & Sergot 1986))? In addition, what happens if the empowered agent's action is temporally qualified and is not performed within the specified time period? As we saw in chapter 2, Austin's general theory of performatives, in particular his Doctrine of Infelicities (1962) provides some useful insights in relation to this last question.

As was the case with relations of the right-set, it is possible to complete the space of relations in the power-set by equivalence substitution. Reconstructing opposite and correlative relations from Hohfeld's examples gave rise to (H7)–(H10) above. We can also derive relations between 'power' and 'immunity', and 'disability' and 'liability', that is, the concepts that lie along the same row in the power-set:

(H11) $power(p_1, p_2, E_{p_1}\alpha, S_1, S_2) \equiv \neg immunity(p_2, p_1, E_{p_1}\alpha, S_1, S_2)$

(H12) $disability(p_1, p_2, E_{p_1}\alpha, S_1, S_2) \equiv \neg liability(p_2, p_1, E_{p_1}\alpha, S_1, S_2)$

Clauses (H11) and (H12) can be regarded as 'correlatively opposite' and they now enable us to define all the relations in the power-set in terms of 'power':

(P1) $immunity(p_2, p_1, E_{p_1}\alpha, S_1, S_2) \equiv \neg power(p_1, p_2, E_{p_1}\alpha, S_1, S_2)$

(P2) $liability(p_2, p_1, E_{p_1}\alpha, S_1, S_2) \equiv power(p_1, p_2, E_{p_1}\alpha, S_1, S_2)$

(P3) $disability(p_1, p_2, E_{p_1}\alpha, S_1, S_2) \equiv \neg power(p_1, p_2, E_{p_1}\alpha, S_1, S_2)$

3.5.3 Elements of the Theory of Normative Positions

The theory of normative positions developed by Sergot (1998b) and automated by Sergot (1998a) is an extended and generalised version of the Kanger–Lindahl theory ((Kanger 1971); (Lindahl 1977)). The theory attempts to apply a combination of deontic logic and the logic of action/agency to the formalisation of Hohfeld's (1913) "fundamental legal conceptions". The generalised theory includes automated inference methods which have been implemented in computer programs intended to facilitate application of the theory to the analysis of practical problems, either for the purpose of interpretation and disambiguation of legal texts, or in the design and specification of a new set of norms (Jones & Sergot 1992; 1993). The objective is to clarify and expand an incomplete and imprecise statement of requirements into a precise formal specification at some desired level of detail. The role of the system is to guide this process, ensuring overall consistency and identifying any possibilities that remain to be explored.

The distinctive features of the theory of normative positions, extended and generalised by Sergot, are:

(i) Whereas the original Kanger-Lindahl theory accounted for normative relations between two agents, the generalised theory deals with any number of agents as well as normative propositions where no agent is specified (i.e., propositions of the form "it ought to be the case that *X*").

(ii) The original Kanger-Lindahl theory employs Pörn's (1977) relativized action operator E_a, with expressions of the form $E_a X$ denoting "agent *a* brings it about that *X*". The generalised theory allows any number of such expressions to be combined and allows compound acts in the scope of the action operator.

(iii) The generalised theory separates the method of generating normative positions between agents from the underlying deontic and action logics. Hence, it is possible to apply any specific deontic and action logic to reason with normative positions, without changing the overall framework for generating normative positions. An advantage of such separation is that it is possible to rely on the structure of normative positions in order to automate the application of the theory to practical problems rather than on modal logic theorem provers.

The original Kanger-Lindahl theory employs a deontic logic component, which is very close to SDL (Chellas 1980), an action logic component and a method for generating the space of all logically possible normative positions between two agents. The language employed is that of propositional logic augmented with two modal operators, *Shall* (for obligation) and *May* (for permission, the dual of obligation) and one action operator *Do.* Hence, *ShallA* denotes "*A* shall be the case", or in other words "it is obligatory that *A*", *MayA* denotes "*A* may be the case", or in other words "it is permissible that *A*", and *Do(a, A)* denotes "agent *a* does *A*" or "agent *a* brings it about that *A*". Here we use *O* for obligation, *P* for permission and $E_a X$ for act descriptions.

The action operator is a success operator ($\vdash E_x p \rightarrow p$, that is, if *x* brings it about that *p*, then *p* is the case) and is closed under logical equivalence (that is, if $\vdash p \leftrightarrow q$ then $\vdash$ $E_x p \leftrightarrow E_x q$).

Given two agents *a* and *b* and one state of affairs *F*, Kanger (and following him Lindahl) constructs characterisations of the agents position with respect to the given state of affairs. The expressions, which Kanger calls "simple types of rights relations" fall under the scheme:

$$\pm O \pm \begin{pmatrix} E_a \\ E_b \end{pmatrix} \pm F$$

The notation was suggested by Makinson (1986) and is used by Sergot (1998b). There are sixteen possible expressions. The eight for agent *a* are summarised in the first column of Table 3–6 (the other eight for agent *b* are identical to those, with E_b substituted for E_a). The second column of the table shows the equivalent expression in terms of the permission operator, given that $P \equiv_{def} \neg O \neg$:

$OE_a F$	
$OE_a \neg F$	
$O \neg E_a F$	
$O \neg E_a \neg F$	
$\neg OE_a F$	$P \neg E_a F$
$\neg OE_a \neg F$	$P \neg E_a \neg F$
$\neg O \neg E_a F$	$PE_a F$
$\neg O \neg E_a \neg F$	$PE_a \neg F$

Table 3-6 Kanger's simple types of rights for one agent

The sixteen simple types of rights relations between two agents can be combined to form what Kanger calls "atomic types of rights relations" for two agents *a* and *b* with respect to bringing about a state of affairs *F*. These compounds, according to Makinson (1986) are expressions belonging to the set

$$\left\| \pm O \pm \begin{pmatrix} E_a \\ E_b \end{pmatrix} \pm F \right\|$$

The double bar[21] in this expression denotes maximal consistent conjunctions of expressions falling under the scheme in the brackets. That is, conjunctions without repetitions and with some standard order and association of conjuncts, which are consistent with respect to some underlying logic (here the deontic and action logics employed by Kanger and Lindahl) and maximal, that is, addition of any other conjunct from the scheme yields an inconsistent conjunction.

Sergot (1998b) proves that this expression can be re-written as follows:

$$\left\| \pm O \pm \begin{pmatrix} E_a \\ E_b \end{pmatrix} \pm F \right\| = \left\| \pm O \pm E_a \pm F \right\| \cdot \left\| \pm O \pm E_b \pm F \right\|$$

Where **P** and **Q** represent set expressions, $\mathbf{P} \cdot \mathbf{Q}$ stands for the set of all the consistent conjunctions with one conjunct from **P** and one conjunct from **Q**. The maximally consistent conjunctions in $\left\| \pm O \pm E_a \pm F \right\|$ are what Jones & Sergot (1992; 1993) call Kanger's "normative one-agent act positions". According to the logic employed by Kanger, there are six expressions (after eliminating redundancies), that Lindahl numbers as follows:

K_1	$PE_aF \wedge PE_a\neg F$
K_2	$O\neg E_aF \wedge O\neg E_a\neg F$
K_3	OE_aF
K_4	$PE_aF \wedge P\neg E_aF \wedge O\neg E_a\neg F$
K_5	$OE_a\neg F$
K_6	$O\neg E_aF \wedge PE_a\neg F \wedge P\neg E_a\neg F$

Table 3-7 Kanger's normative one-agent act positions

These are internally consistent, mutually exclusive and jointly exhaustive. They completely characterise one agent's position with respect to bringing about a certain state of affairs, according to the logical principles employed.

For Kanger's atomic types of rights for two agents, there are therefore 36 conjunctions to consider, with one conjunct from $\{K_1 \ldots K_6\}$ for agent *a* and one conjunct from $\{K_1 \ldots K_6\}$ for agent *b*. Of those 10 turn out to be logically inconsistent, leaving 26 internally consistent, mutually exclusive and jointly exhaustive expressions.

Lindahl constructs his analysis on the set of normative one-agent act propositions denoted by:

[21] In Makinson (1986) and Sergot (1998b) double brackets are used.

$$\left\|\pm P\left\|\pm E_a \pm F\right\|\right\|$$

where there is a maxi-conjunction expression within the scope of the P operator. The set $\left\|\pm E_a \pm F\right\|$ gives rise to the following "act positions":

(A$_1$) E_aF

(A$_2$) $E_a\neg F$

(A$_3$) $\neg E_aF \wedge \neg E_a\neg F$

The third one, according to which agent *a* is passive with respect to the state of affairs *F*, is the one not included in Kanger's analysis. Lindahl abbreviates it to $Pass_aF$. After eliminating logical redundancies $\left\|\pm P\left\|\pm E_a \pm F\right\|\right\|$ gives rise to the following expressions:

T$_1$	$PE_aF \wedge PE_a\neg F \wedge PPass_aF$
T$_2$	$PE_aF \wedge O\neg E_a\neg F \wedge PPass_aF$
T$_3$	$PE_aF \wedge PE_a\neg F \wedge \neg PPass_aF$
T$_4$	$O\neg E_aF \wedge PE_a\neg F \wedge PPass_aF$
T$_5$	OE_aF
T$_6$	$OPass_aF$
T$_7$	$OE_a\neg F$

Table 3-8 Lindahl's normative one-agent act positions

In Lindahl's analysis two-agent positions are constructed from:

$$\left\|\pm P\left\|\pm E_a \pm F\right\|\right\| \cdot \left\|\pm P \pm \left\|\pm E_b \pm F\right\|\right\|$$

Of the 49 conjunctions to consider, only 35 are internally consistent and are what Lindahl calls "individualistic normative two-agent act positions". As Sergot (1998b) points out Lindahl's analysis is more refined than Kanger's. Kanger's (K$_1$) corresponds to the disjunction of Lindahl's (T$_1$) and (T$_3$). Similarly, there are equivalences between Kanger's 26 two-agent atomic types and Lindahl's 35 individualistic two-agent act positions.

Sergot's analysis is more refined than Lindahl's. Sergot takes act positions from the following scheme:

$$\left\|\pm E_a \pm F\right\| \cdot \left\|\pm F\right\|$$

The resulting act positions, which Sergot calls “cumulative fact/act positions” are:

(A$_1$) $E_a F$

(A$_2$) $E_a \neg F$

(A$_{3a}$) $F \wedge \neg E_a F$

(A$_{3b}$) $\neg F \wedge \neg E_a \neg F$

Sergot notes that Lindahl’s passive act position (A$_3$) does not distinguish between (A$_{3a}$) and (A$_{3b}$). The corresponding one-agent normative act positions are:

$$\left\| \pm O \pm \left\| \pm E_a \pm F \right\| \left\| \pm F \right\| \right\|$$

This gives rise to 15 internally consistent conjunctions, of which three are identical to Lindahl’s. Lindahl’s other four correspond to a disjunction of Sergot’s conjunctions:

$$\mathrm{T}_1 \begin{cases} PE_a F \wedge PE_a \neg F \wedge P(F \wedge \neg E_a F) \wedge P(\neg F \wedge \neg E_a \neg F) \\ PE_a F \wedge PE_a \neg F \wedge O(F \rightarrow E_a F) \wedge P(\neg F \wedge \neg E_a \neg F) \\ PE_a F \wedge PE_a \neg F \wedge P(F \wedge \neg E_a F) \wedge O(\neg F \rightarrow E_a \neg F) \end{cases}$$

$$\mathrm{T}_2 \begin{cases} PE_a F \wedge \neg PE_a \neg F \wedge P(F \wedge \neg E_a F) \wedge P(\neg F \wedge \neg E_a \neg F) \\ PE_a F \wedge \neg PE_a \neg F \wedge O(F \rightarrow E_a F) \wedge P(\neg F \wedge \neg E_a \neg F) \\ PE_a F \wedge \neg PE_a \neg F \wedge P(F \wedge \neg E_a F) \wedge O(\neg F \rightarrow E_a \neg F) \end{cases}$$

$$\mathrm{T}_3 \left\{ PE_a F \wedge PE_a \neg F \wedge \neg PPass_a F \right.$$

$$\mathrm{T}_4 \begin{cases} \neg PE_a F \wedge PE_a \neg F \wedge P(F \wedge \neg E_a F) \wedge P(\neg F \wedge \neg E_a \neg F) \\ \neg PE_a F \wedge PE_a \neg F \wedge O(F \rightarrow E_a F) \wedge P(\neg F \wedge \neg E_a \neg F) \\ \neg PE_a F \wedge PE_a \neg F \wedge P(F \wedge \neg E_a F) \wedge O(\neg F \rightarrow E_a \neg F) \end{cases}$$

$$\mathrm{T}_5 \left\{ OE_a F \right.$$

$$\mathrm{T}_6 \begin{cases} OPass_a F \wedge OF \\ OPass_a F \wedge O\neg F \\ OPass_a F \wedge PF \wedge P\neg F \end{cases}$$

$$\mathrm{T}_7 \left\{ OE_a \neg F \right.$$

Two-agent individualistic and collectivistic positions in Sergot's extended framework are constructed as the following maxi-conjunctions, respectively:

$$\left\| \pm O \pm \left\| \pm E_a \pm F \right\| \cdot \left\| \pm F \right\| \right\| \cdot \left\| \pm O \pm \left\| \pm E_b \pm F \right\| \cdot \left\| \pm F \right\| \right\|$$

$$\left\| \pm O \pm \left\| \pm \begin{pmatrix} E_a \\ E_b \end{pmatrix} \pm F \right\| \cdot \left\| \pm F \right\| \right\| = \left\| \pm O \pm \left\| \pm E_a \pm F \right\| \cdot \left\| \pm E_b \pm F \right\| \cdot \left\| \pm F \right\| \right\|$$

To illustrate the application of the theory of normative positions to the analysis of norms, consider an extract from a sample contract concerned with *force majeure*.

(C4) A Party intending to seek relief under this article shall not be entitled to such relief unless such Party shall

> (a) as soon as practicable, but within 7 days of the day upon which the Party first knew of the failure to fulfil its obligation, notify the other Party and provide the other Party with an interim report containing all relevant information
>
> (b) within 30 days provide the other Party with a full report which shall amplify the information contained in the interim report and contain further information as the other Party may reasonably require
>
> (c) upon request give or procure access for representatives of the other Party to examine the scene of the event which gave rise to the failure and such access shall be at the expense of the Party who failed

(C5) Relief under this article shall cease to be available to a Party in respect of an event of Force Majeure if it fails to take as soon as practical all necessary steps to rectify the cause of the failure.

Let A denote that relief from obligations under *force majeure* circumstances is granted, B denote that a party provides an interim report within 7 days, C denote that a party provides a full report within 30 days, D denote that a party gives access to the other party and G denote that a party takes steps to rectify the cause of *force majeure*. Let the index a denote the party that provides relief from *force majeure* and β denote the party seeking relief from *force majeure*. According to the clauses above and the one-agent act-descriptions noted earlier, one possibility for the normative content of these clauses is:

(N1) $(OE_\beta B \wedge OE_\beta C \wedge OE_\beta D) \wedge (\neg(E_\beta B \wedge E_\beta C \wedge E_\beta D) \rightarrow \neg OE_a A$

(N2) $OE_\beta G \wedge (\neg E_\beta G \rightarrow \neg OE_a A)$

The expression “A Party…shall not be entitled…” is captured by $\neg OE_a A$ conforming to the Hohfeldian connection between right and duty (here right is taken to be entitlement and duty is taken to be obligation). The consequent of both norms could be different, depending on what the intended interpretation is. In the version given above, agent a is not obliged to offer relief from liability. Other possible interpretations might be:

- $PE_a A \wedge PE_a \neg A \wedge O(E_a A \vee E_a \neg A)$: agent a is permitted to see to it that relief is granted, permitted to see to it that relief is not granted but obliged to see to it that relief is either granted or not granted;

- $PE_a A \wedge PE_a \neg A \wedge P(\neg E_a A \wedge \neg E_a \neg A)$: agent a is permitted to do nothing about the claim (which seems unlikely as an intended interpretation but the text of the provision alone does not rule it out)

- $OE_a \neg A$: agent a is obliged to see to it that relief from liability is not granted.

There are two points worth discussing here. First, in the example it is not clear whether a party ceases to be entitled to relief from liability by not complying with any or all of the three conditions set out in clause (C4). This is where Allen (1957) is right about ambiguity arising from the relations between sentences. If the conditions (a)–(c) are implicitly conjoined (which seems reasonable to assume), then (N1) above is the intended interpretation. Second, it might be argued that the party seeking relief is not strictly speaking obliged to see to it that A, B, C, D are the case. That is, sub-clauses (a)–(c) might be better conceived as procedural steps that a party seeking relief from liability due to *force majeure* needs to complete in order for it to be granted such relief. Such a reading of the clauses would entail that a party performing these steps has the power to bring about a state of affairs in which it is granted relief from liability due to *force majeure*. Power here is meant in the Hohfeldian sense, that is, it refers to the ability to create legal relations, in this instance the obligation on the other party to grant relief.

Strictly speaking granting relief from liability itself is a power relation. That is, the party granting relief from liability is establishing a state of affairs in which the relieved party is not bound by its obligations. If a party seeking relief from liability performs (a)–(c), then the other party is obliged to exercise that power.

The main benefit of the theory of normative positions is therefore that it brings to the foreground questions about intended interpretation, structural ambiguity, the precise nature of legal notions (whether they are primary or concern other legal relations) and so on. It is a powerful tool in addressing interpretation questions and in disambiguating between legal notions and it can be integrated with the theoretical framework developed by Alchourrón & Bulygin (1971) to address normative consistency, completeness and independence. The next section discusses Alchourrón & Bulygin's framework.

3.5.4 Analysis of Normative Systems

Carlos Alchourrón and Eugenio Bulygin (1971) provide an explication for consistency, completeness and independence, the formal properties of normative systems. Their framework is theoretical, in that it is practically impossible to conduct extensive case analysis but for very narrow domains. It is useful however, in that it provides a formal basis for characterising the consistency, completeness and independence of a set of norms.

In Alchourrón & Bulygin's framework, a normative problem is the deontic characterisation of a set of actions U_A for an agent, that is, the designation of individual actions in the set as permitted, obligatory, forbidden and so on. The set of situations in which such actions may take place is called the 'universe of discourse' U_D: these are the situations for which the drafter or analyst of a set of norms has to cater by specifying what actions are permitted, obligatory, forbidden and so on. In other words a normative problem requires the association of situations of U_D with deontically characterised actions of U_A; such deontically characterised actions are called *solutions*.

The situations of the universe of discourse are essentially state-descriptions, that is, truth-functional compounds of the properties that hold in them. Alchourrón & Bulygin's method requires the identification of a finite set of properties U_P that make up situations. Such properties are intended to be the ones relevant to the normative problem and their selection plays a central role in the whole approach, as we shall see in the discussion section later. Truth-functional compounds of properties of the U_P define the factual range of the problem, the set U_C of elementary cases. It can be shown that every situation in the universe of discourse belongs to exactly one elementary case. Put alternatively, the universe of cases U_C comprises schemata for cases and partitions the individual cases of the universe of discourse in classes. To provide solutions for the situations in U_D it suffices to provide solutions for the elementary cases U_C.

The set of solutions comprises deontically characterised truth-functional compounds of possible actions from the U_A. Possible characterisations are: Obligatory (O), prohibited

($Prohib$), optional or facultative ($Choice$) and permitted (P). Deontic logicians have debated on whether these notions are inter-definable and on which should be taken as primitive. Alchourrón & Bulygin take P as the primitive deontic operator and define the others as follows, where x denotes an action:

- $Ox \equiv_{def} Px \wedge \neg P \neg x$

- $Prohib\ x \equiv_{def} \neg Px \wedge P \neg x$

- $Choice\ x \equiv_{def} Px \wedge P \neg x$

These definitions capture some common-sense intuitions: What is obligatory is not prohibited and vice versa; a facultative act is neither prohibited nor obligatory; an act is obligatory if its complementary is prohibited.

The set of solutions U_S to the normative problem comprises therefore expressions involving these deontic operations. The correlation of solutions with cases is done through the norms that are being examined, that is, the linguistic expressions of the legislative or regulatory text. This is illustrated by the following example taken from a sample contract. The provisions concern relief from liability under *force majeure* circumstances with unnecessary detail omitted for the sake of clarity and space:

> The expression "Force Majeure" shall mean any event or circumstance which is beyond the control of the Party concerned, acting or having acted as a reasonable and prudent operator, resulting in or causing the failure by such Party to perform any of its obligations under this agreement, which failure could not have been prevented or overcome by the Party.

(C1) The circumstances which shall be within the definition of Force Majeure (fulfilling the requirements of clause C1) shall include but not be limited by the following: Acts of God, forces of nature, epidemic and quarantine restrictions, land-sliding, lightning, earthquakes, fire, floods, storms, tidal waves, strikes, lock-outs or other industrial disturbances, sabotage, acts of war (etc.).

(C2) Notwithstanding anything in clauses C1 and C2 the following events shall not be treated as Force Majeure:
[NB: List of situations that do not qualify as Force Majeure]

(C4) A Party intending to seek relief under this article shall not be entitled to such relief unless such Party shall

(a) as soon as practicable, but within 7 days of the day upon which the Party first knew of the failure to fulfil its obligation, notify the other Party and provide the other Party with an interim report containing all relevant information

(b) within 30 days provide the other Party with a full report which shall amplify the information contained in the interim report and contain further information as the other Party may reasonably require

(c) upon request give or procure access for representatives of the other Party to examine the scene of the event which gave rise to the failure and such access shall be at the expense of the Party who failed

(C5) Relief under this article shall cease to be available to a Party in respect of an event of Force Majeure if it fails to take as soon as practical all necessary steps to rectify the cause of the failure.

For this example our Universe of Discourse U_D contains all situations in which any of the parties that have come into the agreement fail to fulfil its obligations and seeks relief from liability attributing such failure to *force majeure*. The question we seek to answer in all those situations is whether the party that failed to fulfil some obligation is granted relief from liability.

We must choose the properties that will be included in our Universe of Properties U_P and to do so we are guided by the clauses (C1)–(C5), which are relevant to the problem. From clauses (C1)—(C3) we see that the nature of the event, in which failure to perform an obligation occurs, is relevant, as some circumstances are recognised as *force majeure* whereas others are not. We can denote all circumstances that qualify as *force majeure* ones by A. From clause (C1) it also seems obvious that the conduct of the Party seeking relief is also relevant, that is, whether the Party acted as a reasonable and prudent operator, and we can denote that by B. The performance of certain actions by the party seeking relief is relevant as can be seen by clause (C4) and we shall denote the property of those actions being executed as C. Finally, rectifying the cause of a *force majeure* event is relevant to whether relief is granted by clause (C5) and D denotes this. The elementary cases to which these four properties give rise are 16.

The action that is to be deontically characterised is providing relief from liability to a party due to *force majeure*, here denoted by ρ.

In contracts, there are at least two agents whose actions we want to characterise deontically. Deontic operators are therefore indexed by a (for the party granting relief), β (for the party seeking relief) and x (where this distinction is not important).

The norms expressed by provisions (C1)–(C5) can be reformulated as follows, where each norm takes the form $^{solution}\!/_{case}$:

(N1) $^{O_a \rho}\!/_{A \wedge B \wedge C \wedge D}$

(N2) $\mathit{Prohib}_a \; \rho \big/_{\neg C}$

(N3) $\mathit{Prohib}_a \; \rho \big/_{\neg D}$

(N4) $\mathit{Prohib}_a \; \rho \big/_{\neg A}$

Their association with the elementary cases identified earlier is shown below:

$\mathbf{C}_i$	A	C	B	D	N1	N2	N3	N4
C_1	+	+	+	+	+			
C_2	+	+	+	–			+	
C_3	+	+	–	+				
C_4	+	+	–	–			+	
C_5	+	–	+	+		+		
C_6	+	–	+	–		+	+	
C_7	+	–	–	+		+		
C_8	+	–	–	–		+	+	
C_9	–	+	+	+				+
C_{10}	–	+	+	–			+	+
C_{11}	–	+	–	+				+
C_{12}	–	+	–	–			+	+
C_{13}	–	–	+	+		+		+
C_{14}	–	–	+	–		+	+	+
C_{15}	–	–	–	+		+		+
C_{16}	–	–	–	–		+	+	+

Table 3-9 Association of elementary cases with solutions

Solutions that appear in the same column derive from the same norm that corresponds to that column. Solutions that appear in the same row are the ones that can be derived from the system for the elementary case that corresponds to that row. A case which is not associated with any solution (i.e. no solution appears in its row) is a *normative gap* (for example C_3 above). As no solution is derived for such an elementary case, no solution is provided by the system for any individual case of the universe of discourse that belongs to the class of that elementary case. A normative system with at least one normative gap is called *incomplete.* A system is *inconsistent* in a case C_i if and only if there are two or more incompatible solutions associated with it; otherwise, the system is *consistent.* The notion of incompatibility between two or more norms is determined by the system of deontic logic that is being used. In the example given here, two norms would be incompatible if one was prescribing an obligation for an agent to do α and the other a prohibition for the same agent to do α under the same circumstances. A system is *redundant* in a case C_i if and only if the same solution is associated with a case through two or more norms (i.e. it appears at least twice in the row associated with the case). The system of the example is redundant as can be seen for example from C_6, C_8, C_{10}, C_{12}, C_{13}, C_{14}, C_{15} and C_{16}. The norms of a system are said to be *independent* if and only if there is no case in which the system is redundant. So the system of the example is incomplete, consistent and redundant and only (N1) is independent of the other norms. As Alchourrón & Bulygin point out, though the presence of redundant norms is undesirable, one must be careful when trying to adjust them lest removal of a norm leaves normative gaps.

Alchourrón & Bulygin's approach offers an attractive theoretical framework within which to view normative systems and their formal properties for legal drafting. The criteria for well-formed provisions can be addressed through an analysis such as the one conducted on the example. Hence the effects of a norm, whether there is ambiguity that needs clarification, the consistency and completeness of a given set of norms can be addressed in the process of associating solutions with cases. It is not quite so easy to put their proposal to practice however and the example that was presented has already raised some issues with the reader:

How are we certain that the properties chosen to define elementary cases are appropriate, or even the *only* appropriate ones? In the example, a simplification was made and various circumstances were grouped under A, B, C or D. How can we be certain that we have chosen *all* of the relevant properties? Alchourrón & Bulygin stipulate that the choice of properties is crucial and that they must fulfil certain requirements:

They must be *logically independent* otherwise, they will give rise to logically empty cases. They must also be *empirically independent* otherwise, they will give rise to empirically empty cases. It is possible for two properties to be logically but not empirically independent (that is, they may have a causal relation). A logically complete system is empirically complete as well so the requirement for logical independence is stronger. Properties must also be logically independent of the actions that we are trying to characterise deontically, otherwise we might get solutions that would determine deontically certain actions that are impossible to realise. Finally the set of properties chosen and the universe of discourse (the set of all situations that the system is meant to address) must be corresponding, otherwise we can no longer rely on providing solutions for elementary cases in U_C and completeness, consistency and independence of the norms for U_C do not necessarily entail completeness, consistency and independence of norms for the universe of discourse.

A second point concerns the actual association of solutions with cases with reference to the linguistic expressions on the legislative, regulatory, or contractual text. The process itself relies on interpretation. For instance, in the example clause (C5) was taken to mean that a party is prohibited from offering relief from liability if the other party does not rectify the cause of *force majeure*. The expression “Relief under this article shall cease to be available to a Party…” however might be regarded as implying that the party seeking relief has no right (in the Hohfeldian sense (Hohfeld 1913)) to expect to be granted relief, that is, that there is no obligation on the other party to provide it. The alternative interpretations of (C5) might therefore be $Choice_a\ \rho$ or $\neg O_a \rho$.

A third point concerns the possible deontic characterisations for actions supported by the logic employed by Alchourrón & Bulygin: Given an action x the possible permission statements that can be formulated for it and its complement are:

$Px \wedge \neg Px$	Contradictory
$Px \wedge P\neg x$	$Choice\ x$
$Px \wedge \neg P\neg x$	Ox
$\neg Px \wedge P\neg x$	$Prohib\ x$ or $O\neg x$
$\neg Px \wedge \neg P\neg x$	Obligation to refrain from action
$P\neg x \wedge \neg P\neg x$	Contradictory

Table 3-10 Possible permission statements for action and its complement

Contradictory statements aside, Alchourrón & Bulygin account for obligatory actions and their dual, prohibited ones. They also account for facultative actions but not for what might seem to be their dual, ‘obligations to remain passive’. What expressions of this kind essentially mean is that an agent is not responsible for an action x or alternatively that x is *ultra vires* for a given agent. This seems to deprive the framework from the ability to account for norms pertaining to agent responsibility. Recent systems of deontic logic do allow for this possibility as is the case in the theory of normative positions, which was discussed in the previous section.

A fourth point concerns the action itself that is deontically characterised and the properties chosen. The representation as it stands does not account explicitly for temporal conditions. Hence, what might be intended by (N1) could be that it is obligatory to provide relief from liability *after* a claim has been made and *during* the period in which *force majeure* circumstances obtain, rather than a temporally unqualified obligation. As regards temporally characterised norms themselves, we might want to establish the consistency and completeness of them with respect to *time*, which would call for a different kind of analysis from the one presented here.

As mentioned in the previous section Alchourrón & Bulygin's framework can be integrated with normative positions for agents, each associated with individual cases. In the framework as it stands, cases are associated with solutions from $\|\pm O \pm F\|$, which are essentially fact positions. These are mutually exclusive and jointly exhaustive. So, if a case is associated with more than one fact positions, the normative system is inconsistent, and if a case is associated with no fact position, the normative system has a gap. Instead, solutions could be chosen from normative positions, à la Kanger $\|\pm O \pm E_a \pm F\|$, à la Lindahl $\|\pm P\|\pm E_a \pm F\|\|$ or à la Sergot $\|\pm O \pm \|\pm E_a \pm F\| \cdot \|\pm F\|\|$. These are also mutually exclusive and jointly exhaustive so the definitions for inconsistency and gaps are the same. But the analysis obtained from them is more detailed and agency for deontically characterised actions is represented explicitly, for one or two agents in Kanger-Lindahl terms and for any number of agents in Sergot terms.

Finally, the representation as it stands does not allow for temporally qualified normative statements. These are very frequently encountered in contracts where parties have obligations that only arise within specific times (whether these are in absolute or relevant terms). Such an extension to the representation would yield different definitions for the consistency and completeness of a set of norms.

4

CONTRACT DRAFTING: DOCUMENT STRUCTURE

4.1 Introduction

This chapter concentrates on legal contract drafting. As mentioned earlier in this dissertation, the term ‘contract’ refers both to a legally binding agreement between (usually) two parties and the document that records such agreement. Consequently ‘contract drafting’ may refer both to the design of agreements and to the design of documents that record such agreements.

Earlier work on the drafting of legislation (Bench-Capon 1987) drew a distinction between the formulation of policies and their implementation via legislative text. An analogy can be made whereby the formulation of a policy is similar to the formulation of a specification (in software engineering terms) and is carried out by Parliament or the appropriate policy-making body. The drafting of legislation, extending this analogy, is then akin to programming; the drafted legislation is intended to implement the specification—the policy that was formulated. A similar analogy can be made for the case of contracts. A contract between (usually) two parties can be considered as an agreement, which is negotiated and finalised as a specification of the goals that are to be achieved in the course of the business exchange regulated by the agreement. Then, the document that records such an agreement is the implementation of such a specification.

Consequently, agreement design and document design raise different issues. For the former, some issues to consider include how parties’ goals are to be represented and checked for consistency. In case of conflicts, how is negotiation to proceed, what compromises can be made that are mutually acceptable to both parties, based on each party’s criteria (and preferences) for what is ‘best’ for it? What is the set of norms that accurately reflects what the parties mutually agree on, as regards the regulation of their exchange, and how is overall consistency and completeness of such norms to be established? As regards document design, the main issue to consider is what structure and linguistic content accurately reflect the agreement that the document records.

It is difficult to demarcate precisely between the design of agreements and the design of documents merely by distinguishing between the negotiation phase and the document drafting phase. It is often the case that negotiation is carried out based on drafts of a contractual document; frequently the document itself *is* the object of negotiation, for example in terms of its express terms and the actual wording employed. However, to some extent we can usefully decompose the problem of contract drafting. At one level, we can concentrate on the document in the large, with emphasis placed on its broad contents, its structure and overall coherence; this is what we call contract drafting at the *macro-level*. At another level, we can concentrate on the contents of the contractual document in detail, with emphasis on the formulation of individual provisions; this is what we call contract drafting at the *micro-level*.

In the rest of this chapter, we concentrate on contract drafting at the macro-level. The next section presents briefly existing work in this area to enable comparison with this research. The rest of this chapter presents the approach developed in the course of this research for the drafting of contracts at the macro-level along with MODELLER, a prototype system that implements it.

4.2 Context

Document drafting in general has been mainly the concern of the Electronic Publishing community and much of its research centred on the development of appropriate document models. Some ideas stemming from work in Electronic Publishing influence this research and they are briefly discussed in what follows. This section also reviews existing legal document drafting systems and their underlying document models.

4.2.1 Document Representation

The central view that underlies research in the Electronic Publishing community is that text can be viewed as an organized collection of objects. The actual characterization of what an object is depends on the intended application for which the text is modelled. Furuta (1989) noted three different models for a given document, namely the *abstract* model of the document, its *concrete appearance* and its *concrete representation.* The abstract model of a document represents its 'logical'[22] structure in terms of its constituent parts and the way in which they are organised (such constituent parts are, for example, chapters, paragraphs, strings of characters, depending on the desired level of detail). The concrete appearance of a document describes it in terms of the layout of its components (for example, on a page basis, in terms of number of lines, number of characters per line, depending on the desired level of detail). The concrete representation of a document describes it in terms of its actual appearance on a display medium (for example, the font and size of individual characters). Having explicit different representations of a document enables independent manipulation of its different aspects. Hence one can change the structure of a piece of text without dealing with its layout or its actual appearance, or alternatively one can change the layout of a document without affecting its contents.

Of the three models for a given document, the first, that is, the abstract model of its 'logical' structure, is of particular interest in the context of this research. Bench-Capon & Dunne (1989) stress the importance of modelling the 'logical' structure of documents as a reader's awareness of structural conventions enhances his understanding of the contents of a document. Bench-Capon & Dunne note that the structural conventions introduced by the author of a document need not be the same in a computer model for the document. Hence the abstract model of the 'logical' structure of a document could encapsulate the author's view (what Bench-Capon & Dunne call *conceptual representation*) or just be a convenient computer model (what Bench-Capon & Dunne call *abstract representation*). Koo (1989) agrees with such a distinction between the conceptual and abstract representation and notes that the two should be closely linked so that the process of producing the final version of a document is facilitated, the functionality and extensibility of a document management system is enhanced, and common problems in collaborative authorship (for example the use of a uniform notation, the format of the text and so on) are solved.

This requirement, that the model of the 'logical' structure of a document be closely linked to the computer model for that document suggests that structural conventions introduced by the author should be captured explicitly. In this way, the abstract and conceptual representations of a document can be closely linked, indeed coincide.

Any given document typically falls into a particular category, which is characterised by its own structural conventions. For example, the structure of novels is different in some respects

[22] The term 'logical' is in quotes, as it is used to refer to syntactic structure by researchers in Electronic Publishing.

from the structure of academic textbooks, and the latter differ from short articles. The structure of tabulated text is different from the structure of non-tabulated text. The structure of alphabetically sorted text is different from the structure of text that is not sorted. The common feature shared by all categories of documents is that text is made up of constituents (paragraphs, sentences, phrases or individual characters depending on the desired level of detail). The way constituents are combined to form larger structures differentiates between categories. Modelling the structure of categories rather than individual documents is therefore desirable, as it is more general. This gives rise to a refinement of the requirement introduced earlier:

> Requirement
>
> The abstract representation of a document should be an instance of the conceptual representation corresponding to the category for that document, which includes the structural conventions introduced by the author.

The most general model for document categories is a graph[23], whose nodes correspond to text constituents and whose arcs correspond to relations between them. Koo (1989) argues that a directed acyclic graph is the most appropriate general model for most document categories, with arcs corresponding to containment relations between nodes that represent document constituents.

Containment relations are of course of principal interest when one examines the syntactic structure of text. They are not however the only useful relations between constituent parts of text. Explicit representation of containment relations disambiguates the syntactic structure of text (its form) but it does not do much towards disambiguating its semantic structure in functional terms[24]. Functional relations between document constituents are part of the logical structure or text. Re-visiting the point made by Bench-Capon & Dunne (1989), a reader's understanding of text is enhanced if he is aware of the logical (syntactic and functional) structure of text. This leads to a further refinement of the requirement:

> Requirement
>
> The abstract representation of a document should be an instance of the conceptual representation corresponding to the category for that document, which includes the syntactic and functional structure introduced by the author.

[23] Tree document models are a sub-category of graphs.

[24] To distinguish between 'semantic structure' that refers to meaning and 'semantic structure in functional terms' that refers to functional interrelationships between text constituents, determined by meaning but not actually representing it explicitly, we shall use the term 'functional structure' for the latter.

The issue of functional structure of text has been pursued mostly by researchers from the Computational Linguistics community. A major theory that emerged from such work is Rhetorical Structure Theory (RST) and its proponents (Mann & Thompson 1987; 1988) put it forward as a unifying framework applicable to virtually any natural text of any size. The main characteristics of RST as a descriptive framework for text are:

(i) It describes relations between parts of text in functional terms, whether such relations are grammatically signalled or otherwise.

(ii) It identifies hierarchical structure in text.

(iii) Its scope is written monologue and it is insensitive to text size.

The purpose of an RST analysis of a given piece of text is to evaluate its coherence and thus determine how persuasive it is in an argumentation context. The analysis is conducted by identifying discourse relations between text spans (that is, uninterrupted linear intervals of text). A number of relations that can obtain between text spans have been identified by Mann & Thompson and are summarised in Table 4–1.

- Circumstance
- Solutionhood
- Elaboration
- Background
- Enablement and Motivation
    - Enablement
    - Motivation
- Evidence and Justify
    - Evidence
    - Justify
- Relations of Cause
    - Volitional Cause
    - Non-Volitional Cause
    - Volitional Result
    - Non-Volitional Result
    - Purpose
- Antithesis and Concession
    - Antithesis
    - Concession
- Condition and Otherwise
    - Condition
    - Otherwise
- Interpretation and Evaluation
    - Interpretation
    - Evaluation
- Restatement and Summary
    - Restatement
    - Summary
- Other Relations
    - Sequence
    - Contrast

Table 4-1 Organization of discourse relations for RST

The outcome of an RST analysis of a given piece of text is a hierarchical tree structure whose leaves, taken from left to right, correspond to text spans in the order in which they appear in the original entire text. The following figure illustrates such a hierarchical structure where numbered items correspond to text spans taken from the opening paragraph of Karl Marx's

Capital[25]. The text was randomly chosen and text spans have been numbered for ease of reference:

1. The wealth of societies in which the capitalist method of production prevails, takes the
2. form of an "immense accumulation of commodities",
3. wherein individual commodities are the elementary units.
4. Our investigation must therefore begin with an analysis of the commodity.
5. A commodity is primarily an external object,
6. a thing whose qualities enable it, in one way or another, to satisfy human wants.
7. The nature of these wants, whether for instance they arise in the stomach or the imagination, does not affect the matter.
8. Nor are we here concerned with the question, how the thing satisfies human want, whether directly as a means of subsistence(that is to say, as an object of enjoyment), or indirectly as a means of production.

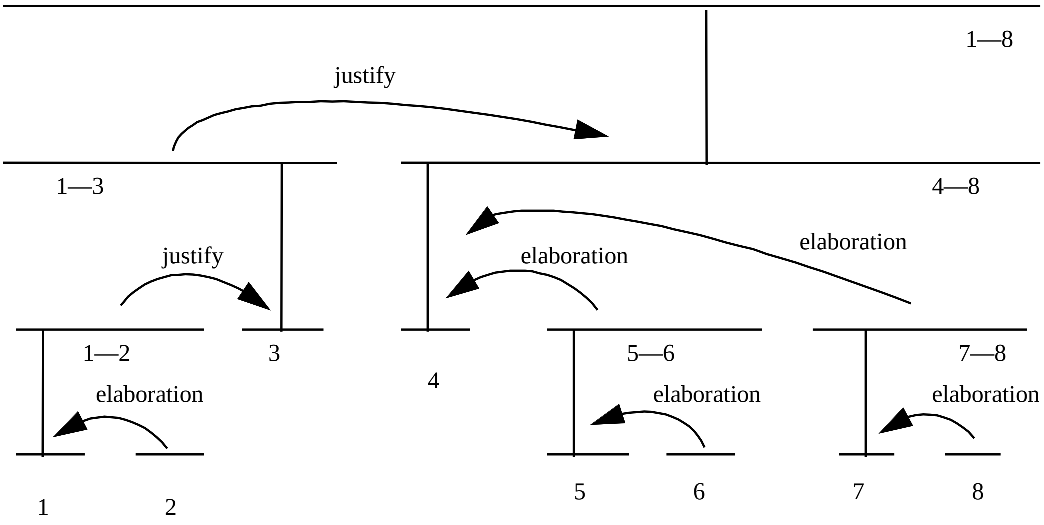


Figure 4–1 RST analysis of Marx example (Daskalopulu & Reed 1998)

RST suffers from a variety of problems that are explained in detail elsewhere (cf. (Daskalopulu & Reed 1998); (Reed 1998)). The most relevant of these problems to the work presented here is that, as RST's proponents acknowledge, although it has been applied successfully to a wide variety of documents, it does not provide a satisfactory account of contractual text (Mann & Thompson 1987, p. 265). The reasons for this are not documented by Mann & Thompson, though. Consequently, an experimental RST analysis of contractual text was conducted (cf. (Daskalopulu & Reed 1998)), which indicated some of the problems.

[25] Karl Marx, Capital, vol. 1, J. M. Dent & Sons Ltd.

The analysis was conducted on an extract from an agreement on arbitration[26] shown below, where individual text spans have been labelled for convenience:

1.1. The arbitral tribunal shall be composed of three members,
1.2. one to be appointed by each party
1.3. and the third member, who shall act as president,
1.4. to be appointed by the <appointing authority>.
2.1. The member of the tribunal appointed by the first party shall be <name and address>
2.2. The member appointed by the second party shall be <name and address>.
3.1. If at any time a vacancy shall occur on the Tribunal
3.2. by reason of the death, resignation, or incapacity for more than 60 days of any member, or for anyother reason,
3.3. such vacancy shall be filled as soon as possible
3.4. in the same manner as the original appointment of that position.

The RST structure produced for this extract is illustrated below:

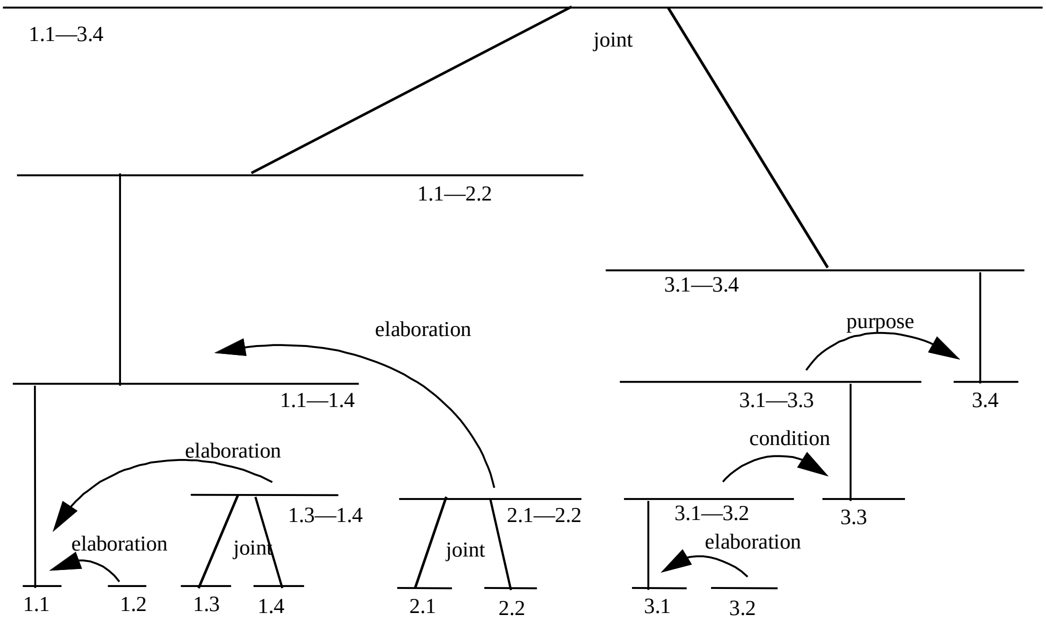


Figure 4–2 RST analysis of extract from agreement on arbitration

The underlying assumptions of RST result in hierarchical tree structures where each text span participates in one relation only. Hence, each text span has a unique functional role within a single RST analysis. Syntactically contractual documents are indeed structured in a tree-like manner but semantically, when one concentrates on functional relations between constituent

[26] Model Business Contracts, Croner Publications Ltd. 1988

parts, they yield graphs. Moreover, each constituent part (for example a single clause) may have multiple functional roles within the document (as we saw in chapter 2, a clause may be providing a definition and at the same time specifying a norm, or a procedure). More importantly, RST analysis relies heavily on making plausibility judgements about the writer's intention and the reader's comprehension given a piece of text. Multiple RST analyses for the same piece of text are therefore possible. For example, the text span comprising 1.3—1.4 could be regarded as providing BACKGROUND to 1.1. Similarly, 2.1—2.2 might have been regarded as being JOINT to 1.1—1.4. Moreover, the set of relations supported by RST is not necessarily complete, as Mann & Thompson point out. Should none of the defined relations be deemed satisfactory to account for the relationship between two text spans, it seems that the analyst may construct a new one, as long as the definition he provides conforms with the RST framework (by specifying all four of the fields for each relation[27]). The relations identified by Mann & Thompson provide a functional account of text when one considers its *argumentative* aspects, and Mann & Thompson make this clear from the outset. A different set of relations is required if we want to model semantic or pragmatic dependencies in contractual text that result in relations of exception, complement, necessary inclusion or exclusion between document constituents.

Given the foregone discussion and the requirement mentioned earlier, the appropriate model for contractual documents seems to be a directed graph, whose nodes correspond to text constituents at the desired level of detail and whose arcs are labelled to denote the nature of the relation between the nodes they connect. Some arcs denote containment relations between the corresponding nodes (and extracting them yields a tree structure corresponding to the syntactic form of the document). Others correspond to functional relations between nodes. A given pair of nodes may be connected by a single containment arc and multiple functional arcs. We shall refer to this model as Functional-Syntactic Document Model (FSDM) in what follows.

4.2.2 Legal Document Drafting

The best known early system for legal document drafting is Sprowl's ABF processor (Sprowl 1980) designed for the automated assembly of wills, tax returns, trusts and other standardised

[27] These fields are: Constraints on the Nucleus (that is, the text span which is at the source of the relation), Constraints on the Satellite (that is, the text span which is at the destination of the relation), Constraints on the combination of Nucleus and Satellite and the Effect of the relation. Examples of definitions for the relations supported in the framework developed by Mann & Thompson can be found in their (1987) and in (Daskalopulu & Reed 1998).

documents. The ABF Processor is based on Allen's (1957) proposal for a normalised language for legal drafting. Sprowl noted the similarity between an ambiguity-free language, such as the one proposed by Allen, and procedural programming languages and built a processor that is 'programmed' by feeding into it normalized statements. Sprowl extended Allen's normalized language by including mathematical statements of elementary algebra.

The processor holds a variety of standardised document templates. For example, Sprowl (1980) presents the following extract from a will that is produced by the processor:

```
                        LAST WILL AND TESTAMENT
                                  OF
                       [the name of the testator]
I, [the name of the testator], of [the city and state where the
testator lives], do make this my last will and testament.
I leave all of my personal effects and household goods to my [the
testator's spouse, a husband or wife], [the spouse's name], if [he
or she, the spouse's subjective pronoun] survives me, and
otherwise to my children who survive me in equal shares.
```

Bracketed placeholders, when processed, generate questions by prefixing them with the phrase "what is". Hence the questions generated for the example above are:

```
What is the name of the testator?
What is the city and state where the testator lives?
What is the testator's spouse, a husband or a wife?
What is the spouse's name?
What is he or she, the spouse's subjective pronoun?
```

The answers provided by the user of the processor (a para-legal or attorney) replace the bracketed place holders in the original text to yield the complete will.

In a similar manner, by appropriately labelling bracketed place holders questions are generated for numerical data. The same technique is applied in the case of optional or alternative passages. For instance the following passage, taken by Sprowl (1980), is optional in that its inclusion in the final will depends on a condition.

```
[IF the death taxes ARE to be paid out of the res of the estate
INSERT]
I instruct my executor to pay all applicable death taxes out of
the res of my estate so that the legatees and devisees do not have
to pay any taxes out or their individual gifts.
[ENDIF]
```

When the first bracketed placeholder is processed, a question is generated.

```
Are the death taxes to be paid out of the res of the estate?
```

An affirmative results in the passage to be inserted in the will and the bracketed placeholders to be removed. The processor operates on a similar vocabulary for alternative passages, lists and repetitive passages.

Sprowl's approach is procedural and comparatively low-level, since in effect he provides a special-purpose imperative programming language for writing programs, which can generate certain types of documents. Fragments of legal documents, which are subject to change, depending on specific data, are identified and the text is encoded with suitable procedures for filling in the values of these parameters. To program the processor an attorney or para-legal must identify suitable placeholders. The system through basic parsing operations identifies them in the text and generates questions in the simple manner we saw earlier. It then proceeds to act on the user's answers based on pre-programmed instructions that guide the substitution of placeholders with user answers, the insertion of passages or the removal of passages.

The author of a legal document supported by the ABF processor has little control over the document that he is drafting. Once he has chosen the type of document that he wants to draft, he can merely specify answers to questions raised by the presence of placeholders. In other words, the author of a document is operating on its abstract representation (its computer model) rather than on its conceptual representation (its logical structure model from his perspective). There does not seem to be any specific underlying model of the syntactic or functional structure of various documents, and different instances of the same document type are represented separately rather than as instances of the same category.

Another well-known document assembly system is Scrivener, described by Lauritsen (1992) as "an expert system shell with query-the-user facilities [...] specifically adapted for text generation" and developed by Evans (1990) with a view to exploit some of the advantages of a declarative expert system approach. The system has been under continuous development and its current version is marketed by the Dianoetic Development Company. The underlying model of the system is a hierarchical organisation of nodes that correspond to text constituents. Each node has associated "conditions that determine its inclusion in the document that is being drafted" (Lauritsen 1992). The term 'conditions' is somewhat misleading in this case, so we shall refer to them as 'variables' here, since they are effectively similar to ABF placeholders that admit specific values obtained through querying the user. If these variables are bound successfully to specific values provided by the user, then the corresponding boilerplate text including the relevant values is inserted in the document that is being drafted. The variables associated with text constituents are combined using logical connectives into rules which are invoked during a consultation session in order to determine variables that are not bound to values, so that appropriate questions may be generated to elicit answers from the user.

Scrivener offers users more flexibility than the ABF processor (for instance it can provide 'why'-explanations) and the representation of mechanisms that manipulate documents is not entirely procedural. Its underlying document model represents explicitly containment relations between text constituents and their 'logical' grouping into themes. However, as in the ABF processor, mechanisms for inserting values for specific parameters and conditions for the inclusion of specific document constituents are incorporated in the text itself, along with interface commands such as ASK, LET, DISPLAY and so on.

There is a growing number of commercially available legal document drafting systems. Unfortunately there is little or no documentation about their underlying assumptions and document models, although Lauritsen & Soudakoff[28] provide succinct reviews of them in *Law Office Computing Online*. What emerges through their reviews is that most of these systems are in spirit similar to Scrivener. Some of these systems, though not the majority, support two separate views, one for programming document models, the other for drafting a document[29]. In the programming view users can set up document templates and specify placeholders, valid values for them and appropriate queries (textual or menu-driven) for retrieving them. In the drafting view users assemble specific documents by selecting document components such as paragraphs and answering queries to instantiate variables. The drafting view in some systems also allows users to carry out formatting work by incorporating standard word-processing facilities or by invoking a word-processor. Most of the commercially available systems require some training, especially as regards the programming view, if they offer it.

The main facilities that are offered by most commercially available document drafting systems (in drafting view) are, in summary:

(i) Selection of document components.

(ii) Instantiation of variables.

(iii) Formatting work.

Lauritsen (1992) notes that of those it is the first that is the most challenging, that is, deciding what components to include in a document and in what form so that it performs its intended function. In other words, the interest lies in designing a document at the *macro-level*, with

[28] http://www.lawofficecomputing.com/jan99/Reviewsdata/shoppersguides

[29] It is hardly surprising of course, that most systems do not offer the programming view but rather require the software provider to customise and extend the system appropriately for each individual user.

emphasis on its structure and overall coherence, and at the *micro-level*, with emphasis on the formulation of detailed contents for each individual component. Legal document drafting at the macro-level can be viewed as a configuration process with characteristics similar to computer-aided design (CAD). This idea seems to underlie existing document drafting systems, as the drafter uses pre-prepared blocks of text to construct a document of a particular kind, much in the same way that a graphics designer uses basic geometric shapes to construct a picture. The idea is not novel and in fact has been articulated by other researchers (for example (Fiedler 1985), (Gordon 1992)). Its full potential has not however been exploited and it is to this end that this research contributes.

Specifically, we regard the process of assembling a document from components as subject to constraints. In existing systems, as we saw, these constraints concern essentially the provision of specific data by the user. It is the user's answers to questions (what the value of a placeholder should be, whether the user requires this or that passage amongst alternatives) that determines whether a pre-prepared block of text is included in the document. These are not however the only relevant constraints for guiding the assembly process. Functional relations between document components (for example, whether one precludes the other, or necessitates the presence of the other in order to maintain coherence) give rise to a richer set of constraints. An explicit model of the functional structure of documents (or more precisely of categories of documents) such as FSDM is therefore much more useful. Separating it from the actual text of the document is desirable in order to enable document component re-use and update and to provide a generic framework for document drafting. This is the underlying hypothesis of MODELLER, a prototype system for legal contract drafting, presented in the rest of this chapter.

4.3 MODELLER : A Solution for Legal Contract Drafting at the Macro-level

The overall architecture of MODELLER is illustrated in Figure 4–3.

Document categories are represented in terms of their syntactic and functional structure as generic documents. MODELLER instantiates such structures to produce individual document instances through an interactive session with the drafter. Document instances, as we shall see in a moment, contain collections of indices to actual text files for their individual components; the latter are held separately in memory and are accessed when the actual text of a particular document instance is required. The main features of the approach are:

(i) The representation addresses the structure and the interrelationships between the constituent parts of contractual documents and not the text of such documents itself.

(ii) The representation of documents is separated from the mechanisms used to manipulate them.

(iii) The drafting process is subject to a collection of explicitly stated constraints that govern the structure of documents.

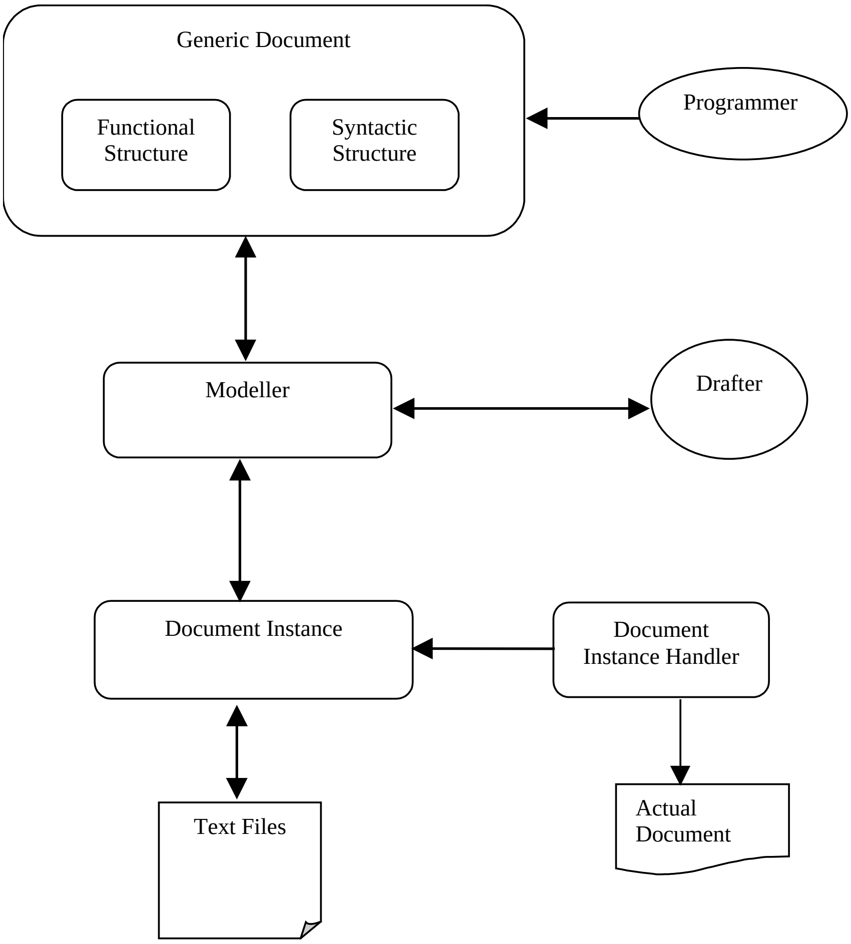


Figure 4–3 The MODELLER architecture

4.3.1 The Drafting Process

As we saw in the previous section, all currently available systems for document drafting operate on standard forms. The development of MODELLER was aimed at engineering contracts, which are in practice frequently drafted on the basis of model-forms, often issued by the relevant professional bodies. For example, many of the contracts that we have been examining are based on model-form contracts published by the Institution of Electrical Engineers (e.g. (IEE 1988; 1991)). Such model contracts have been developed over a considerable period—the first edition of (IEE 1988) was published in 1903. An important feature of these model-form contracts is that they are often accompanied by a detailed commentary, which explains the role of each individual provision in the document, its history and its overall effect. Where model-form contracts are available, they provide a natural starting point for a drafting system. However model-form contracts are not essential; where they are not available any previous document instance will do, although it is obviously most useful if some kind of commentary can be supplied for it.

To create a new document instance the drafter is provided with a model contract—a standard model-form contract or a previous contract of a similar type. Apart from changes in specific data values—or 'parameters'—many of the provisions will be acceptable in the original form. However, there will also be sub-units or passages of the document which do not suit the circumstances at hand and which require some modification. In one example we have examined, section 4-1 (Precedence of Documents) of the model-form contract (IEE 1988) reads:

> Unless otherwise provided in the Contract the Conditions as amended by the Letter of Acceptance shall prevail over any other document forming part of the Contract and in the case of conflict between the General Conditions the Special Conditions shall prevail. Subject thereto the Specification shall prevail over any other document forming part of the Contract.

However, in the actual contract, a different text had been included in this section:

> The documents forming the Contract are to be taken as mutually explanatory of one another and in the case of ambiguities or discrepancies the same shall be explained and adjusted by the Engineer who shall thereupon issue to the Contractor appropriate instructions in writing.

In another example, section 14-6 (Rate of Progress) of the model-form contract (IEE 1988), originally reads:

> The Engineer shall notify the Contractor if the Engineer decides that the rate of progress of the Works or of any section is too slow to meet the Time for Completion and that this is not due to a circumstance for which the Contractor is entitled to an extension of time under Sub-Clause 33-1. [Emphasis added]

In the actual contract, this had been modified and the occurrences of 'shall' and 'decides' were replaced by 'may' and 'considers' respectively. The point is that in neither case is there any indication as to why the modified version had been preferred over the original wording.

We want to provide a system in which the reasons for such modifications are recorded so that subsequent users can make informed choices about which version to select. We also want to allow users to create their own versions of sub-units where none of the existing ones is appropriate, and we want to encourage them to provide a commentary explaining the nature of the modification and the reasons for which it was made. Subsequent users will then be provided with several alternative versions—the original and the ones preferred in previously drafted documents—together with the accompanying commentary.

This collection of alternative versions is the core of what we call a *generic document* for contracts of a given type. The model-form contract should not be confused with the generic document. A model-form contract may be used to construct the initial generic document, but thereafter the generic document grows as new document instances, containing new versions of sub-units, are created.

4.3.2 A Model for Generic Documents

So far we defined a generic document for a given document category as a collection of components each of which may correspond to a number of different textual versions. We have not yet discussed the syntactic and functional structure of the generic document.

Even if a contract does not contain explicit divisions into sub-agreements, we find it convenient to divide it into separate sub-agreements (or ‘parts’ for short). Each part collects a number of related sections dealing with some aspect of the whole. Some of these parts are *compulsory* in the sense that every document instance of a given document category must contain provisions dealing with their corresponding aspects, otherwise it falls short of its intended purpose. Other parts are *optional*, in the sense that the drafter may choose to include or omit them, depending on the circumstances at hand, without compromising the functional effect of the contract. For example, the following lists shows the parts of a model-form contract (IEE 1988)[30], where compulsory parts are in bold, and all parts have been numbered for ease of reference later on:

[30] The model-form contract also contains a number of appendices which are omitted here for simplicity.

1	**Definitions and Interpretations**
2	**Engineer and Engineer's Representative**
3	Assignment and Sub-Contracting
4	Precedence of Documents
5	**Basis of Tender and Contract Price**
6	Changes in Cost
7	**Purchaser's General Obligations**
8	**Contractor's Obligations**
9	**Suspension of Work, Delivery or Erection**
10	Variations
11	Defects Liability
12	**Tests on Completion**
13	Taking Over
14	Performance Tests
15	**Certificates and Payments**
16	Accidents and Damage
17	**Force Majeure**
18	Insurance
19	Disputes and Arbitration
20	**Time for Completion**

Table 4-2 Optional and compulsory parts of a model form contract

One might ask what determines whether a part is compulsory or optional. Is it a legal requirement or perhaps common practice for one of the contracting parties? Is it simply an idiosyncrasy of an individual drafter? Our approach treats compulsory parts in the same manner, without distinguishing between specific reasons, although the associated commentary might provide some indication to the user of such reasons.

Each part is further subdivided, usually—but not necessarily—to a level corresponding to a 'section' of the document. And while sections have turned out to be the appropriate building blocks for the contracts that we have been examining, they can in principle be further subdivided to the required level of detail. Sections can themselves be compulsory or optional in the same spirit as parts.

The syntactic structure of a given contract category in terms of containment relations yields a tree structure whose leaf-nodes are pointers to actual text fragments stored separately.

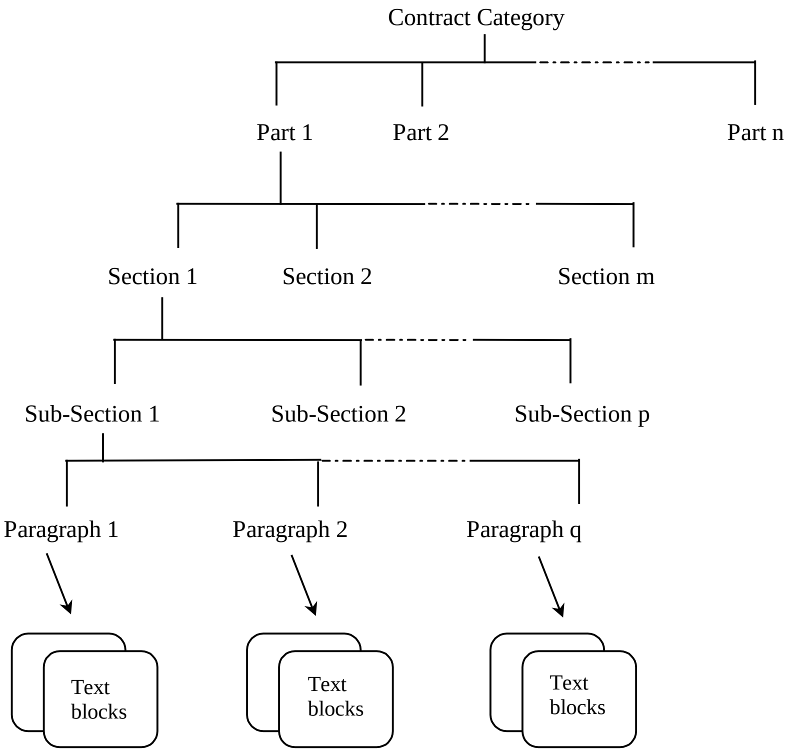


Figure 4–4 The hierarchical structure of generic contractual documents

A natural question for such representations is what document unit to take as the basic building block. For the sample contracts that we have examined, sections seem to be the most appropriate units generally, although a feature of our approach is that we do not need to commit to any particular choice of unit. In fact, different parts of the same document category can be, and often are, represented at different levels of detail. In some cases, it is unnecessary to subdivide the representation even to the level of individual sections; elsewhere, sections may be structured into sub-sections, sub-sections to paragraphs and sub-paragraphs and those in turn to individual sentences. In principle, individual sentences could also be divided into phrases and other fragments, in the spirit of syntactic decomposition for natural language text (for instance by distinguishing noun and verb phrases, and analysing these in terms of their grammatical components). Such a refinement, although not difficult to perform, was not attempted in MODELLER, mainly because, as we shall see in a moment, we are interested in relations between document sub-components that reveal their functional role within a given contractual document. Sentence fragments, such as noun and verb phrases, do not have individual functional roles of the kind we are interested in. Decomposing sentences into their constituent parts would entail that the tips of the document structure (cf. Figure 4–4) would point to files containing the text of those constituent parts. Reconstructing the text of the whole document from such structures would then require additional text processing, for example a mechanism for interleaving sentence fragments (for example (McKeown 1985); (Hovy 1993)), that would dominate the development of the prototype unjustifiably.

Apart from containment relations between document components, the generic document for a given contract category also contains other interdependencies between them, which are used to constrain the drafting process. A major relation between document components in contracts, and in legislation in general, is cross-reference.

The experimental material used for this research gave rise to three major kinds of constraints:

(i) Constraints relating sub-units of the same document.

(ii) Constraints relating data items or parameters.

(iii) Constraints relating data items or parameters with sub-units of the document.

4.3.2.1 Constraints relating sub-units of the same document

Cross-references between document sub-units are represented by means of logical assertions of the form:

`refers(DocType, section`$_i$`, section`$_j$`).`

The first argument ranges over document categories and is merely used to link particular cross-reference information to specific categories. The second and third arguments are identifiers for the units of text of the given document category.

In the generic document, we distinguished between optional and compulsory parts. It may be the case that two parts A and B are both optional, yet there may be a constraint that if A is included in the document being drafted, then B becomes compulsory, in the sense that it should also be included in the document. For example, as we saw in Table 4–2, Sub-Contracting and Assignment and Contractor’s Liability are both optional for the contract category defined on the basis of (IEE 1991). Yet, if Sub-Contracting and Assignment is included in a document of this category, then the part on Sub-Contractor’s Liability should be included as well. Such a constraint between document components is represented by means of logical assertions of the form:

```
forces(DocType, A, B).
```

The intended interpretation of this is “if document unit A is included, then document unit B must also be included”.

Another similar relations that may obtain between document components is incompatibility, that is, “both A and B cannot simultaneously appear in a document”, or alternatively “if document unit A is included, then document unit B must not be included”. This is represented by logical assertions of the form:

```
incompatible(DocType, A, B).
```

To express a third kind of relation, ‘exclusive disjunction’, that is, “exactly one of A and B must be included in the document”, logical assertions of the following form are used:

```
exclusive_or(DocType, A, B).
```

These three kinds of relationships between document units (`forces`, `incompatible`, `exclusive_or`) do not exhaust all the logical possibilities but we have not encountered examples where a more complex language would be necessary. This simple language is quite expressive already, since the specification of constraints may be subject to further conditions which could be expressed as Horn-clauses of the form:

`forces(DocType, A, B) :-` *`<further conditions>`*`.`

(and likewise for , `incompatible` and `exclusive_or`).

Moreover, this simple language is sufficient to capture relations between document sub-units that emerge as a result of the particular drafting principles employed within an organisation. For instance, as was mentioned in chapter 2, the violation of a given contractual provision may have various consequences, depending on whether it concerns a warranty or a promissory condition. But very rarely are contractual clauses explicitly labelled as one or the other type. Sometimes such information can be inferred from the sanctions that are associated with a particular clause. But very often, contractual documents do not specify any sanctions for the majority of contractual obligations, as parties prefer to defer such decisions until an actual violation arises. However, it might be the case that in a given organisation good drafting practice demands that all contractual clauses stipulating obligations be associated with some description (even a vague one) of corresponding sanctions. Hence, when the drafter includes a document unit that concerns obligations then additional document units describing their associated sanctions should be included in the document instance. This could be captured using `forces` relations with associated further conditions, as above.

4.3.2.2 Constraints between data items or parameters

A simple example of such a constraint is a provision of the form “if work is suspended for more than *three* months, then payment is suspended for more than *six* months” where the value of the first data item (duration of suspension of work) determines the value of the second data item (duration of suspension of payment). These constraints are specific to particular categories of contracts and are comparatively rare. More common are constraints between data items applying to contracts in general, irrespective of category. An example is the requirement that contracting parties must not be identical. When operationalized in a drafting system, such constraints essentially audit the drafter’s input to queries for values for data parameters.

4.3.2.3 Constraints relating data or parameters to sub-units of the document

These constraints are used to deal with the cases where values of various data items can affect the contents of the document. An example from (IEE 1991) is a requirement that if the party who supplies the service operates from outside the UK, then the document must include provisions stating arrangements for payment in foreign currency, and provisions stating which law (UK or non-UK) governs the contract. Constraints of this kind are expressed in the same manner as constraints linking sub-units of the same document, for instance through logical assertions of the form:

```
forces(DocType, Data, DocUnit).
```

The structure of the generic document, in syntactic (containment relations) and functional (constraints) terms is essentially a graph with labelled arcs, such as the extract shown in Figure 4–5, where shaded nodes correspond to compulsory parts and unshaded nodes correspond to optional parts. The part numbers used in Table 4–2 are used to label nodes for convenience. Arcs are labelled according to the syntactic or functional relation that obtains between the nodes they connect. For instance, the presence of part 6 (Changes in Costs) necessitates (forces) the presence of part 10 (Variations). If one traces only containment arcs, a tree structure corresponding to the syntactic organisation of the document is obtained, such as the one illustrated in Figure 4–4 earlier.

Such a structure is instantiated to produce a particular document of this category. All compulsory components and the constraints associated with them are inherited by the document instance. Optional components and the functional relations to which they participate may be included, depending on the drafter's choice and their compatibility with constraints.

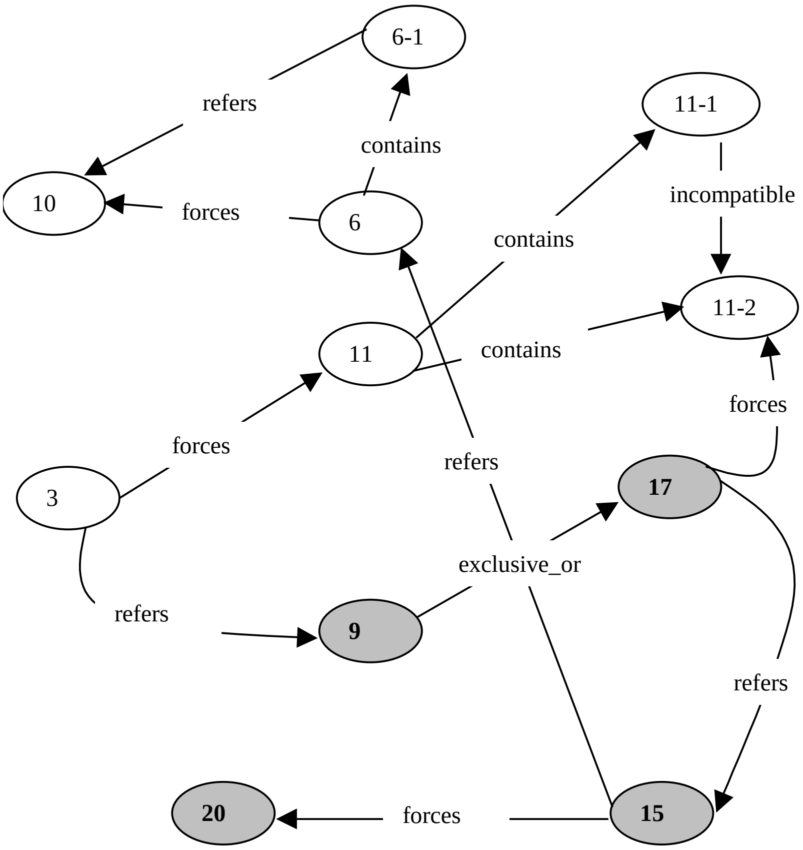


Figure 4–5 An extract of the FSDM model for (IEE 1988)

4.3.3 Representation of Generic Documents and Document Instances

In MODELLER, generic documents are represented as logic databases, by means of Horn-clauses. The following is an example of the representation for (IEE 1988). It does not contain the representation of constraints as this was shown in the previous section:

```
generic('IEE MF/1').

document_parameters('IEE MF/1',[$Engineer]).

part('IEE MF/1', 'Definitions and Interpretations',c).
part('IEE MF/1','Assignment and Sub-Contracting',o).
                     :
                     :
part('IEE MF/1', 'Time for Completion',c).

section('IEE MF/1', 'Time for Completion',
                     'Extension of Time for Completion',          1,[]).
section('IEE MF/1', 'Time for Completion',
                          'Delays by Sub-Contractors',            2, []).
```

```
                    :
                    :
text_file(‘IEE MF/1’,
      sect(‘Time for Completion’,
            ’Extension of Time for Completion’),
      1,[], tf1).
text_file(‘IEE MF/1’,
      sect(‘Time for Completion’,
            ’Extension of Time for Completion’),
      2,[], tf2).
```

(This is standard Prolog syntax. The layout is simply to aid readability).

The predicate `generic` records that the type ‘IEE MF/2’ is a generic document category. The predicate `document_parameters` associates document types with parameters that are specific to them. In the example a value for `$Engineer` will be required when a document instance of this type is drafted. The actual IEE MF/2 requires a number of other parameters besides `$Engineer` which have been omitted here for simplicity. Values for parties and a date for the agreement are common parameters for all contract categories and do not have to appear in the list of parameters specific to some document type. The assertions for the predicate `part` record the various parts that make up documents of a given document type and whether they are compulsory (`c`) or optional (`o`). The predicate `section` records the sections contained within each part of the given document type. It is through this predicate that the syntactic structure of the document category is derived. Identifiers for sections can be chosen arbitrarily but we tend to use the keywords, section titles or margin notes if they are present in the document. The fourth argument corresponds to the number of the section within a given part. This is not strictly necessary but it is convenient for generating the actual text of the document. The fifth argument is the list of parameters (in the excerpt given here, none) of the given section. Such parameters may be associated with any sub-unit of the document at any level of detail, but we have found that they are most useful when associated with the whole document and the basic building blocks of text (sections in our example). The predicate `text_file` associates atomic document units with their respective text files. Here `sect` is a function symbol used to construct a document unit identifier. The third argument corresponds to the *version* of the given section, the fourth is its list of parameters and the last a pointer to the file containing the actual text for the given version.

This representation of the generic document is used by the drafting program to guide the creation of document instances. Each document instance records:

(i) The category to which it belongs and its unique identifier for reference purposes.

(ii) Information about the contracting parties and the date on which the contract comes into effect.

(iii) Optionally, other data (depending on the document type) such as associated keywords, data values for various parameters and so on.

(iv) A list of pointers corresponding to the sub-units of the document (and indexing the actual textual content of those sub-units).

Each of the sub-units is also represented as a term with similar structure (type of document, identifier, data values if any and further subdivisions). Each document instance may also record additional information, which does not necessarily appear in the document but may be useful for other purposes. For example, details about the parties' contracting record, or keywords, which provide a simple mechanism by which users can search and retrieve documents, annotations by the drafter and general commentaries about his choices and the modifications he made.

An example of a document instance is as follows:

```
doc('IEE MF/2',
        'Q6',
       parties(...),
       date(...),
       [$Engineer = 'Frank'],
       [(sect('Definitions and Interpretations',
                       'Singular and Plural'), 1),
       <other sections>])
```

The first argument shows the document type; the second is the identifier by which the document instance is referred to by the system (this is unique and automatically constructed); the third argument carries details about the parties and the fourth about the date of the contract; the fifth is the list of parameters and their values for the given document type. The sixth is a list of provisions, where each is a pair of a section identifier and the corresponding *version* used in the particular document instance. Hence, the document instance carries information about which *version* of a document sub-unit was chosen at the time of drafting.

The representation scheme provides some flexibility. Consider the following example (IEE 1988) of section 38-1 ('Contractor's Equipment'):

> The Contractor shall within [30] days after the Letter of Acceptance provide to the Engineer a list of the Contractor's Equipment that the Contractor intends to use on the Site.

This can be represented in a number of different ways. Let §38-1 stand for the section identifier for readability. One view is that different values for “[30]” give rise to different versions of the section, in which case the representation would be:

```
text_file('IEE MF/2', §38-1, 1, [], tf1).
text_file('IEE MF/2', §38-1, 2, [], tf2).
```

Another possibility is to treat “[30]” as a *parameter* value, and in this case, our representation would take the form:

```
text_file('IEE MF/2',§38-1,1,[$days=30],tf1).
```

Consider the case where the original section had the following wording:

> The Contractor shall provide to the Engineer a list of the Contractor’s Equipment that the Contractor intends to use on the Site. Such list will be provided within [30] days after the Letter of Acceptance.

Then we could represent this section in a third way, as two sub-sections, one for each sentence. The first sentence would have one version while the second would have (a) alternative textual versions or (b) a parameter value, as in the example above. Note however that this last option is difficult if the original wording is maintained. In this case, we would need more complicated text-processing mechanisms for interleaving sentence fragments.

As noted already, the drafter can create his own wording for (atomic) sub-units of the document, and our system provides a simple text editor for this purpose. However, it is the drafter’s responsibility to ensure that the new text is meaningful and has the same properties as that which it replaces. The drafter is also allowed to extend the commentary and adjust the keywords corresponding to the section he modifies. However, if more dramatic modifications are required, which concern the structure of the document, then knowledge of the internal representation is demanded. Such modifications can be made but they are not supported by the drafting system at present.

The commentary, which is associated with the various sub-units, is recorded with the generic document and inherited by all instances. Consequently, as a user creates new versions with additional commentary this becomes available automatically to all previous document instances as well.

Keywords associated with each document instance can be adjusted by the user, and so, unlike commentary, they are recorded with the document *instance*. Keywords, which are stored with the generic document, are made available to the user during the drafting session as an initial suggestion. The system also provides a facility to record personal notes for each document instance.

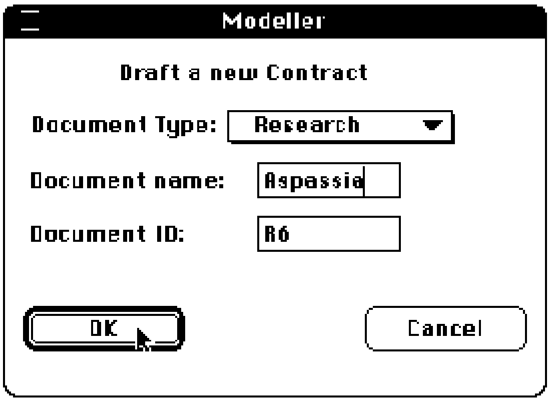
4.4 Implementation

This section provides a sample session of the drafting program and illustrates how the representation of generic documents and document instances is used. MODELLER is implemented in MacProlog. Prolog allows for efficient prototyping and is ideal for implementing the constraint-checking component. The MacProlog environment supplies a number of very useful primitives for the construction of a usable interface (LPA 1992).

Normally the drafting of a document is spread over several sessions but in what follows it is presented in its entirety as if it is done in one session. During the drafting session, the user provides specific data values and makes choices about the contents of the document, which he drafts. The drafting module uses the input provided by the user and the generic information that we have stored in order to construct a document instance, while checking it against the constraints that are imposed on the whole process.

A danger in drafting large documents is that the user can become disoriented in the detail and large number of steps that need to be taken. In order to impose some structure on the process we have adopted a specific order in which drafting is performed. The user is permitted to go back and modify or adjust previous choices. Constraint checking can be activated or de-activated during the session from a menu. Thus, the user can choose whether checking takes place in a step-by-step fashion and/or at the end of the drafting session before the instance is actually stored.

A user begins the drafting session by making a selection from the database of available generic documents. A unique identifier for the new document instance is automatically constructed; the user can supply his own choice of a name by which the new document will be referred. The following figure shows the initial screen of MODELLER:



Draft a new Contract

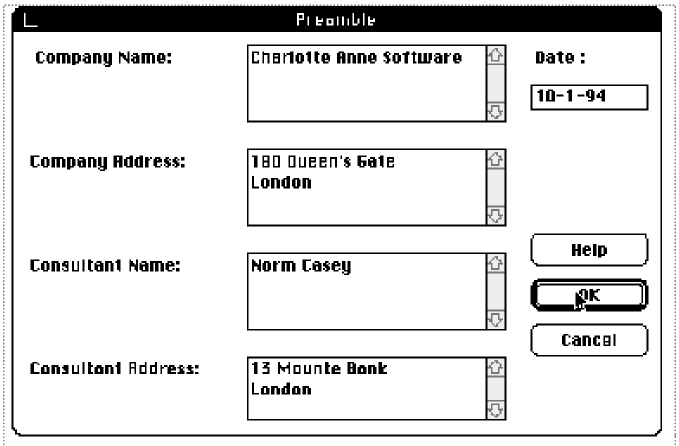
Document Type: Research

Document name: Aspassia

Document ID: R6

Figure 4–6 The initial screen of MODELLER

At some stage during the drafting (often but not necessarily at the beginning) the user is required to input information about the contracting parties (names, addresses) and the date the agreement is drafted.



Company Name: Charlotte Anne Software

Date : 10-1-94

Company Address: 180 Queen's Gate
London

Consultant Name: Norm Casey

Consultant Address: 13 Mounte Bank
London

Figure 4–7 The preamble screen of MODELLER

The associated help files simply explain what kind of information is required and where it typically appears in documents of this type. Some of the IEE model contracts also provide some relevant commentary, which is made available to the user in the same way. General constraints, such as that the contracting parties should not be identical, or that the starting date of the agreement cannot be earlier than the current date, limit the user's input.

Compulsory parts for the selected document type are presented in a menu. A user who is familiar with the contents of a part may choose to have it included automatically in the document, without going into it in detail.

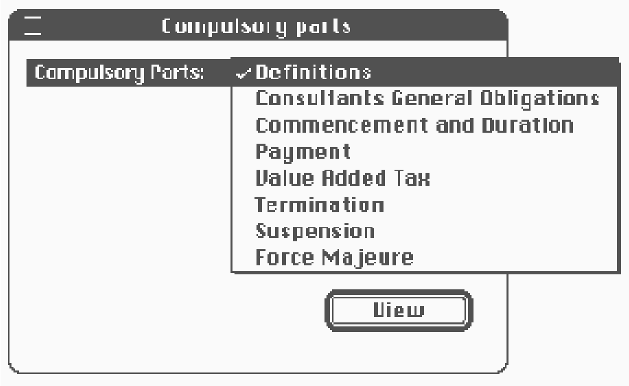
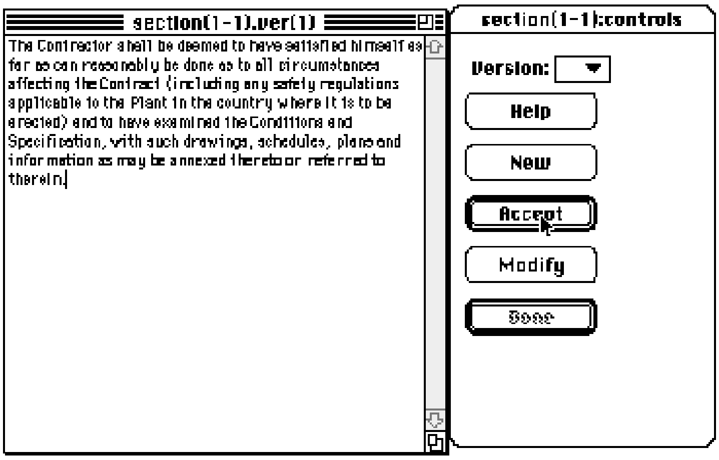


Figure 4–8 Selection of compulsory document parts in MODELLER

If the user prefers to view the contents of the selected part—or if a part chosen to be included automatically contains alternative versions or requires data values—the sections that it contains are presented one at a time, in a manner described below.



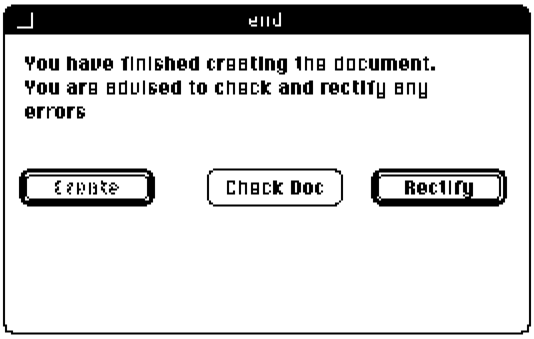
The Contractor shall be deemed to have satisfied himself as far as can reasonably be done as to all circumstances affecting the Contract (including any safety regulations applicable to the Plant in the country where it is to be erected) and to have examined the Conditions and Specification, with such drawings, schedules, plans and information as may be annexed thereto or referred to therein.

Figure 4–9 Section viewing in MODELLER

If there are alternative versions of the section, then these can be displayed and compared. The *Help* button accesses the associated commentary and notes. If the user accepts an existing version of the section, a note is made in dynamic memory and the session continues with the next section or the next part. If no existing version is satisfactory, the user can create his own, possibly by modifying an existing one. A simple text editor is provided for this purpose. As indicated in the previous sections this version of the system does not support more drastic modifications affecting the structure of the document, since this requires knowledge of the internal representation of documents or, in other words a *programming view*. The user can however change the order in which sections or parts appear in the document. The keywords associated with the section may also be modified.

Once compulsory parts have been dealt with, the user normally proceeds with optional parts in similar fashion, except that at this stage he has the opportunity to indicate that the contents of the document are now complete.

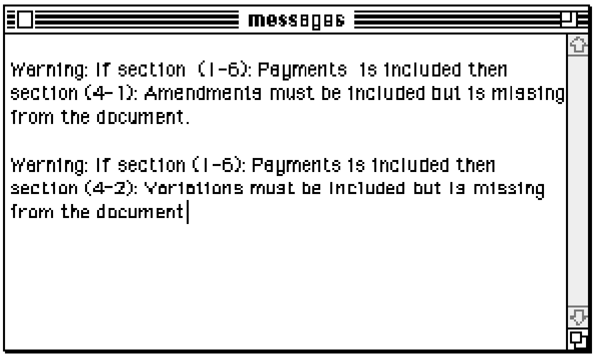
The user can check the document against the constraints at any point of the session or he can choose to have them checked automatically after every new entry.



You have finished creating the document.
You are advised to check and rectify any errors

Figure 4–10 Completion of assembly process in MODELLER

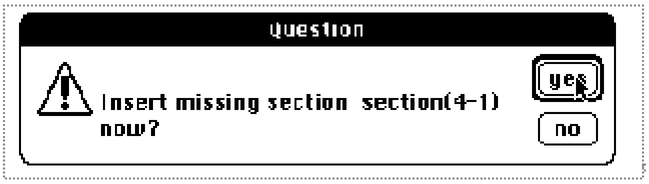
If violations occur, appropriate warning messages are displayed and remedial action is recommended.



Warning: If section (1-6): Payments is included then section (4-1): Amendments must be included but is missing from the document.

Warning: If section (1-6): Payments is included then section (4-2): Variations must be included but is missing from the document

Figure 4–11 Constraint violation messages in MODELLER



Insert missing section section(4-1) now?

Figure 4–12 Proposed remedial action for constraint violation in MODELLER

4.4.1 Retrieval of Document Instances

A common requirement in organisations that engage in contractual activity is the storage of contractual documents for reference purposes. Depending on the nature of the business exchange that such agreements regulate, contractual documents may have to be consulted on a regular basis for a variety of reasons. For example, our industrial collaborators from British Gas stated that their contractual documents are consulted for purposes such as the following:

(i) To determine particular values for parameters of the agreement (e.g. quantity of Gas to be delivered from a hydrocarbon field, amount of instalment, time period within which payment must be made in response to an invoice and so on).

(ii) To determine actions that the organisation may take at some given point in time (e.g. in relation to filing a *force majeure* claim, serving notifications to counter-parties of changes in arrangements, appointing arbitrators for the settlement of disputes and so on).

(iii) To determine whether actions taken by the counter-party comply with the agreement.

(iv) To update contractual documents where changes have been effected by external events or policy changes within the organisation (e.g. inflation effects on pricing information, changes of reference to legislation where new Acts of Parliament replace previous ones and so on).

(v) To compile information about counter-parties that is to be used for contractual or extra-contractual purposes (e.g. to notify all counter-parties in Scotland about *force majeure* events or to arrange exchange visits with all counter-parties in agreements whose value is above a certain level).

Although MODELLER is not directly aimed towards document retrieval, the representation scheme for generic documents and document instances enables retrieval of document instances. Queries can be formulated for the retrieval of whole document instances, parts of documents, sections or party information, as the following figure illustrates.

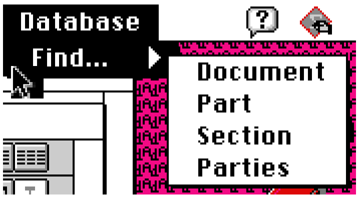


Figure 4–13 Document instances retrieval menu in MODELLER

Each menu option yields a query window through which the user specifies his query. The less information the user provides the more general the query is and consequently more answers are returned. Information that users can specify in their queries includes:

(i) the category of documents they are interested in;

(ii) the date on, before or after which the documents were drafted;

(iii) name and/or address information about the parties;

(iv) keywords and/or headings and/or status (compulsory or optional) for document parts;

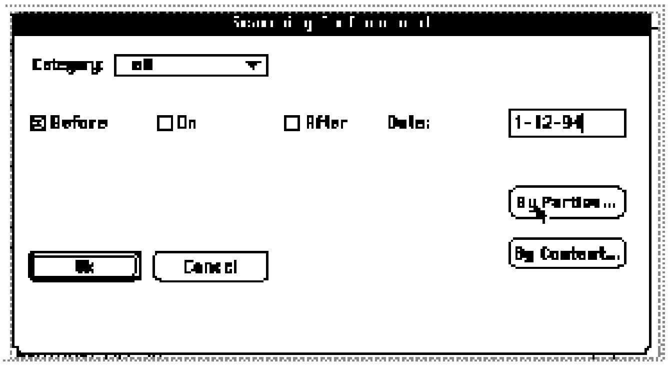
(v) keywords and/or version information for sections.

Hence example queries are:

- "Find all research contracts";
- "Find all research contracts drafted before January 1998";
- "Find all Gas purchase contracts with counter-parties in Scotland";
- "Find all consultancy contracts which contain version 2 of Payment terms";
- "Find all contract sections that refer to *force majeure* for any contract category";
- "Find all research contract parts whose keyword is 'liability' and which contain version 3 of Defects Liability terms and which were drafted after September 1992";

and so on for other combinations of the indexing information.

An example compound query is: "Find all contracts for research, which were drafted before December 1994, where the company has contracted with a party based in France, which contain version 3 of payment terms".

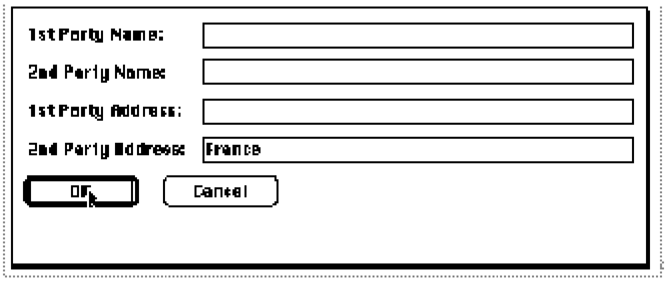


Category: all

☒ Before □ On □ After Date: 1-12-94

Figure 4–14 General retrieval query in MODELLER

The user supplies additional information about parties and contents by selecting the appropriate buttons:



1st Party Name:

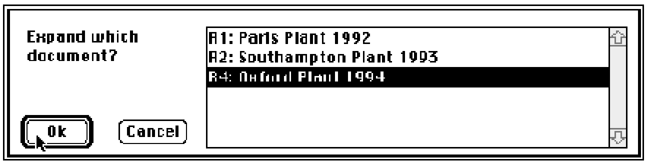
2nd Party Name:

1st Party Address:

2nd Party Address: France

Figure 4–15 Retrieval query in MODELLER

Documents that satisfy the requirements set out in the query are presented to the user as shown in the following picture and he can then select the ones he wishes to examine. Text files that correspond to the selected document(s) are accessed and manipulated by a separate module, allowing the actual text of the document to be viewed in its entirety.



Expand which document?

A1: Paris Plant 1992
A2: Southampton Plant 1993
A4: Oxford Plant 1994

Figure 4–16 Results of retrieval query in MODELLER

4.5 Evaluation

MODELLER was tried successfully on a variety of contracts of different sizes and complexity. Appendix A shows the complete representation of a generic document based on (IEE 1991). The clauses (predicate `text_file`) that associate atomic document components with text files (each file corresponding to different versions for that component) have been omitted, as they are self-explanatory. As was mentioned earlier the generic document grows as drafters create new versions of atomic document components and consequently more `text_file` assertions become available. Appendix B shows the detailed list of sample documents that were used as starting points in the construction of generic documents. For each, we provide an indication of its complexity in terms of the total number of atomic document components that were identified and the total number of constraints between them. Appendix B also shows a list of sample documents for which detailed representations were not actually constructed. The documents were however examined to establish that they, too, could be analysed in MODELLER terms.

In contrast to systems such as the ABF processor and Scrivener and their more recent variants, MODELLER separates the text of documents from the mechanisms that manipulate it. This allows for flexibility of the form of the actual output document. Hence actual output documents can be plain text, SGML ((Smith & Stutely 1988); (Bryan 1988)) marked-up or HTML marked-up. However, more importantly it enables drafters to have more control over the document that they are creating since in effect they are not restricted to merely re-using boilerplate text with appropriate instantiation of data parameters with specific values. Drafters may create new versions of document components (by modifying existing ones slightly or by creating them from scratch) and these become assimilated in generic documents and available for future use. Currently MODELLER does not provide any mechanism for ensuring that such new versions are indeed appropriate for the purposes the drafter has in mind, that the text he enters is correct or that any dependencies between new versions and other document components are specified. This would be part of the programming view of the system, which currently is possible only through direct manipulation of the generic document representation and is one of the aspects on which future work is required.

The validation of the prototype did not reveal any problems in terms of functionality and behaviour. MODELLER was also evaluated by our industrial collaborators at British Gas in order to establish its usefulness and potential improvements in the functionality it offers. This evaluation confirmed that the underlying concept of analysing documents in terms of their sub-components and the syntactic and functional interrelationships between them is useful and affords drafters a better global view of the document instance they create. The commentary associated with atomic document units was especially useful when teams of drafters rather than a single individual engaged in document instance creation since members of the team could trace each other's rationale for using a particular version of a document unit or for creating a new one. Automated constraint checking eliminated to some extent the need for paper 'checklists' that were routinely used. The retrieval facility was found useful especially in situations where drafters wanted to check whether instances of the same contract category with different counter-parties used particular versions of terms consistently (particularly in relation to payment and *force majeure* terms where it was desirable for the company to have streamlined processes). Our principal liaison with British Gas[31] commented that the Research and Development unit of the company were considering developing a commercial version of the prototype.

The evaluation brought up many ways in which MODELLER can be improved. Usability issues were not considered in detail and although the interface is adequate for evaluation purposes, a sophisticated version would require re-design and integration with commercial word-processors and hypertext browsers. For collaborative drafting, a distributed version of MODELLER would be preferable with appropriate locking and authentication mechanisms. The retrieval facility could be enhanced to allow full text search of the contents of document instances. More importantly, the programming view of MODELLER needs development to allow users to create new generic documents without directly manipulating the representation. As was mentioned previously, MODELLER in its current state does not check whether the text that drafters enter when they modify or creating new versions of document units is appropriate and mechanisms towards this would be desirable. This however would entail a representation of the contents of documents at the micro-level, which is beyond the scope of MODELLER. An obvious direction for improvement is therefore to expand the framework afforded by MODELLER with detailed representations of the contents of individual document sub-components. The next chapter discusses some possibilities in that direction.

4.6 Related Work

The MODELLER approach for document creation adopts a CAD-like perspective whereby a 'whole' is constructed from atomic parts as we saw earlier. In this light, the approach is

[31] John Piggott, Head of Mathematics and Computing Research and Development Unit, British Gas Plc., personal communication 22 April 1995.

similar in spirit to other projects outside the (legal) document domain, such as PRO*forma*, a tool for authoring clinical guidelines (cf. (Fox & Thomson 1998)), RaPiD, a knowledge-based tool for designing dental prostheses (cf. (Hammond *et al.* 1993)), and Design-A-Trial, a knowledge-based tool for designing clinical trials (cf. (Hammond *et al.* 1994)). The design of clinical guidelines, dental prostheses or clinical trials is negotiated between the corresponding system and the user: the user makes choices which the system checks against constraints and either accepts or reports as inappropriate. Constraints in these domains are not so much the product of functional relations between components of the 'whole' that is being constructed, as is the case in MODELLER. Rather they are the result of rules that define good practice (some of these rules exist explicitly as regulations or are elicited by experts in the corresponding domain). This is an interesting difference between the broader medical domain of such projects and the domain of legal contracts of MODELLER. In the latter, there are no explicit rules that define 'good contracts'. There are of course rules, as a product of English Law, that define valid contracts as we saw in chapter 2. However, what makes a contract effective or acceptable to a party is context-dependent, organisation-specific, and comes into play when one considers micro-level representation of contractual content.

In the domain of document drafting, the importance of modelling explicitly the functional as well as the syntactic structure of documents has been acknowledged by other researchers, following the first publication on MODELLER (Daskalopulu & Sergot 1995). An important strand of related work is that led by Branting and described in a series of papers ((Branting 1998); (Branting & Lester 1996); (Branting *et al.* 1997); (Branting *et al.* 1998)). Branting and his associates have implemented their ideas in a prototype system called DOCU-PLANNER. DOCU-PLANNER is influenced by MODELLER in the following respects:

(i) Rather than addressing individual documents it represents document genres, that is, document categories (or generic documents in MODELLER terms).

(ii) The creation of a document that belongs to a particular genre (a document instance in MODELLER terms) is viewed as a configuration task, subject to constraints, which are not restricted to containment relations between document components but rather reflect other dependencies between them as well. Such dependencies in DOCU-PLANNER are relevant to the performative roles of document components, that is, to the aim that they purport to achieve. This will be explained in more detail in a moment.

(iii) During the creation of a document instance, the performative role that a given document component plays is taken into account when the drafter decides whether to include it or not (such role is captured by the associated commentary in MODELLER terms).

However, whereas MODELLER does not represent what a document actually 'says', DOCU-PLANNER goes further into what we called micro-level drafting. Drawing upon ideas from research in natural language generation, argumentation and speech act theory, DOCU-PLANNER takes a discourse-based view of document drafting and models documents in terms of their illocutionary and rhetorical aspects. A brief digression is necessary here to explain these terms.

Stemming from speech act theory, 'illocutionary' refers to the action aspect of performative statements. As we saw briefly in chapter 2 (section 2.4.2), performative statements are those that do not merely describe some state of affairs truly or falsely (as does for example the statement "The cat is on the mat"). Instead issuing such statements is tantamount to performing an action, which has consequences. For example, the statement "I promise that money is on the table" is not merely describing that money is on the table. Its propositional content, which may be true or false, is indeed "money is on the table". The statement "I promise that money is on the table" cannot however be ascribed a truth-value. Instead it corresponds to an action on my part (the action of promising that a certain state of affairs holds) which has consequences such as creating an obligation on my part to keep that promise and ensure that money is indeed on the table, convincing the hearer that "money is on the table" is true, causing the hearer to rely on the truth of "money is on the table" and so on. Such consequences are the perlocutionary aspects of performative statements in speech act theory terms.

Performative utterances are analysed in the F(P) framework of speech act theory. P denotes the propositional content of the utterance and F denotes its illocutionary force (a propositional attitude towards the propositional content)[32]. The following table shows some examples of utterances that have the same propositional content but different illocutionary forces (1–3 and 4–7) and utterances that have the same illocutionary force but different propositional contents (1 and 4; 2 and 5).

[32] There is an infinite number of illocutionary forces (corresponding linguistically to performative verbs) but according to Searle (1969) they fall into five distinct types: assertives, commissives, directives, declaratives and expressives. Other frameworks exist in the literature however. Austin (1962) also identified five classes for performatives which are not exactly the same as Searle's.

	Utterance	Propositional Content	Illocutionary Force
1	"It rains"	"it rains"	Assertion
2	"Will it rain?"	"it rains"	Question
3	"It will rain"	"it rains"	Prediction
4	"Bob is the boss"	"Bob is the boss"	Assertion
5	"Is Bob the boss?"	"Bob is the boss"	Question
6	"I appoint Bob as the boss"	"Bob is the boss"	Declaration
7	"I suggest Bob as the boss"	"Bob is the boss"	Suggestion

Table 4-3 Examples of performative utterances in the F(P) framework

In Austin's (1962) terms, each performative utterance has a locutionary aspect (what is being said, the propositional content which has meaning), an illocutionary aspect (the force in saying something) and a perlocutionary aspect (what is achieved by saying something). Hence the perlocutionary aspect of an assertion might involve, among other things, convincing the hearer about what is asserted; the perlocutionary aspect of a promise would involve, among other things, creating an expectation on the hearer for the promise to be kept.

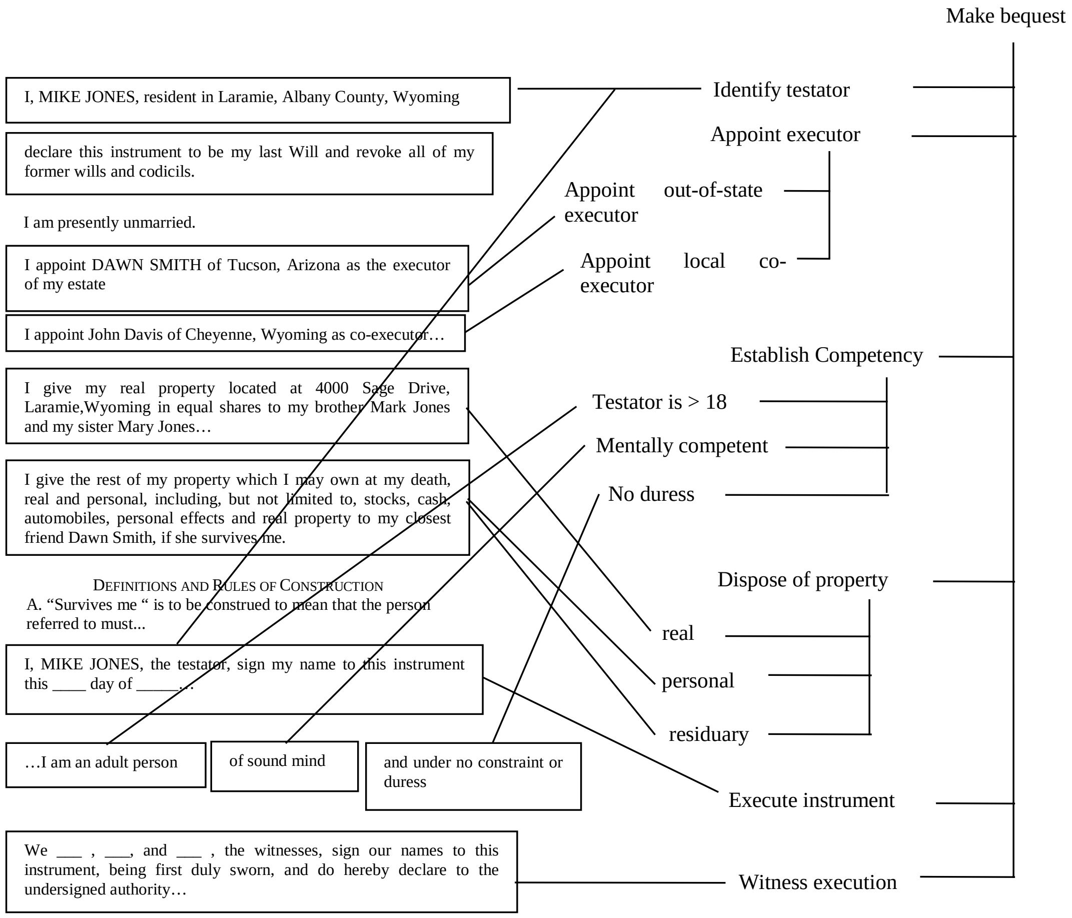
The term 'rhetorical' stems from linguistics and refers to functional relations between discourse components. We saw earlier in this chapter (section 4.2.1, Table 4–1) a number of such relations that have been identified within the RST framework, such as exemplification, sequence, generalization, elaboration and so on.

In DOCU-PLANNER documents are represented, according to Branting and his associates, in terms of their illocutionary structure (a collection of their illocutionary forces) and their rhetorical structure (a collection of the rhetorical relations that obtain between document components).

Figure 4–17 illustrates the illocutionary structure of a will. 'Identify testator', 'appoint executor', 'establish competency', 'dispose of property', 'execute instrument' and 'witness instrument' are the illocutionary sub-structures (an illocutionary force and a propositional content) corresponding to individual statements within the text. They are all specialisations of the overarching illocutionary structure 'make bequest'. Some of the illocutionary sub-structures are themselves decomposed in more specialised ones (for instance 'appoint out-of-state executor' and 'appoint local co-executor' emerge by decomposing the propositional content of 'appoint executor' and retaining the same illocutionary force). Some structures that appear diagrammatically as illocutionary are lacking their illocutionary operator. Hence 'testator is > 18', 'mentally competent' and 'no duress' are descriptive (true/false) rather than performative. Perhaps the intended reading is 'establish testator is > 18', 'establish [the testator is] mentally competent' and 'establish no duress' (in the same spirit that 'appoint executor' is decomposed). Similarly 'real', 'personal' and 'residuary', which come about by decomposing the propositional content of ‘dispose of property’, are lacking, as they stand, their illocutionary force and perhaps the intended one is ‘dispose’. Certain parts of the will are not ascribed any particular illocutionary structure (namely the statement “I am presently unmarried” and the Definitions part), perhaps because they are not essential when drafting wills.

What the example in Figure 4–17 indicates is that in DOCU-PLANNER statements are ascribed functional roles, not in relation to each other as was the case in MODELLER, but in terms of their contribution to the global goal of the document, which is to make a bequest. In other words, such analysis reveals the performative aspects of statements (what they do) within the document.

Figure 4–18 illustrates the rhetorical structure of a will. Branting & Lester (1996) note that the rhetorical structure of documents in DOCU-PLANNER captures both discourse relations and relations between document components that result from stylistic conventions in a particular document genre. Their example of the rhetorical structure of a will illustrates the latter, rather than discourse relations. Labels such as ‘Prologue’, ‘Family Description’ and so on correspond to document parts or sections in MODELLER terms and containment relations between such components are explicit in the structure. Perhaps the implicit discourse relation amongst such parts is ‘sequence’ (that is, they are supposed to appear in a certain order).



I, MIKE JONES, resident in Laramie, Albany County, Wyoming

declare this instrument to be my last Will and revoke all of my former wills and codicils.

I am presently unmarried.

I appoint DAWN SMITH of Tucson, Arizona as the executor of my estate

I appoint John Davis of Cheyenne, Wyoming as co-executor…

I give my real property located at 4000 Sage Drive, Laramie,Wyoming in equal shares to my brother Mark Jones and my sister Mary Jones…

I give the rest of my property which I may own at my death, real and personal, including, but not limited to, stocks, cash, automobiles, personal effects and real property to my closest friend Dawn Smith, if she survives me.

DEFINITIONS AND RULES OF CONSTRUCTION

A. "Survives me " is to be construed to mean that the person referred to must...

I, MIKE JONES, the testator, sign my name to this instrument this ____ day of _____…

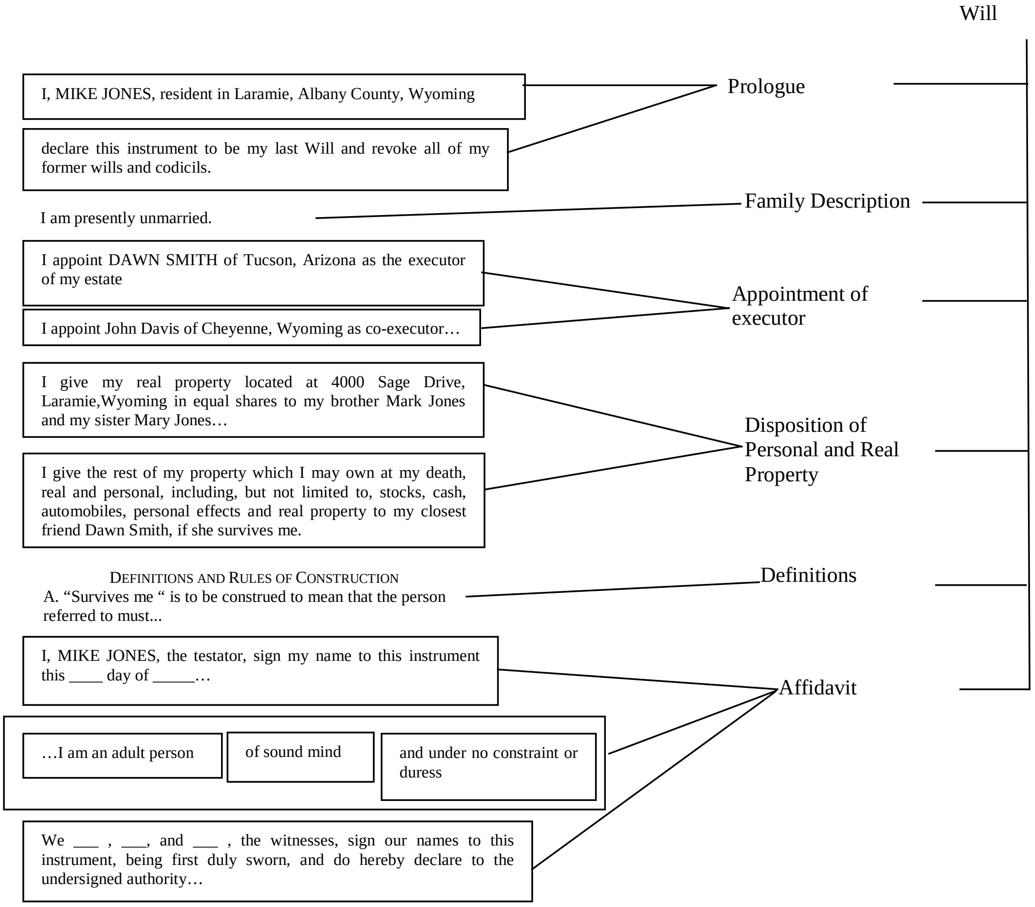
…I am an adult person

of sound mind

and under no constraint or duress

We ___ , ___, and ___ , the witnesses, sign our names to this instrument, being first duly sworn, and do hereby declare to the undersigned authority…

Figure 4–17 The illocutionary structure of a will (Branting & Lester 1996) [re-drawn]



I, MIKE JONES, resident in Laramie, Albany County, Wyoming

declare this instrument to be my last Will and revoke all of my former wills and codicils.

I am presently unmarried.

I appoint DAWN SMITH of Tucson, Arizona as the executor of my estate

I appoint John Davis of Cheyenne, Wyoming as co-executor…

I give my real property located at 4000 Sage Drive, Laramie,Wyoming in equal shares to my brother Mark Jones and my sister Mary Jones…

I give the rest of my property which I may own at my death, real and personal, including, but not limited to, stocks, cash, automobiles, personal effects and real property to my closest friend Dawn Smith, if she survives me.

DEFINITIONS AND RULES OF CONSTRUCTION
A. "Survives me " is to be construed to mean that the person referred to must...

I, MIKE JONES, the testator, sign my name to this instrument this ____ day of _____…

…I am an adult person

of sound mind

and under no constraint or duress

We ___ , ___, and ___ , the witnesses, sign our names to this instrument, being first duly sworn, and do hereby declare to the undersigned authority…

Figure 4–18 The rhetorical structure of a will (Branting & Lester 1996) [re-drawn]

The main categories of documents on which the DOCU-PLANNER approach has been tried are wills and show-cause orders. The latter are judicial documents that specify what is defective in a given appeal and order the appellant to rebut the defects within a specified period of time otherwise the appeal is dismissed (for illocutionary/rhetorical structures of them see (Branting *et al.* 1998)).

Creation of a new document in DOCU-PLANNER is conducted as follows: The drafter provides an explicit model of the goals that he intends the document to achieve (for instance in the case of show-cause orders, to establish the conditions for dismissal of the appeal and in the case of wills, to make a bequest), and the relevant case facts (for example the date of the court decision, the date of the filing of the appeal, the appellant's details and so on for show-cause orders, or the names of executors, the names of beneficiaries and so on for wills). The system then finds and instantiates a set of illocutionary operators that achieve the goals specified by the drafter and a set of rhetorical operators that correspond to the given document genre (such as show-cause orders or wills). It then proceeds to synthesize a document that fits the illocutionary and rhetorical operators by retrieving the text associated with them, putting it in the appropriate order and instantiating any parameters according to the case facts.

On a point of terminology employed in the DOCU-PLANNER approach there seems to be some confusion. A performative statement has a point (other than the act of actually making the statement), an effect that it intends to produce, and actual production of it is the perlocutionary effect of the statement. Thus, what is intended by "appointing an executor" is that certain powers be conferred upon the named individual (by virtue of the power held by the speaker of the performative statement). Asserting that certain property is to be disposed of in a certain way has the intended effect of issuing orders to the executors to see to it that such property is indeed disposed of in the specified way. Hence, what is achieved by such a statement is perhaps close to creating an obligation on the executors to bring about the specified state of affairs. It seems therefore more appropriate to say that when drafters specify goals that they wish the document at hand to achieve, they specify intended perlocutionary effects. These in turn are associated with performative statements that have certain illocutionary forces. It is not altogether clear whether associations of perlocutions with illocutions are underlying DOCU-PLANNER models, or whether illocutions and perlocutions are collapsed into the same thing (that is, whether there is an implicit notion of success associated with each performative statement—"I promise to pay you five pounds" becomes then associated with "you have five pounds", whether I actually kept my promise or not).

Illocutionary forces themselves are linguistically denoted by performative verbs, of which we saw some examples earlier. Speech act theorists like Searle (1969) and Austin (1962) have identified broad classes in which those illocutionary forces fall. The classifications themselves are a matter of debate (both Searle and Austin identify five classes but there are slight differences between them, see Appendix C for details) but what they seem to suggest is that each class corresponds to distinct intended effects. Hence performative verbs such as 'appoint', 'authorise', 'nominate', 'dismiss', 'order', 'bequeath' and so on, all distinct illocutionary forces, are *exercitives* in Austin's classification and *declaratives* in Searle's. In both cases, they are associated with the speaker's legal power (in the Hohfeldian sense) and as a consequence they confer powers, obligations, prohibitions and so on upon others. Hence, "I appoint X as the executor of my will" and "I nominate X as the executor of my will" essentially have the same intended effect, although they are different statements in terms of the performative verbs that they employ. This seems to suggest that characterising documents in terms of their intended perlocutionary effects rather than by individual performative verb might yield more general models, allowing for linguistically varied documents that have essentially the same effects.

By modelling explicitly the illocutionary structure of documents, DOCU-PLANNER can provide explanations to drafters about why a particular component is included in the document they create. In addition, documents and their parts can be retrieved by their performative aspects. In these respects DOCU-PLANNER's functionality is better than MODELLER's in which the commentary associated with document components—which contains explanations about the reasons for including a component and records its role and overall effect in a document—is unstructured text, not explicitly represented and consequently not usable for retrieval.

It is however not clear how the approach employed by DOCU-PLANNER could accommodate legal contracts. The types of documents which DOCU-PLANNER has addressed so far, that is, wills and show-cause orders do not appear to have any explicit prescriptive content, although they have legal consequences. Although it is possible to conceive of similar illocutionary structures for contractual documents (where illocutionary operators such as 'establish', 'order', and so on) it is harder to conceive of appropriate rhetorical structures (other than ones describing stylistic and structural conventions which modeller covers at the moment). As we saw briefly in section 4.2.1, the rhetorical operators of RST that underlie DOCU-PLANNER are not appropriate for contractual text, as acknowledged by RST proponents and as the experimental RST analysis of an extract of an arbitration agreement showed. As mentioned then, a given piece of text may have multiple RST analyses in terms of discourse relations, in other words, it may yield multiple rhetorical structures. It is not clear how DOCU-PLANNER deals with this situation. It appears that each document genre is associated with a unique illocutionary and rhetorical structure and it might be interesting to explore if and how multiple rhetorical structures per document category can be accommodated and what their impact on the drafting process is.

5

THE REPRESENTATION OF CONTRACTS AT THE MICRO-LEVEL

5.1 Introduction

This chapter seeks to explore possibilities for the representation of contracts at the micro-level, that is, representations of their detailed provisions. The representation of a given contract at the micro-level will be unavoidably less general than its representation at the macro-level.

As was noted in the previous chapter when addressing the representation of contractual content it is difficult to separate issues that pertain to the design of agreements from issues that pertain to the design of documents that record such agreements. This becomes particularly noticeable when we consider detailed representations of contractual provisions, that is, of what the contract actually 'says'.

Chapter 4 mentioned that agreement design raises issues that are relevant to negotiation and to textual realisation of the output of the negotiation process. For example, issues relevant to negotiation include:

(i) How are parties' goals to be represented and checked for consistency?

(ii) In case of conflicts, what compromises can be made that are mutually acceptable to both parties, based on each party's criteria (and preferences) for what is 'best' for it?

Issues relevant to textual implementation of agreements (following negotiation) include:

(i) What document structure is appropriate to reflect the agreement?

(ii) What is the set of norms that accurately reflects what the parties mutually agree on for the regulation of their exchange?

(iii) How can overall consistency and completeness of such set of norms be established?

(iv) What particular linguistic content is appropriate to reflect the set of norms, the agreement?

This research does not focus on negotiation issues themselves. Rather it assumes a (possibly incomplete and vague) specification of what contracting parties agree during the negotiation phase and explores how this is recorded in contractual documents. Hence, it is concerned with some of the issues in the second group. However, because negotiation itself is sometimes conducted by exchanging drafts of contractual documents, issues relevant to the textual implementation of agreements may shed some light on issues relevant to negotiation.

The macro-level view of contractual documents presented in chapter 4 provides some help towards the question of document structure. The theory of normative positions, discussed in section 3.5.3 provides a theoretical and a computational framework that brings to the foreground questions about intended interpretation, structural ambiguity and the precise nature of legal notions. Case analysis, discussed in section 3.5.4, provides an overall theoretical framework for questions of consistency and completeness. The fourth issue noted above, namely the question about the appropriate linguistic content that accurately reflects an agreement, touches upon linguistic communication. Recent research, particularly within the Electronic Commerce community, has been concerned with the development of formal languages for business communication to facilitate electronic exchanges between business partners (for example (Kimbrough 1998); (Kimbrough & Lee 1997); (Kimbrough & Moore 1997)). Some promising developments in that direction have emerged that put forward logic-based frameworks grounded on speech act theory. Section 5.2 outlines the ideas of that research and extends them to address legal contracts specifically. Section 5.3 focuses on the representation of obligations, rights and such other legal relations between contracting parties.

Section 5.4 is concerned with the representation of contracts as protocols and discusses the application of model checking techniques, which have been used in the verification of electronic design, to the verification of contractual protocols.

5.2 Promises and Contracts as Speech Acts

Legal contracts are putative entities that are brought about by communicating parties. Such communication takes the form of a dialogue: A party makes an offer, which the other party accepts; alternatively, a party makes an offer, to which the other party responds with a counter-offer, which in turn is countered or accepted. During the dialogue, a party may issue a variety of statements. For example, it may prompt the other party for information (e.g. by posing a question), volunteer information to the other party unprompted (e.g. by making assertions of fact) make promises about its future conduct, accept or reject promises made by the other party and so on.

Speech act theory has been concerned with the analysis of statements that make up linguistic communication. More recently such statements have been the focus of researchers in the Electronic Commerce community, who seek to develop a formal language for business communication which would be more flexible and expressive than existing EDI[33] standards (cf. (Kimbrough 1998); (Kimbrough & Lee 1986)). Messages exchanged between (at least) two parties are analysed and represented in first-order logic in terms of their content with a view to identify their primary function (for example, whether they are offers, acceptances to offers, promises, requests for services or delivery, instructions for payment and so on). In what follows we present the basic idea through an example, similar to the ones used by Kimbrough. We then discuss its relevance and applicability to contractual exchanges.

Consider the following (telephone) exchange between Peter and Susan:

Peter: Hi, I would like to order a pizza from your menu please.

Susan: Certainly. What kind of pizza would you like and what size?

Peter: The “Good Earth Vegetarian”[*] please, but without onions. Large, please.

Susan: Very well, that will be £13.95, cash please. What is the address?

Peter: 12 Hunger Lane. How long will that be?

Susan: It is now 7pm and we promise to deliver within half an hour. If our driver takes any longer than that, we deduct £1.00 from your bill.

Peter: Ok, thank you.

[*] The menu description of “Good Earth Vegetarian”: mushrooms, onions, red and green peppers, all topped with mozzarella.

[33] **E**lectronic **D**ata **I**nterchange.

The result of this exchange is a contract between Peter (the purchaser) and Susan (the seller) about the provision of some goods (pizza) in return for financial consideration (£13.95 cash). The events that took place and led to this contract are as follows: Peter specified his desired description of pizza in terms of size and toppings based on what Susan's menu offers with one modification (no onions). Susan stated that she can deliver such pizza at the address provided within specific time limits (7:00–7:30 pm) and stated that in case of delay in delivery she deducts £1.00 from the bill. Peter was satisfied with that and (presumably) proceeded to wait for his pizza.

The exchange that led to such contract can be analysed in terms of the utterances involved: There is a request made by Peter to Susan, concerning the delivery of pizza of specified size and toppings. There is an acknowledgement of this request made by Susan to Peter and an assertion (made by Susan to Peter) that the kind of pizza he specified is available and can be delivered to the address provided. Moreover, there is a promise made by Susan to Peter that the specified pizza will be delivered in return for the quoted price within specified time bounds, and a promise that should there be delay in delivery there will be a deduction in price. Peter's acceptance implies a promise on his part to Susan that upon delivery of the pizza he will pay the required price[34].

In spirit similar to Kimbrough (1998), we can represent the performative utterances of the exchange in speech act theoretic terms using event semantics and thematic roles. Kimbrough's idea is essentially the following:

Every utterance conforms to the F(P) framework of speech act theory (which we saw briefly in section 4.6), that is, every utterance U can be rewritten in F(P) form (U ⇒ F(P)), where F is an illocutionary force (such as "assert", "promise", "request", "confirm" etc.) and P is the truth-functional propositional content of the utterance. For example the F(P) structure of the utterance "Susan promised Peter to deliver a pizza" is

promise("Susan delivers a pizza to Peter")

[34] To be precise there are other utterances involved, such as Susan's question about Peter's address and Susan's question about Peter's order, which are omitted from this discussion for simplicity, as they are not instrumental in forming the agreement between them. Rather they facilitate the information exchange between them which fixes the contents of their agreement. Similarly, there is an implicit declarative utterance on Susan's part that "Good Earth Vegetarian" corresponds to the item described in her menu and it is on this knowledge that Peter operates when he uses the term to describe his order. (Declaration here is used in its speech act theory sense, whereby "saying so, makes it so"—following Searle's (1969) classification (cf. Appendix C)).

Hence, “Susan promised Peter to deliver a pizza” is modelled as a promising event where the agent doing the promising is Susan, the recipient of this promise is Peter and the theme or content of the promise is delivering a pizza. The propositional content of an utterance is itself a sentence, which can be analysed grammatically in terms of its noun phrase and verb phrase. Kimbrough (1997) drawing upon ideas from (Parsons 1990) analyses the verb phrase component using event semantics and thematic roles. Hence, the propositional content of our example utterance (“Susan delivers a pizza to Peter”) is modelled as a delivery event with Susan as its agent, Peter as its beneficiary and the pizza (which can be described, if desirable, in terms of its size and toppings) as its content.

Let us first illustrate Kimbrough’s idea by showing how one of the utterances of the exchange between Peter and Susan (Peter’s request for the delivery of a pizza) can be represented. The details of the representation are slightly different from that used by Kimbrough. We comment on this and on possible improvements presently.

- Peter’s request to have one large pizza delivered, of the description corresponding to “Good Earth Vegetarian” but without onions, as a requesting event from Peter to Susan about a delivery event caused by Susan with Peter as its recipient and the said pizza as its content[35]:

$$
\begin{array}{llll}
(\exists e)(\exists e')(\exists c)(\exists t1)(\exists t2)(\exists t3)(\exists t4) & (\mathit{request}(e) \wedge & & \\
 & \mathit{agent}(e, \mathit{peter}) \wedge & & \\
 & \mathit{recipient}(e, \mathit{susan}) \wedge & & \\
 & \mathit{content}(e, e') \wedge & & \\
 & & \mathit{delivery}(e') \wedge & \\
 & & \mathit{cause}(e', \mathit{susan}) \wedge & \\
 & & \mathit{recipient}(e', \mathit{peter}) \wedge & \\
 & & \mathit{content}(e', c) \wedge & \\
 & & & \mathit{pizza}(c) \wedge \\
 & & & \mathit{quantity}(c, 1) \wedge \\
 & & & \mathit{size}(c, \mathit{large}) \wedge \\
 & & & \mathit{topping}(c, t1) \wedge \\
 & & & \mathit{mushroom}(t1) \wedge \\
 & & & \mathit{topping}(c, t2) \wedge \\
 & & & \mathit{red\_pepper}(t2) \wedge \\
 & & & \mathit{topping}(c, t3) \wedge \\
 & & & \mathit{green\_pepper}(c, t3) \wedge \\
 & & & \mathit{topping}(c, t4) \wedge \\
 & & & \mathit{mozzarella}(t4))
\end{array}
$$

[35] The layout is to assist readability.

Notice that this can be converted to normal form and after skolemization gives rise to the following collection of assertions (grouped to assist readability)[36]:

request(e1).
agent(e1, peter).
recipient(e1, susan).
content(e1, e2).

delivery(e2).
agent(e2, susan).
recipient(e2, peter).
content(e2, c).

pizza(c).
quantity(c, 1).
size(c, large).
topping(c, t1).
mushroom(t1).
topping(c, t2).
red_pepper(t2).
topping(c, t3).
green_pepper(t3).
topping(c, t4).
mozzarella(t4).

Note however that the representation, as it stands, does not allow one important distinction to be made, namely the distinction between actual events, that is, events that have occurred, and stipulated events, that is, events that have not yet occurred and might or might not happen in the future. The representation above needs therefore to be complemented with assertions that discriminate actual events and stipulated (future) events. We could represent both kinds explicitly by adding the following assertions:

actual(e1).
future(e2).

Alternatively, if we define *future(Event)* implicitly as the absence of *actual(Event)*, we need only add the first assertion.

We can analyse other utterances of Peter and Susan's exchange in similar manner. If it is desirable, we can attach times to events and define temporal relations between them in the spirit of the event calculus (Kowalski & Sergot 1986). For example, Peter's request that concerns a future delivery can be represented as follows, with an appropriate definition for the relation *after* between events:

request(e1).
agent(e1, peter).
recipient(e1, susan).
content(e1, e2).
time(e1, 7:00).

delivery(e2).
agent(e2, susan).
recipient(e2, peter).
content(e2, c).
time(e2, t2).

pizza(c).
quantity(c, 1).
size(c, large).
topping(c, t1).
mushroom(t1).

[36] Note that our representation so far, is only slightly different from Kimbrough's (1998) representation of a request for delivery. The main difference is that Kimbrough uses unquantified dummy variables and the requesting event is labelled explicitly as "request-to-deliver". We show quantifiers explicitly so that the result of skolemization is clearer. We choose to label the F-event as merely "request" instead of "request-to-deliver" as we label explicitly the P-event as "delivery".

```
 actual(e1).            after(e2, e1).            topping(c, t2).
                                                  red_pepper(t2).
                                                  topping(c, t3).
                                                  green_pepper(t3).
                                                  topping(c, t4).
                                                  mozzarella(t4).
```

- Susan's acknowledgement of Peter's request for delivery, as an acknowledgement event originating from Susan and directed to Peter, with the said request for delivery as its content:

```
aknowledgement(e3).
agent(e3, susan).
recipient(e3, peter).
content(e3,e1).
actual(e3).
```

- Susan's assertion that the said pizza of the required quantity, size and description is available for £13.95:

```
assertion(e4).              available(c).
agent(e4, susan).           cost(c, p).
recipient(e4, peter).       price(p).
content(e4,c).              value(p, £13.95).
actual(e4).                 available(t1).
                            available(t2).
                            available(t3).
                            available(t4).
```

- Susan's promise to deliver the requested pizza at the address provided, between specific time bounds (where *loc* can be further analysed in terms of street number, street name and so on):

```
promise(e5).                time(e2, between(7:00, 7:30)).
agent(e5, susan).           location(e2, loc).
recipient(e5, peter).
content(e5,e2).
actual(e5).
```

Note that the *time* predicate is just one way to represent the time bounds within which Susan promised to make the delivery. A more interesting point to consider is whether the content of Susan's promise (*e5*) is indeed the delivery event (*e2*) contained in Peter's request. Put more generally, the question is whether the P-component of a promise is the *same* as the P-component of a request that gave rise to that promise. We think that in the general case, this is not so, for it is difficult to decide when two events are identical. For example, suppose that Peter, in making his request, only specified certain features of the delivery event, such as what was to be delivered and when. If Susan promises to cause a delivery event, which matches the features specified by Peter, but has one or more additional features, say, that delivery will be done by motorbike and the pizza will be packed in a red carton, is the delivery event that she is describing to be taken as identical to the one described by Peter? Strictly speaking the two delivery events are not identical perhaps but they are very similar[37]. If Susan does indeed proceed with her promise and the delivery event she stipulates becomes actual, then Peter (in the absence of any information to the contrary) would probably regard it as matching his stipulated delivery event and therefore would consider Susan's promise as fulfilled. A more accurate representation of Susan's promise to deliver is therefore the following, which shows explicitly that the content of Susan's promise is a delivery event (*e21*) that *matches* the delivery event contained in Peter's request:

```
promise(e5).                  time(e2, between(7:00, 7:30)).
agent(e5, susan).             location(e2, loc).
recipient(e5, peter).         matches(e21, e2).
content(e5,e21).
actual(e5).
```

The predicate *matches* should contain additional arguments to indicate its intended meaning here, that is, that some event "counts as" fulfilment of a request. For other types of speech acts *matches* might need to be defined differently. We omit such considerations here for simplicity.

The natural question that arises is how similarity between events is to be established. In practice, this can be done in a number of ways:

Perhaps common sense knowledge is to be used: consider, for example, the case where Peter asks for a pizza to be delivered in the early evening, and Susan promises to make such delivery at 7:00 pm. One could argue that the time specified by her satisfies the description "early evening" stipulated by Peter and therefore a "delivery at 7:00 pm" is to count as a "delivery in the early evening". Another possibility for establishing similarity between events is through contractual provisions. Although the example we use here is too simple to show

[37] Questions about reference and identification have been at the core of much philosophical debate (for example, see among others Leibniz's thesis on the Identity of Indiscernibles (Loemker 1969), Frege's 'On Concept and Object' (Geach & Black 1977), Wittgenstein (1961), Quine (1960) and many others) but we do not consider such debate here.

this, consider the case where Peter is operating not as an individual client, but rather as the representative of a supermarket chain, and he specifies that the delivered pizzas are to comply with a number of requirements (for example, be packed in red cartons labelled with the supermarket chain's name, have a minimum of five two-millimetre thick slices of green pepper, be topped with mozzarella of a specific quality and quantity and so on). For Susan's delivery to count as fulfilling her promise, her stipulated delivery event (whose content is the pizza description) would have to comply with these requirements. The sample gas purchase contracts that we examined for this research abound with provisions that lay out standards for the quality of gas, the state of the equipment used to monitor it and so on. A third possibility for establishing similarity between events is through legislation that might govern a particular aspect of business transactions between parties, in similar manner to contractual provisions. Just as vague terms found in legislation are sometimes analysed and interpreted with the assistance of case law, precedents could help to establish whether a delivery event *matches* a request.

- Susan's assertion that upon untimely delivery the price of the said pizza is reduced:

```
assertion(e6).                  price(p).
agent(e6, susan).               value(p, £12.95) ← brings_about(susan, e2, after(7:30)).
recipient(e6, peter).
content(e6,p).
actual(e6).
```

- Peter's (implicit) promise to pay the required price upon timely delivery:

```
promise(e7).              price(p).
agent(e7, peter).         value(p, £13.95) ← brings_about(susan, e2, between(7:00-7:30)).
recipient(e7, susan).     value(p, £12.95) ← brings_about(susan, e2, after(7:30)).
content(e7,e8).
payment(e8).
agent(e8, peter).
recipient(e8, susan).
content(e8, p).
actual(e7).
```

This is only a simple way to represent the content of Susan's assertion and Peter's promise. Generally the content of both (the P-component) is a conditional statement. Conditional promises and assertions raise similar problems as conditional obligations.

The reader may have noticed by this stage that all events have the same markers for certain thematic roles such as agent, recipient and content. This is not exactly in accordance to Kimbrough's suggestion. In analysing verb phrases (that correspond to F-verbs and P-verbs denoting illocutionary and propositional aspects respectively) Kimbrough adopts the thematic roles identified by Parsons (1990). Examples of such thematic roles are included in Table 5–1. Kimbrough (1998) notes that "there is as yet no consensus regarding the exact list and definitions of the thematic rôles, but neither do there appear to be theoretically significant debates over the list"[38]. The choice of markers for certain roles in our examples is for the sake of simplicity, rather than the product of an informed decision after careful consideration of the list. Were we to adopt the list, we would need to define dependencies between such roles, for example the relation between the cause of an event, its instrument and its agent, or the relation between the patient, the experiencer and the theme of an event, in order to yield some ontology within which to describe performative utterances involved in the formation of contractual relations. However, the representation can be easily adjusted should a standard collection of thematic roles emerge.

Rôle	**Description**
Agent	Volitional initiator of action
Patient	Object or individual undergoing action
Theme	Object or individual moved by action
Goal	Individual toward which action is directed
Source	Object or individual from which something is moved by the event, of from which the event originates
Experiencer	Individual experiencing some event
Beneficiary	Object that benefits from the event
Location	Place at which the event is situated
Instrument	Secondary cause of event; the object or individual that causes some event that in turn causes the event to take place

Table 5-1 Examples of Thematic Roles (Kimbrough 1998)

Rather than attempting to explore such considerations, we are interested in the effects of such performative utterances. As we saw in chapter 2, where the classification of contractual obligations was discussed briefly from the perspective of Contract Law (section 2.2.2) both explicit promises (that an action will or will not happen in the future) and representations about states of facts are regarded as undertakings. Undertakings are distinguished by Law into promissory conditions, warranties and intermediate terms. Breach of a promissory condition by a party gives the innocent party the right to treat the whole contract as terminated and absolves him from performing any of his obligations. Breach of a warranty however, merely gives the innocent party the right to claim damages (and creates the obligation for the party in breach to see to it that such damages are properly awarded). Breach of an

[38] As noted earlier, Kimbrough's work is aimed at developing a formal language for business communication that is more expressive and flexible than current EDI standards. One of the requirements that he establishes for such a language is that such

intermediate term has *a priori* undefined consequences which are established through litigation, where the court assesses the extent of the loss for the innocent party and decides whether to treat the term as a promissory condition or a warranty.

In the pizza-ordering example, we concentrated therefore on the participants' promises and assertions (representations about states of facts). The questions that we might be interested to answer, in relation to the effects of such promises and assertions are for example:

(i) What is Susan obliged to do (if anything)?

(ii) What is Peter obliged to do (if anything)?

(iii) What happens if a pizza gets delivered to Peter within the specified time bounds but it does not conform to his required size and description?

(iv) What happens if a pizza of the required size and description gets delivered to Peter outside the specified time bounds?

(v) What happens if a pizza of the required size and description gets delivered within the time bounds but is not intended for Peter? (Say for example in the case that as chance would have it, the resident of 13 Hunger Lane sharing the same taste in pizza as Peter, ordered a large, no-onions "Good Earth Vegetarian" from Susan but the driver mixed up the address).

Of those questions, the first two are the most essential. Answering those facilitates answering the rest. Our common intuition about a promise is that it is the undertaking of an obligation to perform a certain act (or more generally to bring about a state of affairs, to see to it that a certain act that results in a state of affairs is performed). Assertions of a statement of fact are also regarded as undertakings in some versions of speech act theory[39]—to assert that *p* counts as an undertaking to the effect that *p* represents an actual state of affairs; hence if *p* is not already true in the actual world, the speaker of the assertion places himself under an obligation to see to it that *p* becomes true in the actual world. It is the obligations that result from performative utterances such as promises and assertions that are of interest in representing the contents of contracts at the micro-level. Analysing exchanges at the negotiation stage (whether they be oral or in writing, if negotiation takes place by exchanging drafts of

language be based only on publicly available lexicons and a public grammar. Consensus over the list of thematic roles seems therefore essential in order to satisfy this requirement.

proposed terms) helps bring to the foreground the legal relations that are created between the parties, which we may subsequently want to check for consistency, completeness and so on.

The obligations imposed on Susan and Peter are not explicit in the representation of their utterances presented earlier, so they are not directly accessible either for inspection or for reasoning about them (or with them). What is lacking from our representation of promises and assertions is their effects or in event terms, their postconditions. Predicates such as *promise, assertion* and *request* are intended to have some fixed meaning that we have not discussed yet. The pizza-ordering example suggests that the contract formed between Peter and Susan is unilateral. As we saw in chapter 2 in unilateral contracts a promise on one party is exchanged for an act on the other party and only the promissor is bound to perform. Peter promises to pay the quoted price for pizza if Susan gets it delivered; Susan may do nothing at all at the risk of losing a customer, but once she does get the pizza delivered, Peter is obliged to pay. Peter's promise (and the obligation it effects) is in other words conditional upon Susan's performance. Our representation of promises and assertions in event terms requires therefore the notion of precondition.

Towards representing the effects of performative statements, Kimbrough (1998) suggests that promissory performative utterances such as "Susan promised Peter to deliver a pizza" can be paraphrased more transparently as "there is a promising event in which the speaker is Susan and the benefactive is Peter, and in which Susan keeps this promise if and only if she does something that causes there to be a delivery of pizza to Peter". Kimbrough notes that illocutionary forces such as promising create intensional contexts, in which substitution of equivalents cannot be done confidently[40]. The first-order logic representation of this idea, as suggested by Kimbrough, becomes:

$$\begin{aligned}
&\exists e(promise(e,a^*) \wedge \\
&\quad speaker(e, susan, a^*) \wedge \\
&\quad addresee(e, peter, a^*) \wedge \\
&\quad \forall w\,(A(a^*,w) \rightarrow (K(e,w) \leftrightarrow \exists e'(delivery\,(e', w) \wedge \\
&\qquad\qquad cause(e', susan, w) \wedge \\
&\qquad\qquad benefactive(e', peter, w) \wedge \\
&\qquad\qquad theme(e', pizza, w))))
\end{aligned}$$

[39] See Appendix C, Searle (1969) on the structure of illocutionary acts. See also Walton & Krabbe (1995).

[40] Kimbrough (1998) notes for instance, "to promise that you will come to the party is not the same as to promise that you will skip Esmeralda's wedding, even if you skip Esmeralda's wedding if and only if you come to the party". This seems to raise

Such a representation, Kimbrough suggests, yields the paraphrase: "there is a promising event in the actual world, $a*$, in which the speaker is Susan and the addressee is Peter, and for any possible world w, accessible from the actual world $a*$ (denoted by $A(a*,w)$), the promise is kept in that world, (denoted by $K(e,w)$) if and only if there is a delivery event e', in that world, Susan causes that event, the benefactive of the event is Peter and the theme of the event is the pizza".

The simpler representation of this without possible worlds and intensionality, is according to Kimbrough:

$$
\begin{aligned}
&\exists e(promise(e) \wedge \\
&\qquad speaker(e, susan) \wedge \\
&\qquad addresee(e, peter) \wedge \\
&\qquad (K(e) \leftrightarrow \exists e'(delivery(e') \wedge \\
&\qquad\qquad cause(e', susan) \wedge \\
&\qquad\qquad benefactive(e', peter) \wedge \\
&\qquad\qquad theme(e', pizza)))
\end{aligned}
$$

Such a representation however departs from the *prima facie* meaning of the utterance "Susan promised Peter to deliver pizza". For it does not represent explicitly what Susan promised to do (or more generally to bring about) and only that. Rather it represents what Susan promised to do in terms of the circumstances under which the promise will be deemed as kept. This seems to suggest that the representation of each promissory statement carries its own distinctive notion of being kept. The notion of keeping a promise is generic, (that is, independent of what is promised in specific circumstances) and corresponds to the intuition that the content of the promise (that is, the event that is being promised) is brought about, or in event terms that it occurs. Hence, Susan's promise to Peter to deliver pizza can be represented (after skolemization) as a collection of assertions:

considerations about the appropriate principles of detachment for illocutionary operators, and according to Kimbrough, it is not the case that *promise(p↔q) ≡ promise(p) ↔ promise(q)*.

```
promise(e1).
speaker(e1,susan).
addressee(e1, peter).
actual(e1).
theme(e1,e2).
delivery(e2).
cause(e2,susan).
benefactive(e2, peter).
theme(e2, pizza).
```

Peter’s promise to pay Susan some price *after* he receives the promised pizza can be represented as another collection of assertions:

```
promise(e3).
speaker(e3, peter).
addressee(e3,susan).
actual(e3).
theme(e3,e4).
payment(e4).
cause(e4, peter).
benefactive(e4,susan).
theme(e4, price).
precondition(e4,brings_about(susan,e2)).
```

Both promises share the same notion of being kept, which can be expressed as a rule of the form:

$$promise(X) \rightarrow (kept(X) \leftrightarrow agent(X,A) \wedge theme(X,Y) \wedge brings\_about(A,Y)).$$

with an appropriate definition for *brings_about(A,Y)* (“A brings about Y”). Variables such as *X*, *Y*, and *A* are implicitly universally quantified.

This is not necessarily the best representation, but hopefully it illustrates that keeping a promise can be expressed in more generic terms. Put abstractly, the general schema that we propose for representing the notion of a kept promise is the following (omitting for clarity the full description of events):

$$\forall e \forall e'[(promise(e) \wedge content(e,e')) \rightarrow [kept(e) \leftrightarrow \exists e''(happens(e'') \wedge matches(e'',e'))]]$$

Another, more interesting point, however is that obligations effected by promises are still not explicit in this representation. Satisfying an obligation that results from a promise entails that the promise is kept. Therefore, the definition of a kept promise conceals the notion of a fulfilled obligation. Susan's promise, for example, to deliver a pizza to Peter, initiates an obligation on her part to bring about such delivery. This can be expressed as the postcondition of the promising event:

promise(e1).
speaker(e1,susan).
addressee(e1, peter).
theme(e1,e2).
actual(e1).
postcondition(e1,o1).
obligation(o1).
bearer(o1,susan).
counterparty(o1, peter).
content(o1,e21).
delivery(e2).
cause(e2,susan).
benefactive(e2, peter).
theme(e2, pizza).
matches(e21,e2).

Here Susan's obligation *o1* can be analysed in terms of its bearer, counter-party and content. We shall discuss the representation of obligations in more detail in the next section. But before we turn to that let us summarise the main points of the discussion so far.

Contractual agreements between parties come about through an exchange of statements (oral or written). Contractual documents recording such agreements are declarative[41] instruments stating what the legal relations between the parties are. Such relations result from the pre-contractual exchange between parties, during which assertions of fact are made, promises are issued, accepted and met with counter-promises, questions eliciting information are posed and answered and so on. Contractual documents do not record such pre-contractual exchange; rather they record its outcome. Of all the performative statements made by parties during the pre-contractual exchange, assertions, promises and declarations are most interesting. An explicit representation of them, supported by an appropriate formalization of speech act theory (for example Searle & Vanderveken 1985) facilitates establishing whether a valid contract exists between two parties and what its content is about. Analysing promises, assertions and declarations enables the association between the parties' legal relations (obligations, rights and so on) and the performative statements that gave rise to them to be made explicit, which can be useful in the context of negotiation, where an agreement is designed.

Inspired by Kimbrough's work on the development of a formal language for business communication, we have discussed how performative statements exchanged between parties can be represented. Our proposal improves on Kimbrough's suggestion in the following respects:

(i) We argued for the distinction between actual events and stipulated (future) events to be made explicit. In this way the fact that a promise or a request refers to a future, unrealized at the time of the promise, event is clearer. Whether assertions refer to actual facts (past or present events) and hence have informative nature or future events and hence have promissory nature is also made clearer.

(ii) We argued for the association of events with times, where appropriate, and the extension of Kimbrough's framework with a suitable temporal representation, such as the event calculus. This is a necessary extension of the framework to accommodate contractual situations realistically, since the notion of time (in relative or absolute terms) is intrinsic to contracts.

(iii) We argued for the explicit representation of similarity between events, where appropriate, and the extension of Kimbrough's framework with suitable definitions for similarity, based on common sense knowledge or information provided by the contract or legislation where it is available. This is more flexible than insisting upon identity between events and enables questions about fulfilment or otherwise of promises (and consequently of requests that gave rise to such promises and of obligations that are entailed by such promises) to be addressed more realistically.

(iv) We argued for a generic representation of the notion of a fulfilled promise in place of Kimbrough's proposal, which associates each promise with an explicit

[41] Here "declarative" is meant in Searle's sense.

representation of what would render it fulfilled. Our proposal stems from the observation that the notion of fulfilment can be defined abstractly in terms of the occurrence of the content of a promise independently of the particular type of such content.

(v) We argued for the extension of Kimbrough's event descriptions with preconditions and postconditions. The former may go some way towards expressing conditional performative statements (such as conditional promises). The latter enable the association between performative statements and their effects to become explicit. Hence, obligations that are incurred by promises or assertions of a promissory nature are highlighted and become accessible.

The representation was tried on extracts from contractual documents. The extracts were analysed as collections of assertions, declarations and promises (as mentioned previously, contractual documents do not contain questions, which might have been included in the pre-contractual exchange between parties).

Since contractual documents record the outcome of pre-contractual exchanges between parties, the process of constructing such representations from contractual provisions is (not surprisingly) interpretative. Most assertions of fact are found in definitional provisions (that is,, where the meaning of certain terms is fixed for the purposes of the contract) and can be taken to be issued mutually by parties. Identifying promissory statements relies on interpreting prescriptive provisions that specify obligations for the parties and working backward towards the promises that brought about such obligations. For example, consider the following extract from (IEE 1991) which specifies an obligation on the part of the Engineer towards the Purchaser[42]:

> The Engineer shall, with due care and diligence, design, manufacture, test and deliver the Plant within the Time for Delivery.

This is construed as a promise on the Engineer's part, directed to the Purchaser, whose content is the conjunction "design, manufacture, test and deliver". In event terms, this can be analysed as four actual promissory events, each of which has as content one of the (future) P-events corresponding to "design", "manufacture", "test" and "deliver" and as postcondition the corresponding obligations for the Engineer. The P-events themselves have as content the Plant, and are time bounded. The expression "with due care and diligence" is not specified any further in (IEE 1991). Future debate between the parties (possibly in a litigation context) about whether a design (or testing, or manufacturing or delivery) event actually brought about by the Engineer, matches the stipulated one in the promise might centre on the meaning of this expression and common sense or legislation might decide the outcome.

The representation was also tried on contractual provisions that specify procedures followed by the parties in order for certain goals to be achieved. Such procedures essentially determine the way in which the agreement is operationalised. For example, provisions that specify what a party must do in order to process a claim raised by another party were analysed (by tracing backwards the promises and assertions made by the parties that led to such provisions to be established).

Our experimentation suggests that such a representation of performative statements is very promising both for analysing and representing the content of messages as an alternative to EDI standards (which is Kimbrough's objective) and as a possible way for representing contractual provisions and other documents (such as requests, invoices, notifications etc.) that are exchanged between parties during the course of their transaction in accordance to their agreement. However, many issues require further examination. The content of a promise, a request or an assertion of a promissory nature is often much more complicated that a single one-off event. For instance, using the terms of the pizza-ordering example, Susan might promise to deliver pizza every day at 7:00 pm, unless she is too busy, in which case she will be delivering garlic bread. In our sample British Gas contracts the details of how, how often and what quantities of gas is to be delivered are typically given by an extremely complex set of conditional statements and procedures.

The treatment of conditional promises itself is an unresolved issue. It is also problematic in Kimbrough's proposal. For example, when Susan promises to lower the price for a pizza if it is delivered past 7:00 pm, what is the content of her promise? Let *a* denote "the time is after 7:00 pm" and *b* denote "the price for the pizza is lowered". Susan's promise could be viewed as an instance of the following schema:

$$a \rightarrow promise(b)$$

But it seems more natural to view it as an instance of the following schema:

[42] "Purchaser" and "Engineer" are the labels used for the two parties in (IEE 1991).

$promise(a \to b)$

That is to say that the content of the promise is a conditional statement. This suggests that even for the simple one-shot delivery scenario discussed by Kimbrough and analysed in this section, one can question whether, in generic terms, the content of promissory statements are best viewed as events or as conditional statements (with empty conditions sometimes) that certain events happen. In order to generalise Kimbrough's proposal so that its applicability becomes wider, fundamental questions need to be addressed in more detail. Such considerations form part of our agenda for future work.

We now turn to the parties' legal relations themselves, as they are established by a contract, focusing on obligations.

5.3 Patterns of Contractual Obligations

In this section, we concentrate on the legal relations that obtain between parties once they have entered an agreement. We take a process view of contracts and analyse them in terms of the obligations that are active at various points during their life span. The progression of an agreement through various states is illustrated by state transition diagrams. An informal notation is introduced that summarizes conveniently the states of an agreement as it evolves over time. It should be stressed that the aim of this section is not to produce a formal characterization of contractual content but simply a concise informal notation that can be used to summarize the structure of interrelated obligations that constitute a particular agreement.

Let us first consider a simplified version of the pizza-ordering agreement between Peter and Susan, where detailed temporal elements are ignored. This simple version of the agreement is that Susan is to deliver a pizza (of specified size, quantity and description) and Peter is to pay the required price in return for such delivery.

Let us introduce some elements of the notation that will be used in our discussion of the simplified pizza-ordering agreement. Let $x:a$ stand for "agent x sees to it that a". Expressions of the form $not\ x:a$ are intended to be read as "it is not the case that agent x sees to it that a". The simplified agreement *C* between Susan and Peter can be represented in terms of the obligations that are imposed on the parties. These can be represented by expressions of the form $O_x a$, which denote that it is obligatory for agent x to see to it that a. Such expressions are not necessarily assumed to be closed under logical consequence. As we will see in a moment, we also need a conditional.

We could generalize the notation for obligations to show explicitly the counter-party, for example by using expressions of the form $O_x^y a$ to denote that agent x bears an obligation towards agent y to see to it that a. However, this is not necessary because in two-party agreements the counter-party of a legal relation can be inferred easily as the counter-party of the contract. This is also the case for agreements where more than one legal persons are mentioned. For example, in (IEE 1991) the contract is formed principally between the Engineer and the Purchaser (of the mechanical or electrical plant). The contract mentions separate duties for the Engineer's Representative. Some of his obligations are explicitly stated as directed to the Engineer (who appoints him) and some appear to be directed to the Purchaser. Where e denotes the Engineer, p denotes the Purchaser and r denotes the Engineer's Representative, certain provisions seem to correspond to obligation expressions of the form $O_r^e a$ and others seem to correspond to obligation expressions of the form $O_r^p a$. In the case of the latter expressions, however, as other provisions of the contract reveal, where such duties are violated the liability is borne by the Engineer. Hence, although some provisions appear to be specifying duties of the Engineer's Representative towards the Purchaser, they are in fact specifying duties of the Engineer towards the Purchaser—the Engineer's Representative is merely the instrument through which the Engineer is to satisfy such duties. The Engineer appoints the Engineer's Representative by exercising his power to do so[43] and it is by virtue of such appointment that the Representative has certain duties to perform. Provisions that seem to raise obligation expressions of the form $O_r^p a$ correspond rather to obligations whose structure is $O_r^e a \wedge O_e^p a$—where the first conjunct is derived from the separate agreement between the Engineer and his Representative. This is in accordance with the intuition that was discussed in chapter 3, that an agent bears obligations that concern his own actions, which might indirectly influence other agents' actions[44].

The simplified agreement between Peter and Susan goes as follows, where a stands for "delivering a pizza of the specified size, quantity and description to Peter", b stands for "paying the required price to Susan", s stands for Susan and p stands for Peter:

[43] "Power" here means that if the Engineer appoints the Representative, then according to the contract between the Engineer and the Purchaser, the latter recognizes such appointment. The Engineer must also have power to appoint his Representative as far as the Representative is concerned, but this would be part of a separate agreement between the Engineer and the Representative.

[44] This is also relevant to the discussion on indirect action, influence and responsibility in (Santos & Carmo 1996).

It is obligatory for Susan to see to it that a pizza (of specified size, quantity and description) is delivered to Peter, that is, $O_s a$. If Susan fulfils her obligation, then it is obligatory for Peter to see to it that the required price is paid to Susan, that is, if $s:a$ then $O_p b$. If Susan does not fulfil her obligation, that is, if $not\ s:a$ then it is not obligatory for Peter to see to it that she gets paid, that is, $\neg O_p b$. In this case, other norms may also apply. As was noted in chapter 2, non-fulfilment of contractual obligations yields different consequences, depending on the status of the violated provision. If a party violates a provision that was deemed as a promissory condition, then the counter-party may claim damages and declare the contract void (thus all of the counter-party's obligations under the contract are discharged). On the other hand, if a party violates a provision that was deemed as a warranty, then the counter-party may claim damages, but the contract persists, that is, the counter-party remains bound to his own obligations under the contract. If Susan's obligation $O_s a$ is a promissory condition, then following its breach Peter bears no obligations and she may be obliged to pay damages. That is, where f denotes paying damages, if $not\ s:a$, then $\neg O_p b \wedge O_s f$. If Susan's obligation is a warranty, then following its breach Peter is still obliged to see to it that she gets paid and she is obliged to see to it that damages are paid to Peter, that is, if $not\ s:a$, then $O_p b \wedge O_s f$. It might be the case that the contract does not specify anything in particular for the case where Susan violates her obligation (perhaps because when the agreement was formed, the status of this obligation as a promissory condition or a warranty was not established, that is, it was left as an intermediate term). In this case following $not\ s:a$ the agreement is perhaps terminated and no specific remedies are available to Peter, although possible litigation might ensue to establish whether he may claim damages.

Given these possibilities for the case where Susan violates her obligation, let us assume that the first is the intended one, that is, $\neg O_p b \wedge O_s f$ after $not\ s:a$. A natural question to pose is what happens if Susan fulfils her obligation to see to it that damages are paid to Peter, that is, what happens after $s:f$? One possibility is that the contract returns to its normal course, that is, Peter's original obligation becomes active again ($O_p b$). Another possibility is that the contract is terminated. A third possibility is that the contract continues but with different obligations assumed by the parties, for instance $O_s c$ for Susan and some state of affairs c, and $O_p d$ for Peter and some state of affairs d. A similar line of questioning can be pursued in relation to Peter's obligation to see to it that Susan gets paid, that is, we can explore what consequences are entailed by the fulfilment or the violation of such obligation.

The foregone discussion highlights the need for a notation that summarizes the structure of the contract between Peter and Susan so that we can determine what obligations hold for them given certain facts that concern the performance or otherwise of actions against given obligations.

One possibility is to represent the contract as a state diagram such as the one illustrated in Figure 5–1, which summarizes some of the discussion of the simplified pizza-ordering example. Dashed lines indicate the options that were discussed earlier, that is, whether after Susan's violation of her obligation (S_3), Peter's original obligation comes into force (S_1), or whether the contract is terminated (S_3 is final) or whether the contract continues with new obligations for both parties (S_5).

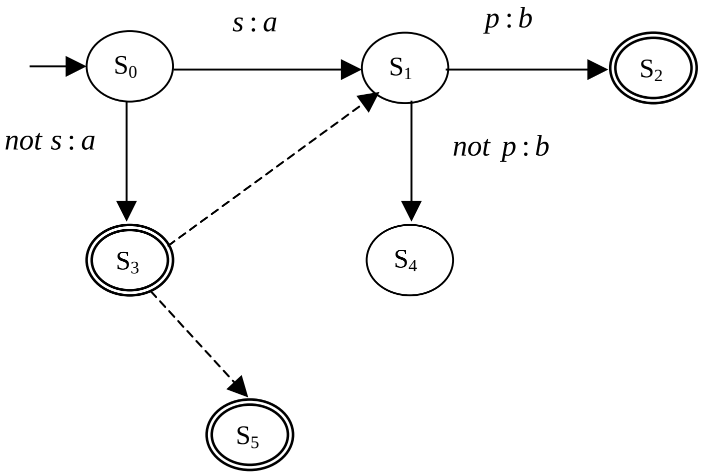


Figure 5-1 State diagram for simplified pizza-ordering example

Each state offers a (possibly partial) view of the status of the agreement as it evolves given the parties' actions with respect to their obligations. That is, each state corresponds to the obligations (and more generally other legal relations such as rights, powers and so on) that obtain between the parties. Transitions correspond to the performance or non-performance of actions by parties that effect transformations to the status of the agreement. There is no attempt here to provide a formal account of action negation. Expressions such as $not\ s:a$ are merely labels for a kind of transition.

In the state diagram of Figure 5–1, state S_0 corresponds to Susan's obligation $O_s a$ and Peter's obligation $O_p b$ (which is conditional on the fulfilment of $O_s a$). State S_1, which follows Susan's fulfilment of her obligation, corresponds to Peter's obligation coming into force. Similarly, state S_3, which corresponds to Susan's violation of her obligation, corresponds to $\neg O_p b \wedge O_s f$ as was noted in the earlier discussion.

Instead of drawing state diagrams explicitly, we can adapt a notation found in modal languages for transition systems such as modal action logic or dynamic logic.

Expressions of the form $[\tau]A$, where τ is the label of a transition, denote that A necessarily holds in all states that are reachable following transition τ. Expressions such as $\langle\tau\rangle A$ are defined as the dual of $[\tau]A$, that is, $\langle\tau\rangle A \equiv_{def} \neg[\tau]\neg A$ ("A possibly holds in states that are reachable following transition τ"). Therefore, $\langle\tau\rangle \mathrm{T}$[45] holds at a state S when there is a transition labelled τ out of S. $[\ ]A$ denotes that A is true in all states. This notation appears to be similar to Meyer's (1988) deontic logic which is based on dynamic logic. In Meyer's account however, obligations apply to actions. The τ in expressions of the form $\hat{O}\tau$[46] is an action. In the account presented here, obligations apply to states of affairs Y in expressions of the form $O_x Y$ and τ is a transition between states.

One state is selected as the initial state and *initially* A means that A holds in the initial state. The simplified contract between Peter and Susan can now be represented as follows, where *terminated* is a special symbol denoting a final state for the agreement:

[45] T is the constant symbol for "true".

[46] $\hat{O}$ is Meyer's obligation operator.

$$initially \begin{cases} O_s a \\ \langle s:a \rangle \mathrm{T} \\ \langle not\ s:a \rangle \mathrm{T} \end{cases}$$

$$[\ ](O_s a \to [s:a] O_p b)$$
$$[\ ](O_s a \to [not\ s:a](\neg O_p b \wedge O_s f))$$
$$[\ ](O_s F \to [s:f] O_p b)$$
$$[\ ](O_s F \to [not\ s:f] terminated)$$
$$[\ ](O_p b \to [p:b] terminated)$$
$$[\ ](O_p b \to [not\ \ p:b] terminated)$$

Rules such as the above implicitly define the state space for the agreement. Initially it is obligatory for Susan to see to it that a pizza gets delivered to Peter. It is possible for Susan to see to it that a pizza gets delivered to Peter and it is also possible for her not to see to it that this state of affairs is achieved. A transition labelled $s:a$ from any state in which Susan's obligation holds necessarily leads to a state where it is obligatory for Peter to see to it that Susan gets paid. A transition labelled $not\ s:a$ from any state where Susan's obligation is in force necessarily leads to a state in which Peter is not obliged to see to it that she gets paid and Susan is obliged to see to it that Peter receives damages. Peter's obligation to see to it that Susan gets paid becomes active in all states following either fulfilment of $O_S a$ or fulfilment of $O_s f$. The agreement is terminated 'happily' following Peter's fulfilment of his obligation or 'unhappily' (with possible litigation ensuing) following Susan's failure to see to it that damages are awarded to Peter (after she fails to deliver the pizza) or Peter's failure to pay the required price after Susan successfully delivers the pizza.

In addition, the following rules are built in, where x is a variable ranging over agents' labels and Y is a variable ranging over propositional states of affairs:

(i) $[\ ](O_x Y \to \langle x:Y \rangle \mathrm{T})$ that is, where an agent bears an obligation it is possible for him to perform an action that fulfils it;

(ii) $[\ ](Pow_x Y \to \langle \Pi_x Y \rangle \mathrm{T})$ that is, where an agent has power to effect a legal relation Y it is possible for him to exercise such power;

(iii) $[\ ](Pow_x Y \rightarrow [\Pi_x Y]Y)$ that is, where an agent has power to effect a legal relation Y, the legal relation obtains in all states following his exercise of such power[47].

We could also consider building in the following:

$$[\ ](O_x Y \rightarrow \langle not\ x : Y \rangle T)$$

That is, where an agent bears an obligation it is possible for him to violate it. As we shall see later however, this is a point worthy of more discussion.

In the representation of the simple pizza-ordering agreement above, we observe structures that are similar to contrary-to-duty obligations. For example, concentrating on Susan's obligation we have the following[48]:

$$[\ ](O_s a \rightarrow [s : a]O_p b)$$
$$[\ ](O_s a \rightarrow [not\ s : a]\neg O_p b)$$
$$[\ ](O_s a \rightarrow [not\ s : a]O_s F)$$

That is, if Susan violates her primary obligation $O_s a$ then a secondary obligation $O_s F$ comes into effect. Representing and reasoning about contrary-to-duty structures in a consistent way has been the topic of much debate (cf. Alchourrón 1993)). The consensus seems, however, to be that in temporal settings contrary-to-duty structures are not problematic (cf. (Prakken & Sergot 1996; 1997)).

At first glance, it might appear that contrary-to-duty structures are a common feature of contracts. They seem to emerge from contractual provisions that specify reparations in cases where an obligation is violated. Usually such provisions are associated with warranties as we saw earlier. As mentioned in chapter 2, contractual provisions are not typically explicitly labelled as warranties, promissory conditions or intermediate terms. Normative propositions that stipulate secondary obligations can be used to establish whether normative propositions specifying primary obligations are warranties. In contracts, it is more common for the violation of a primary obligation by one party x to be associated with power for the counter-

[47] This is a simplifying assumption for the purposes of this discussion. As we saw in Chapter 3, having legal power does not generally entail having the ability to exercise it.

[48] Transition labels distribute over compound propositional formulae.

party z to bring about a legal relation R. That is, in general, the following pattern is typically encountered:

$$[\ ](O_x Y \rightarrow [not\ x : Y] Pow_z R)$$

Such legal relation R might entail that the contract is declared void (terminated) if the primary obligation that was violated is treated as a promissory condition. Alternatively, such legal relation R might entail new obligations for the offending party. For simplicity, in the pizza-ordering example Susan's violation of her primary obligation entailed a new obligation for her (to see to it that damages are paid to Peter $O_s F$). It is more accurate to say that Susan's violation of her primary obligation entails Peter's power to create a new obligation $O_s F$ for her. That is:

$$[\ ](O_s a \rightarrow [not\ s : a] Pow_p O_s F)$$

Whether this new obligation will actually come into force depends on whether Peter chooses to exercise his power.

In discussing violation of obligations, we are interested in transitions with negative labels. Let us revisit the original pizza-ordering example, in which Susan's obligation to see to it that a pizza is delivered to Peter was time-bounded and the pizza in question had to conform to some requirements of size, quantity and description. The transition $not\ s : a$ denotes that Susan violates her obligation. Such violation however may come about in different ways. For example, Susan might not deliver a pizza at all. Or a pizza might be delivered, but it might not conform to the agreed specification (in terms of size, quantity and description), that is, Susan brings about something which is different from what she is obliged to bring about ($s : a$'). A third possibility is that the 'right' (in terms of size, quantity and description) pizza is delivered but not within the agreed time bounds; again Susan brings about something which is different from what she is obliged to bring about ($s : a$''). A fourth possibility is that the 'wrong' pizza gets delivered late ($s : a$'''). Other possibilities exist by considering various combinations of what might potentially count as violation of Susan's primary obligation (for instance, the 'right' pizza is delivered on time but not by Susan, the 'wrong' pizza is delivered on time by somebody other than Susan, or the 'wrong' pizza is delivered late and by somebody other than Susan and so on). Instead of a single transition labelled $not\ s : a$, emerging from a state in which $O_s a$ holds, we have a cluster of refined transitions, each corresponding to one aspect of "it is not the case that Susan sees to it that a pizza of the required size, quantity and description is delivered to Peter within specified time bounds". This is illustrated in Figure 5–2, where $A \equiv (a \vee a' \vee a'' \vee a''')$:

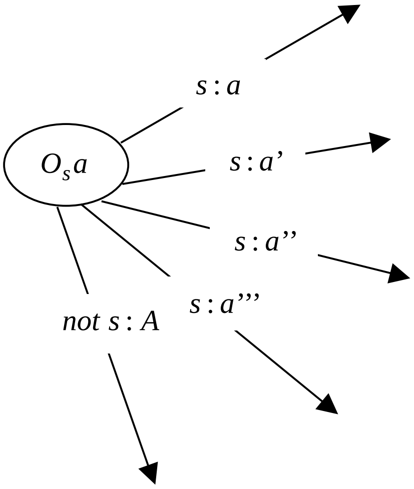


Figure 5-2 Possible transitions following Susan's obligation

It should be stressed again that these are merely labels for transitions and no attempt is made to provide a formal account of negation in action descriptions. Moreover, as the figure above suggests, concurrent actions are in principle possible from a state but no attempt is made here to address the logic of concurrent action.

Exploring the consequences entailed by the violation of an obligation raises other interesting points. For example, as was mentioned above, there are various possibilities, which can be construed as violations of Susan's obligation towards Peter. One way of perceiving the agreement between Susan and Peter is as follows: Susan is obliged to deliver a pizza of any type from a range of available types[49], say *t1* or *t2*. If Susan delivers a pizza of type *t1*, then Peter is obliged to pay a price *p1*. If Susan delivers a pizza of type *t2*, then Peter is obliged to pay a price *p2*. Another way of perceiving the agreement between Susan and Peter is this: Susan is obliged to deliver a pizza of type *t1*. If she delivers a pizza of type *t1*, then Peter is obliged to pay a price *p1*. If she delivers a pizza of type *t2*, then Peter is obliged to pay a price *p2* and Susan is no longer obliged to deliver a pizza of type *t1*. The difference between the two versions is subtle. In the first version, Susan is only obliged to deliver *a* pizza. The exact amount that Peter is obliged to pay in return for such delivery is dependent on what particular

[49] "Type" here is intended to refer to size, quantity, description and possible time restrictions that apply.

kind of pizza is delivered. Nothing seems to preclude the provision of a pizza of type *t1* after a pizza of type *t2* has been successfully delivered. In the second version, Susan is obliged to deliver a *specific* pizza. Non-fulfilment of her obligation to deliver a pizza of type *t1* discharges such obligation and possibly leads to secondary (reparation) obligations on her part that are effected through Peter's exercise of power. The British Gas sample contracts that we examined tend to be like the second version. They specify precisely what happens if the natural gas that is delivered does not conform to pre-agreed quality standards or if delivery quantities and times depart from those stipulated in the contract (for instance prices are adjusted, additional quantities must be delivered at more frequent intervals to make up for the under-delivery and so on).

Let us now turn to the temporal elements of the original pizza-ordering example. Our informal notation for representing agreements implicitly defines a state space. In principle, time can be associated with states (which correspond to the collection of obligations and other legal relations between parties) or with transitions (which correspond to actions performed by parties). That is, expressions of the form $O_x^t Y$ can be used to denote that at time t, agent x is obliged to see to it that Y is the case. On the other hand, expressions of the form $O_x Y^t$ can be used to denote that agent x is obliged to see to it that Y is the case at time t. In the latter form, time is implicitly associated with the agent's action. Put alternatively in the latter case time is associated with the transition $x:Y$. We show this explicitly in our informal notation by extending the labels for transitions to include time in expressions of the form $(x:Y)^t$, in which time is specified in absolute or relative terms, for example via temporal relations such as *before(t), after(t),* or *between(t1, t2)*. Let r denote "paying a reduced price to Susan", a denote "a pizza of the specified size, quantity and description is delivered to Peter" and b denote "paying the normal price to Susan". The state diagram for the pizza-ordering example is now as illustrated in Figure 5–3:

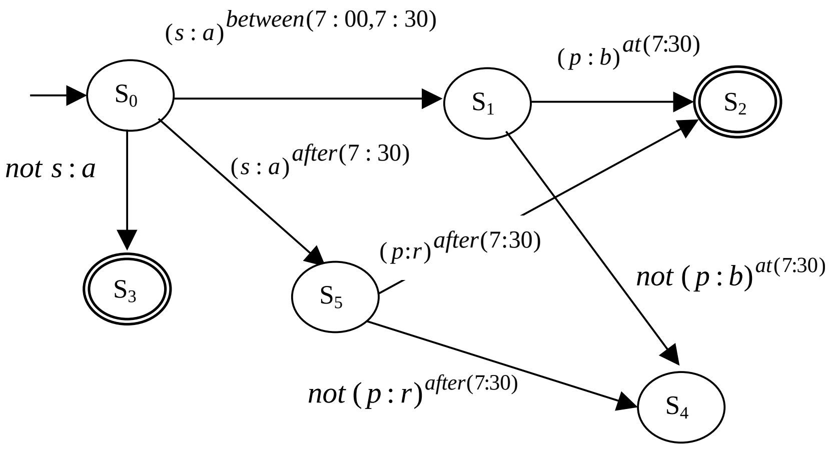


Figure 5-3 State diagram for pizza-ordering example with temporal elements

Note that time is here used informally as a label on transitions—no attempt is made to define formal semantics for it.

Such representation offers a process view of the contract and enables us to determine what the status of the agreement is (what obligations hold), given an event or a sequence of events that concern the performance of actions by agents. As the underlying model is so familiar in Artificial Intelligence (for example situation calculus (McCarthy & Hayes 1969), or simplified event calculus (Kowalski & Sergot (1986)), it would be feasible to build an implementation.

In representations such as this, where events effect transitions between states, the initiation or termination of properties that hold in states (norms in this case) can be expressed through the definition of appropriate relations (such as *initiates* or *terminates*) between events and states as was mentioned earlier. Such relations essentially define the postconditions of events. As mentioned previously, states in our state diagram do not necessarily offer a complete view of the contract. We have only been showing activated obligations in states that are arrived at through some transition, and we have assumed that any obligations from the previous state that are not explicitly stated are assumed to be discharged. In addition to establishing what norms become activated or discharged, we are also interested in what norms persist over transitions between states. This is one of the subjects of ongoing work in the area of temporally qualified obligations (for example (van Eck 1982); (Åqvist & Hoepelman 1981) and the discussion in (Prakken & Sergot 1997)) and has not been dealt with in this work. It is however relevant to possible future extensions to our proposed representation of contracts as processes. As was noted earlier we do not provide a formal account for action negation (negated transition labels) and time constraints here. This is another possible avenue for future work.

The informal notation introduced in this section makes it convenient to represent those parts of agreements that stipulate interrelated obligations. In its present state, the notation is only useful for analysis purposes and further development would be required in order for it to be of practical use. As it stands it enables us to observe the status of legal relations between parties as a contract evolves over time, as a dynamic system. Dynamic systems raise interesting questions about their safety and liveness properties. The next section explores how safety and liveness requirements for contracts can be verified using *model checking*, a technique that has been successful in the domain of software and hardware verification. The discussion in the next section is conducted around Petri net representations of contracts. This is because the software tools that were readily available to us operate on such representations. We show through an example how one can move from state diagrams to Petri nets.

5.4 Verifying Trade-Procedures by Model Checking Petri Nets

The set of norms which parties agree to abide by during their business exchange essentially specifies the parties' ideal mode of exchange. Some researchers, such as Lee and his associates refer to such norms as "trade procedures" (cf. (Lee & Dewitz 1992); (Bons *et al.* 1995)) or "business protocols" (cf. Wrigley 1992) and as was noted earlier in this dissertation they are similar to what software and hardware engineers call "specifications". Questions of completeness and consistency of such sets of norms arise in the context of contractual negotiation and drafting at the micro-level just as questions of verification arise in relation to specifications in software and hardware engineering. Whether a contract (or trade procedure, business protocol, specification) covers all intended cases without conflicts and whether the ideal mode of exchange that it describes has the appropriate safety ("nothing bad will happen") and liveness ("something good will happen") properties is a concern both during pre-contractual exchanges and when drafters formulate detailed provisions to record the result of such pre-contractual exchanges. Ill-defined contracts may result in undesirable situations when put into practice, with parties finding that they cannot execute them or that unanticipated circumstances arise which cannot be resolved without resorting to costly and lengthy litigation. If a contract can be both formally specified and *verified*, that is, checked for undesirable pathological features, then not only is its negotiation conducted more effectively but also its subsequent performance is smoother.

A number of formal techniques have proved effective in finding pathological features in specifications of hardware (and software) systems. What follows presents the essential features of such techniques and discusses how they can be applied in the case of contracts[50].

In section 5.3, we saw how contracts can be modelled as state diagrams. The verification techniques discussed here operate on Petri net representations. Petri nets were chosen because there was available software that implements these verification techniques. In what follows we show how Petri nets are constructed from initial state diagrams for a contractual scenario that we use as an example.

5.4.1 An Overview of the Verification Technique

Initially, a system is described using a number of Petri nets (Peterson 1981) that model it from different perspectives. In the case of hardware design, one Petri net is constructed for each individual component of the system or for each set of different requirements. In the case of contracts, each Petri net represents one of the parties' view of the business exchange (see section 5.4.2 for more details). Representing contracts as Petri nets is not novel and has in fact been attempted by Lee and his associates (cf. Bons *et al.* 1995). In our approach the separate nets are combined into one large Petri net using a composition algebra (see section 5.4.3 for more details). The final Petri net is likely to be too large and complicated to have been reliably designed and safely input by hand. The composition and every other process mentioned in this section, is fully automated. The only human inputs to the system are descriptions of the initial Petri nets and some behavioural requirements. More information on the use of this composition algebra in micro-electronic design can be found in (Zimmer *et al.* 1992) and (Zimmer & McDonald 1993).

The composed Petri net is processed in order to construct a model of its possible states (see section 5.4.4). Behavioural requirements for the model are then expressed in a species of temporal logic and the state model is queried (see sections 5.4.5 and 5.4.6 for details). This process is called *model checking* ((Holzmann 1998); (McMillan 1993)). Temporal logic enables the expression of properties about all future times from initialisation. Examples of the

[50] A more general version of the discussion in this section can be found in (Zimmer, Daskalopulu & Hunter 1999).

kinds of questions that can be expressed are “Is there a possible future in which a given request will never be answered?” or “Can the contract ever be in a position in which one party must perform two conflicting actions?”. Model checking systems applied to hardware design are reasonably efficient and have reported some notable successes, including the discovery (in minutes) of bugs that eluded months of NASA testing (Holzmann 1998).

The verification models and processes involved in model checking are explained in the sections that follow in terms of a contractual example.

5.4.2 Specification of a Sales Contract through State Diagrams

This section considers a contractual scenario in which a seller interacts with a purchaser[51]. The parties agree that delivery of goods will happen when and if the purchaser can take such delivery. The parties’ contract stipulates that the business exchange will operate in two phases. The first phase concerns the transfer of goods from the seller to the purchaser. The second phase concerns the transfer of funds from the purchaser to the seller in payment of the goods that were exchanged.

The parties agree that the first phase of the transaction will operate as follows: The seller can indicate that he has goods available for delivery (GAV). The purchaser can indicate that he requests goods (RFG) and that he has taken delivery of goods (GAC). The required sequence of events for the transfer of goods from the seller to the purchaser is as follows: GAV, RFG and GAC are initially false (denoted as GAV_F, RFG_F, GAC_F), that is, the seller has no goods available for delivery, the purchaser does not request any goods and has not accepted any goods. When the purchaser requests goods (RFG_T) the seller may assert that goods are available (GAV_T) and the request for goods is discharged (RFG_F). When the purchaser has taken the specified quantity of goods GAC becomes true (GAC_T) and only then may the seller assert that there are no more goods available (GAV_F). Finally the purchaser may set GAC back to false and the cycle may repeat.

The second phase of the transaction, where funds are transferred between the purchaser and the seller will operate as follows: The purchaser can indicate that he has funds ready for payment (FAV). The seller can indicate that he requests funds (i.e. that he requests payment) (RFF) and that he has received funds (FAC). The reader may notice that the propositions

[51] This is a variation of the multi-party trading procedure described in (Zimmer, Daskalopulu & Hunter 1999) where a single seller interacts with multiple purchasers.

indicating availability of funds, request for funds and acceptance of funds are of the same form as those indicating availability of goods, request for goods and acceptance of goods. Indeed the sequence of events for the second phase of the transaction is of the same form as the sequence of events for the first phase of the transaction. What follows concentrates therefore on the first phase of the transaction but is applicable to the second phase as well.

The state diagram in Figure 5–4 illustrates the seller's view of the first phase of the transaction. The seller is initially in state S_0 (which corresponds to his being not ready for the transaction, for example because he has no goods available). When the seller is ready (in state S_1), if he receives a request for goods from the purchaser he moves to a state where he starts the sale (S_2). When the goods have been delivered to the purchaser (that is, from the point of view of the purchaser when he has accepted them, GAC_T) the seller is in state S_3, where the sale has finished and he may end the first phase of the transaction.

The state diagram in Figure 5–5 illustrates the purchaser's view of the first phase of the transaction. The purchaser is initially in state P_0 (not ready for the transaction, for example because he does not require any goods). When the purchaser is ready and requests goods (state P_1) if goods are available (GAV_T) he moves to a state where he starts the purchase (P_2). When the goods have been delivered the purchaser is in state P_3 where he has finished his purchase and accepted the goods.

The initial specification of our example contract was given in terms of state diagrams. The verification techniques discussed in this section operate on specifications given as Petri nets, which are generalisations of state machines.

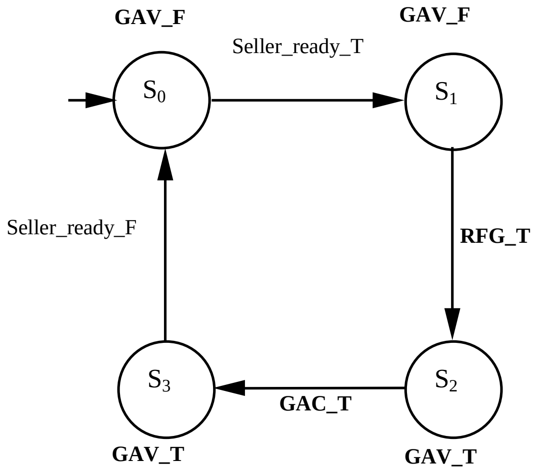


Figure 5-4 The seller's view of the contract

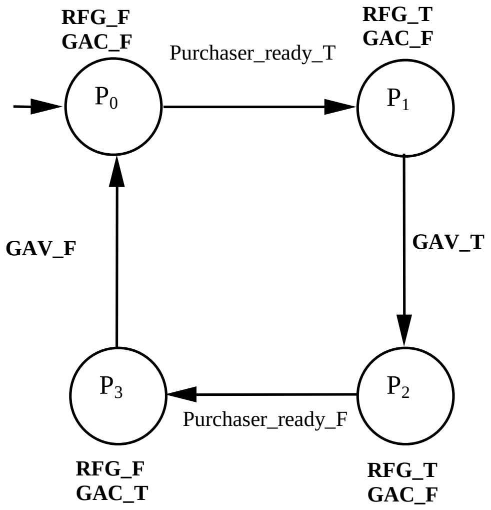


Figure 5-5 The purchaser's view of the contract

5.4.3 Petri Nets and their Composition

A Petri net is essentially a bi-partite directed graph, which comprises two sets of nodes: a set of places *P* (represented as circles) and a set of transitions *T* (represented as bars). Arcs connect places to transitions or transitions to places (but not transitions to transitions or places to places). The dynamic behaviour of the system being modelled is represented by tokens (shown as dots) flowing through the net. Each transition in a Petri net has certain requirements (preconditions) and effects (postconditions). At any given time, each place of a net may or may not have a token in it. Places containing tokens are said to be *marked.* A *marking* of a Petri net is a snapshot of the net that indicates what commodities are available at any given time. A transition is said to be *fireable* in a given marking if all of its preconditions are marked. When such a transition fires, its preconditions are unmarked and its postconditions become marked. Dynamic behaviour in a Petri net is obtained as a sequence of markings, each derived by firing one or all of the fireable transitions from the previous marking.

Petri nets are usually given pictorially as in the following example, where the single transition has preconditions *p1* and *p2* and postconditions *p1* and *p3.* The transition is fireable when *p1* and *p2* become true (in their respective places) and as a result it renders *p1* and *p3* true.

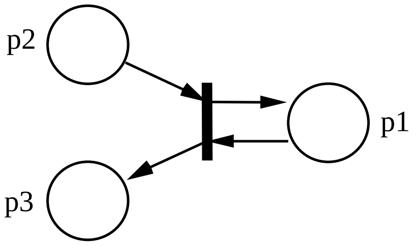


Figure 5-6 Example Petri Net with three places and one transition

There are various flavours of Petri nets, allowing for example transitions to have outputs (Genrich & Lautenbach 1981). Lee and his associates (cf. Bons *et al.* (1995) have introduced another variant, called *Documentary Petri Nets*, for modelling trade procedures. In Documentary Petri Nets some of the places are distinguished (and represented as squares rather than circles) to denote documents—Bons *et al.* find this a useful distinction for their method of composing different views of the same transaction.

In the context of the techniques discussed here Petri nets are composed quite differently, so we restrict the discussion to simple Petri nets. However, it should be noted that all the mathematical results and software alluded to here have variants for several different flavours of Petri nets. Moreover, since Petri nets are a generalisation of finite state machines, the techniques described here are directly applicable to any modelling technique based on finite state machines.

Composition of Petri nets is conducted through labelling them by naming some of the transitions with elements from a labelling set *L*. The labels provide synchronisation information (those transitions in the individual nets without labels are not available for synchronisation). The composed Petri net may contain unlabelled transitions, which cannot be synchronised with any transitions in other component nets. The synchronisation information used in the composition is given by partial functions from an event set *E* to the various labelling sets. A transition *t* from one Petri net synchronises with a transition *t'* from another, if the labels of the two transitions are both mapped to by the same event in the event set *E*. The seller's state diagram that we saw in Figure 5–4 translates to the Petri net illustrated in Figure 5–7. The purchaser's state diagram that we saw in Figure 5–5 corresponds to the Petri net shown in Figure 5–8.

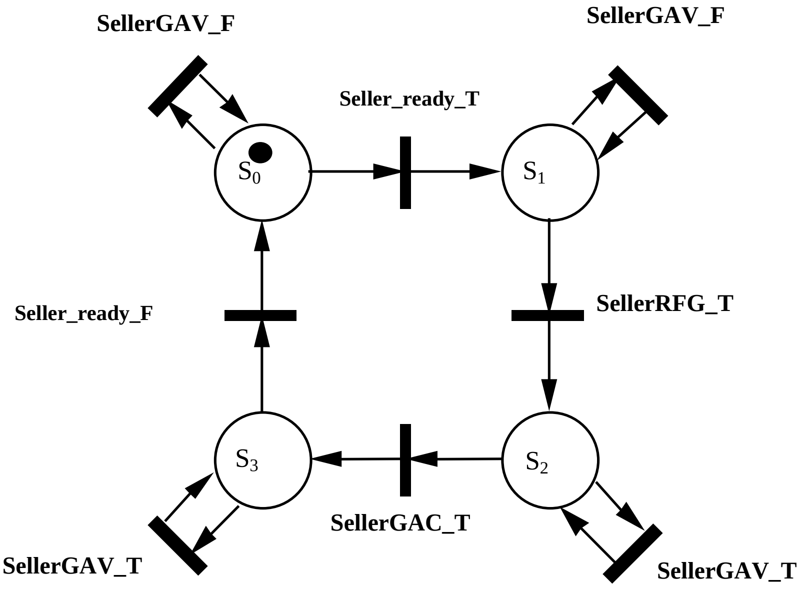


Figure 5-7 The seller's Petri net of the transaction

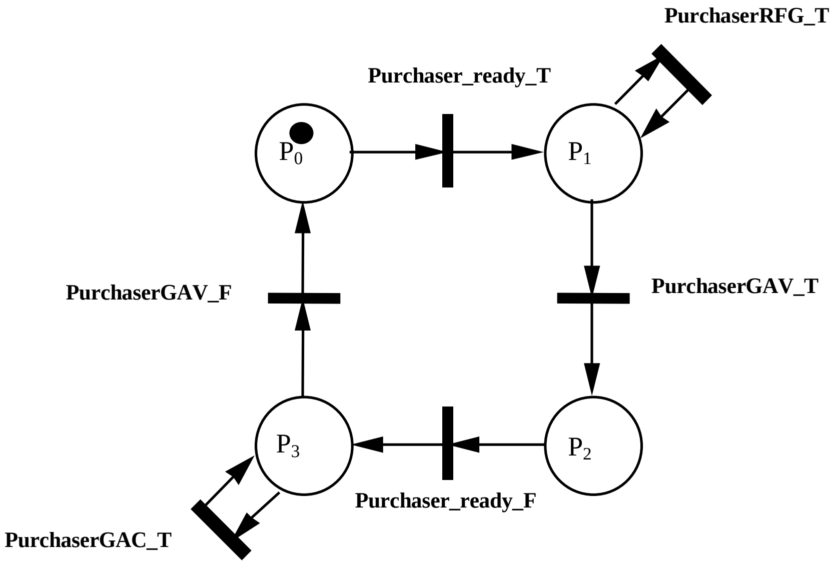


Figure 5-8 The purchaser's Petri Net of the transaction

The self-loops on some of the places are necessary in synchronisations. The specification of the transaction is completed by adding the synchronisation information, that is, by providing an event set and the information about how each event is viewed in each net.

In this case the event set is:

$$E = \{RFG, GAC, P\_GAV\_T, P\_GAV\_F\}$$

The events are "request for goods", "goods accepted" and goods becoming available (P_GAV_T) or unavailable (P_GAV_F) to the purchaser. The information about each of the parties' view of these events is given by the following partial functions, which map events (shown on the left of the $\mapsto$ symbol) to transitions in each of the Petri nets that model each party's view of the transaction:

Seller:

$$\begin{aligned} &RFG \mapsto SellerRFG\_T \\ &GAC \mapsto SellerGAC\_T \\ &P\_GAV\_T \mapsto SellerGAV\_T \\ &P\_GAV\_F \mapsto SellerGAV\_F \end{aligned}$$

Purchaser:

$$\begin{aligned}&RFG \mapsto PurchaserRFG\_T\\&GAC \mapsto PurchaserGAC\_T\\&P\_GAV\_T \mapsto PurchaserGAV\_T\\&P\_GAV\_F \mapsto PurchaserGAV\_F\end{aligned}$$

The composed Petri net with one purchaser and one seller is illustrated in Figure 5–11.

The composition allows us to combine more than two component Petri nets (say, in the case of multi-party transactions) but this example is intended to illustrate how difficult it would be to describe directly the resulting Petri net for more components.

In a composed Petri net there are two kinds of transitions: some are transitions from the component Petri nets, for which there is no synchronisation information. Such transitions are merely carried over from the component nets to the composed net. The other transitions are the result of synchronising one transition from each component Petri net, in accordance to the event set. Consider for instance the two component Petri nets in Figure 5–9 and suppose that the synchronisation information available is that an event *x* synchronises transitions *a1* and *b1*, while there is no synchronisation information for transition *b2*. That is, the event set is $E = \{x\}$ and we have the following partial functions defined for each component Petri net:

A: $x \mapsto a1$

B: $x \mapsto b1$

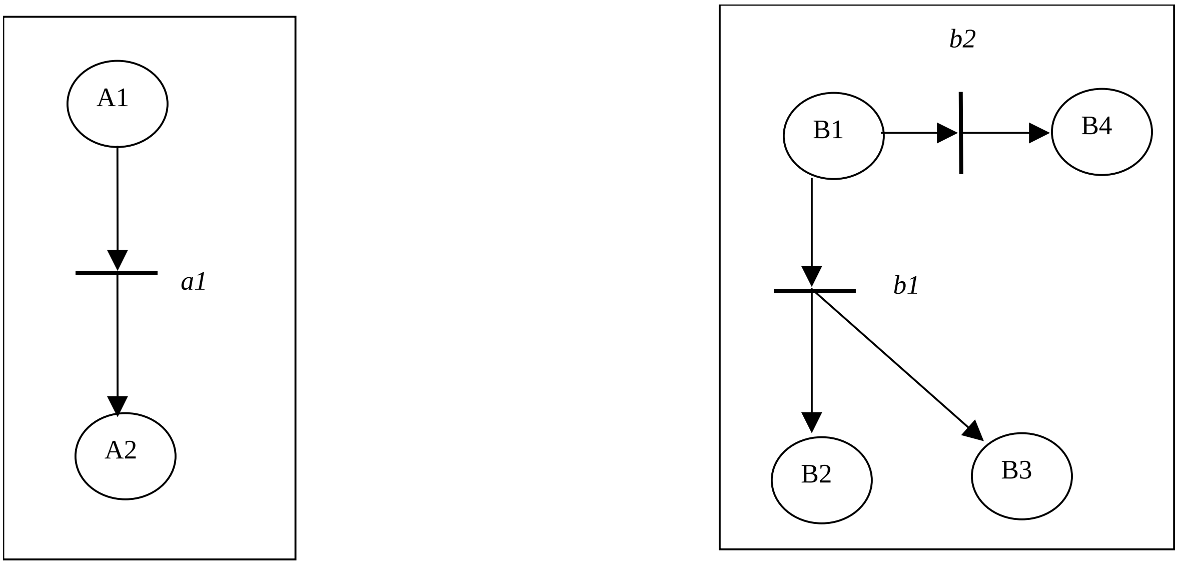


Figure 5-9 Example Component Petri Nets (A and B)

In their composition, which is shown in Figure 5–10, the transition marked (*a1, b1)* corresponds to the two synchronised transitions of the corresponding component Petri nets, while the transition *b2* is carried over from Petri net B. A composed transition such as *(a1, b1)* has as input and output places all the input and output places of its component transitions, as can be inferred from the example in Figure 5–10.

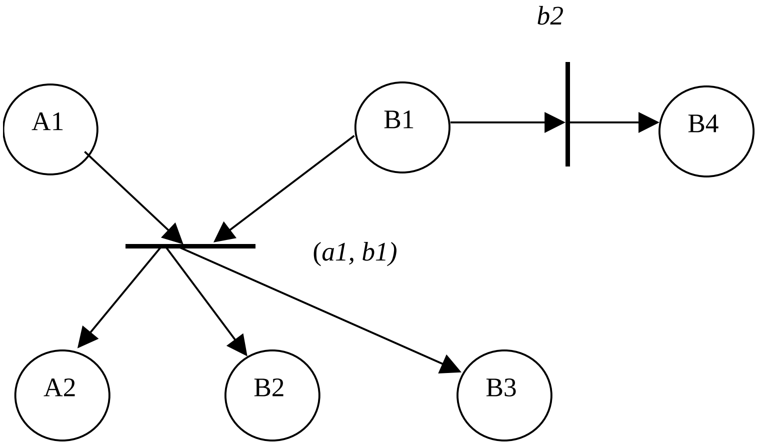


Figure 5-10 The Composition of Petri Nets A and B

More details on the composition of Petri nets can be found in (Zimmer & McDonald 1993) and (Zimmer *et al.* 1992).

5.4.4 State Models

State models (from a systems modelling perspective) are directed graphs or finite state machines without the inputs, that show the possible state changes of a system. State models are generated as the markings of Petri nets. In our example, the initial marking has S_0 and P_0 marked in the individual Petri nets of seller and purchaser. This leads to the state model shown in Figure 5–12.

If we had a multi-party scenario, the state model would not have been deterministic. The next stage of verification is to test the truth of various conditions of the state model. The logics used in model checking are all species of propositional temporal logic. The most frequently used is called *Computation Tree Logic* (or CTL) (Clarke *et al.* 1986).

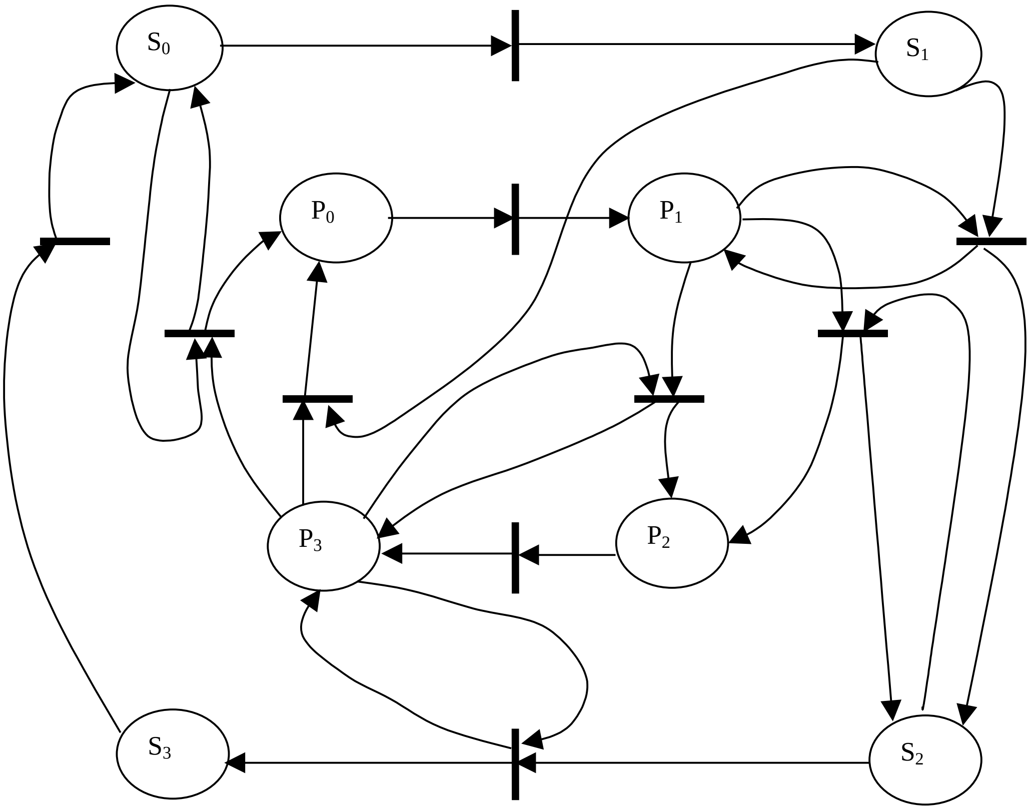


Figure 5-11 Composed Petri Net of seller and purchaser

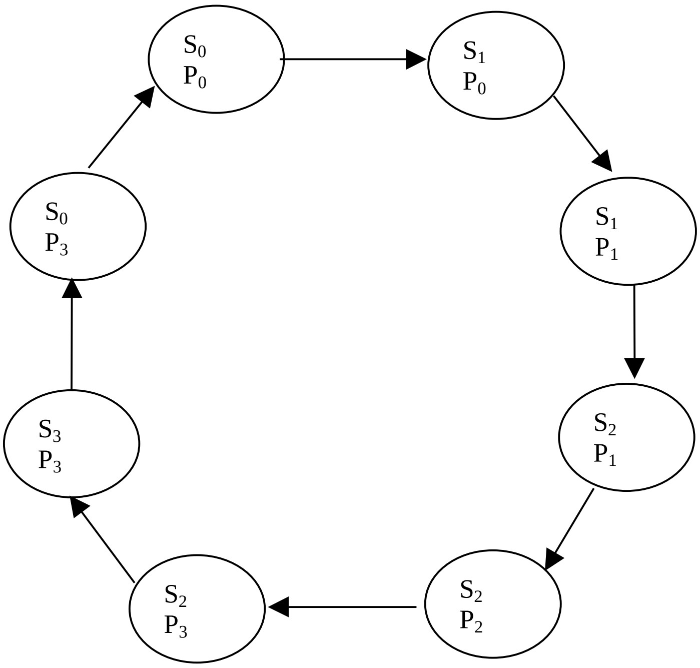


Figure 5-12 State model of composed Petri net

5.4.5 Computation Tree Logic

Well-formed formulae in CTL are atomic propositional ones or compounds formed by using logical connectives and some temporal operators. Semantic relations are defined between states and formulae. Expressions of the form $s \models \varphi$, where s is a state and φ is a formula, mean that φ is true in state s. The definition of $\models$ is as follows:

$s \models \neg\varphi$	Iff s does not satisfy φ.
$s \models \varphi \vee \psi$	Iff s satisfies at least one of φ and ψ.
$s \models \varphi \wedge \psi$	Iff s satisfies both of φ and ψ.
$s \models \varphi \rightarrow \psi$	Iff $s \models (\neg\varphi \vee \psi)$.
$s \models \mathrm{E}(\varphi)$	Iff in some immediate successor state of s, φ holds. That is, it is possible for φ to hold in the next state
$s \models \mathrm{A}(\varphi)$	Iff in all immediate successor states of s, φ holds. That is, it is necessary that φ will hold in the next state.
$s \models \mathrm{F}(\varphi)$	Iff in some future state reachable from s, φ holds. That is, φ holds eventually.
$s \models \mathrm{G}(\varphi)$	Iff in all future states reachable from s, φ holds. That is, φ holds (globally) for ever after s.
$s \models (\varphi\ \mathrm{Au}\ \omega)$	On all paths from s, φ holds at all states until the first state in which ω holds (ω is not assumed to ever hold).
$s \models (\varphi\ \mathrm{Eu}\ \omega)$	On some maximal path from s, φ holds at all states until the first state in which ω holds (ω is not assumed to ever hold). A maximal path is either a path that comes to a dead-end or an infinite path.
$s \models (\varphi\ \mathrm{EU}\ \omega)$	On some path from s, φ holds at all states until the first state in which ω holds (ω is assumed to eventually hold).
$s \models (\varphi\ \mathrm{AU}\ \omega)$	On all paths from s, φ holds at all states until the first state in which ω holds (ω must eventually hold on every maximal path).

Table 5-2 Semantics in CTL

The difference between u and U is subtle. Consider the state model of Figure 5–13, where letters outside states are the names of the states and letters inside the states represent properties that are true in the states.

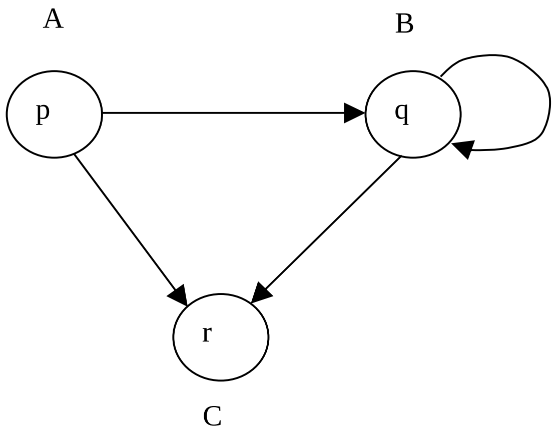


Figure 5-13 Example state model

There are infinitely many maximal paths from A: there is a length-one path to C; there are paths from A to B staying in B for a number of cycles and then moving to C; and there is an infinite path that goes to B and stays there by repeatedly going round the loop. On all of those paths, q remains true until r is true. That is:

$A \models (q \text{ Au } r)$

On the infinite path r is never true, therefore:

$A \models \neg(q \text{ AU } r)$

There are some paths in which r is eventually true:

$A \models (q \text{ EU } r)$

The distinction between u and U does not show up in our example. However, when more than two parties are admitted (say one more purchaser) the distinction does show up: a purchaser making a request for goods will persist until he is satisfied. However, with two purchasers, if the contract stipulates that the sale starts when all purchasers are ready to take goods, and the second purchaser never requests goods, then the first purchaser's request will never be satisfied.

5.4.6 Verification

One condition, a form of liveness that we might wish to check, is that from every state the transaction can eventually return to its initial state. In other words, we wish to establish that the transaction between seller and purchaser will eventually terminate (that the contract between them will at some point be discharged). The condition is expressed below, where B is the initial state:

$$B \models G(EF(S_0 \wedge AP_0))$$

Some safety conditions we have also checked for the example presented here are:

- The seller should assert GAV_T only when the purchaser is ready (in state P_1). In other words, we wish to establish whether there is a situation in which the seller has goods available for delivery but the transaction cannot be realised because the purchaser is not ready to accept delivery:

$$B \models G((\neg \text{SellerGAV\_T} \wedge \text{ESellerGAV\_T}) \rightarrow P_1)$$

- Once GAV is true, it should remain true until the purchaser has accepted goods (that is, the purchaser is in state P_3). In other words, we want to establish whether it is possible for the seller to retract goods, once the transaction has started, before they have been accepted by the purchaser:

$$B \models G(\text{SellerGAV\_T} \rightarrow (\text{SellerGAV\_TAUP}_3))$$

- The purchaser should not accept goods until GAV is true, that is, the purchaser should not enter P_2 until GAV_T. In other words, we want to establish that there is no situation in which the purchaser's obligation to pay for goods is active before such goods are available and delivered to him:

$$B \models G((\neg P_3 \wedge EP_3) \rightarrow \text{SellerGAV\_T})$$

These liveness and safety conditions were checked and all hold for the contract defined in our example.

Checking simple conditions (that involve a state and its immediate successor) is straightforward and computable; the algorithm involves looking ahead at most one step (Zimmer & McDonald 1993). For each of the more complicated temporal constructs, the model checking algorithm uses iterative search. The method is computable because it can be proven that these iterative techniques all finish after a number of steps bounded by the length of the longest non-looping path in the state space. The model checking algorithm keeps performing iterations until the same result is reached twice (*fixed point*). It can be proven that a fixed point can be reached in bounded time. However, the bound on time is the number of states of the graph, which is at worst exponential on the number of variables. This is called the *state explosion problem*. The example used in this section is relatively small and the number of states was manageable. In the general case, the state model is transformed into another model called a *binary decision diagram* (BDD). The BDD is still exponential in the number of variables but it admits reduction techniques to yield a *reduced ordered binary decision diagram* (ROBDD), which is usually considerably smaller than the original state space[52]. The ROBDD is then queried using a species of temporal logic[53]. More details on the transformation of state spaces into ROBDDs can be found in (MacMillan 1993), and a shorter discussion is in (Zimmer, Daskalopulu & Hunter 1999).

5.4.7 Discussion

Model checking has proved to be a powerful technique for verifying high-level behaviour of hardware systems. The example that was discussed in this section showed how the technique can be applied in a contractual setting to check a business transaction (that comes about after two parties have entered a contract) against some temporal conditions. As we saw, existing model checkers operate on two specifications: one, which is the operational specification of system behaviour expressed as Petri nets and the other, which is a declarative specification of behavioural requirements expressed in temporal logic. In a contractual setting, the system being modelled is the agreement. The specification of its behaviour emerges by taking an operational view of the agreement, expressed using Petri nets, as illustrated by the example. Behavioural requirements are requirements about the operational view of the agreement and these can be expressed in temporal logic. By representing agreements in this way, existing formal methods for the verification of hardware and software systems are directly applicable to the verification of business transactions.

For the example discussed in this section bespoke software developed by Zimmer & McDonald (1993) was used to construct the composed Petri net and to check it. This choice

[52] The reduction techniques have been used to reason effectively with systems with more than 10^{20} states. (Burch *et al.* 1992).

was primarily made because these tools were readily available. Any standard CTL-based model checker would be suitable, such as CheckOff, which is currently marketed by Siemens[54]. The techniques described here were also tried on examples of trade procedures discussed in Lee (1999). Of those, the one that appears to be the most complex concerns a sales contract between a Buyer and a Seller, where a Bank provides a letter of credit and a Carrier is subcontracted to deliver the goods. The abstract view of the transactions between the four parties of the trade procedure is illustrated in Figure 5–14:

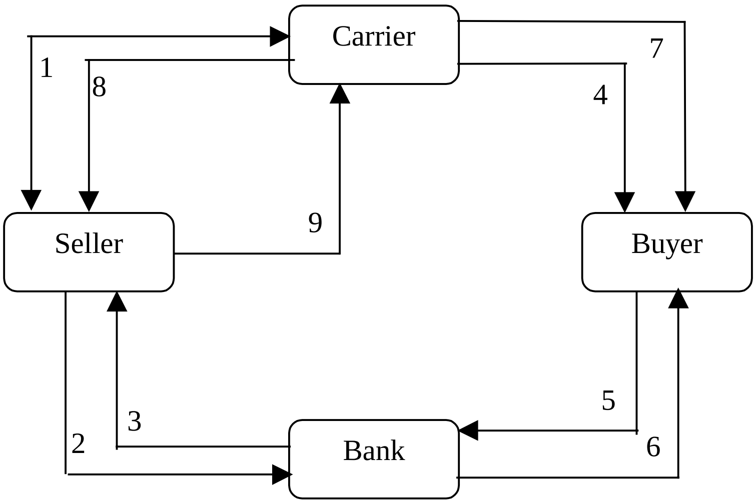


Figure 5-14 Sales Contract with Carrier Sub-Contracting and Letter of Credit (adapted from (Lee 1999))

The various transactions that take place between the parties involved in the trade procedure are as follows:

1. Seller makes goods available to Carrier for dispatch. Carrier provides Seller with certificate that goods have been dispatched.
2. Seller supplies Bank with certificate that goods have been dispatched and certificate that the Buyer owns the goods.
3. Bank pays Seller the required price for goods.
4. Carrier notifies Buyer that goods have arrived.
5. Buyer instructs Bank to pay the required price and interest.
6. Bank sends to Buyer certificate that he owns the goods.

[53] In our example we used CTL but other model checkers operate on different temporal logics such as interval temporal logic or linear temporal logic.

[54] In fact CheckOff has been tried on larger specifications for hardware and was faster than the model checker developed by Zimmer & McDonald.

7. Buyer presents Carrier with certificate that he owns the goods, Carrier gives the goods to Buyer, and Buyer gives Carrier receipt.
8. Carrier provides Seller with receipt issued by Buyer.
9. Seller pays Carrier the agreed price for delivery.

The similarity between this seemingly complex example and the simple scenario that we used in this section should be obvious. In the simple case, we considered only the part of a trade procedure that is relevant to the transfer of goods. In Lee's example, there are goods, funds and various certificates that are exchanged between parties. A Petri net can be constructed showing each of the four participants' views of the aspects of the transaction that are relevant to him and these can be composed and checked against behavioural requirements (Lee lists some such requirements, for instance that transition 2 in the list above must happen before transition 3).

The techniques discussed here were also applied to a few trade procedures that we extracted from our sample British Gas contracts. Our experimentation suggested that the larger the trade procedure that is modelled, the more individual Petri nets need to be formulated so that they can be subsequently composed using the Petri net composition tool. The difficulty in this process lies in establishing the procedure, aspects of which may be described in different parts of a large contractual document, rather than modelling and model checking it. It is also difficult sometimes to establish the precise behavioural requirements that we would want the procedure to satisfy.

We found that the operationalisation of the transaction and its representation as Petri nets conceals the distinction between ideal and actual behaviour. Obligatory, permissible or prohibited actions that parties may perform during the transaction are interpreted and incorporated in the model implicitly, rather than explicitly.

Such interpretation might be incorrect or incomplete (for example a safety or liveness condition might be omitted or misinterpreted). The verification process establishes whether a given operational specification is correct against a declarative one. It does not however address whether *each* of the two specifications is correct or complete.

5.5 Summary

This chapter considered some possibilities for representing contracts at the micro-level. First, we concentrated on contractual relations as the outcome of promissory exchanges between parties. Kimbrough's (1998) speech-act theoretic framework for representing promises, assertions, requests, and other performative statements using event semantics was extended so that the illocutionary effects of such statements can be shown explicitly as the postconditions of their corresponding events. In this way, the obligations assumed by parties entering contractual relations can be tied to the promises and assertions of a promissory nature that they extended.

We then concentrated on obligation relations between parties and discussed how contracts can be modelled as collections of interrelated obligations using state diagrams. In such diagrams, each state corresponds to a (possibly incomplete) view of the status of an agreement during its life-span. We introduced an informal notation for describing paths between states and showed how relative or absolute time can be incorporated in the representation enabling us to determine what obligations hold given an event or a sequence of events brought about by the parties' actions. In this way, contracts are modelled as dynamic systems. Dynamic systems raise interesting questions about their safety and liveness properties. This led us to the third discussion contained in this chapter, in which we concentrated on the temporal verification of contractual provisions. We explained how model checking, a technique which has been successful for the verification of software and hardware specifications, can be applied to verify trade protocols defined by contracts.

6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions and Contributions

This dissertation explored the potential for developing logic-based tools for the analysis and representation of legal contracts. The research was motivated by the fact that despite the similarity in nature and function between legislation and legal contracts, a significant volume of Artificial Intelligence research has addressed the former while the latter domain has been largely overlooked. We found this surprising given the close relationship between legal contracts and legislation. As was explained in the beginning of the dissertation, contracts are putative entities that lie within the scope of legislation and have similar structure, function and even linguistic expression to that of legislation. The general contribution of this research is therefore that it provides a study on the potential for the application of Artificial Intelligence techniques to the domain of legal contracts. This general contribution is supported by a number of specific ones, which are discussed later.

Anne Gardner (1987) concentrated on contract formation rules as her case study in developing a framework for the representation of legal rules informed by jurisprudence. Her work was still concerned with legislation about the nature of exchanges that lead to contractual relations, rather than legal contracts themselves. The ALDUS project (1992) explored the potential for developing tools to support contract drafting, focusing on sales contracts, and concluded that there was no real need for such efforts because the contents of contracts are straightforward and the activity associated with them is simple.

We contended that the conclusions of the ALDUS project were perhaps correct insofar as one concentrated on contracts at the simple end of the scale such as standardised sales agreements which are often regulated by relevant legislation. In contrast, this research concentrated on contracts in engineering areas. We examined sample documents provided by our industrial collaborators and model form contracts issued by professional bodies, such as the Institution of Electrical Engineers, and found them to be lengthy documents with a mass of cross-referenced and complicated detail that is difficult to follow and to apply in specific cases. Our investigation highlighted three broad areas where automated support would be desirable to facilitate contractual activity, namely management, administration and drafting.

This research focused mainly on the third of these areas. As was explained in chapter 1, this choice was motivated by two reasons. First, of the three areas of contractual activity that were highlighted, contract drafting is in practice the one associated with the highest costs (in time and labour). We have anecdotal evidence from our industrial collaborators at British Gas that there are engineering projects where the associated contracts are an order of magnitude larger than the ones that we examined, yielding drafting costs that account for a significant proportion of the total cost of the project. Second, we contended that a framework for representing contracts for the purpose of drafting provides a natural basis on which to develop extensions to support contract management and administration. Put alternatively, in seeking a representation of contracts for the purpose of drafting, their structure and content become prominent. Appropriate query schemas can then be defined in order to support contract management and mechanisms for "executing" the representation can be defined in order to support contract administration.

The problem of drafting itself was addressed by exploiting the dual meaning of the term "contract", which is used to refer both to a legally binding agreement and to the document that records such an agreement. We argued that drafting can be addressed at two levels: At the *macro-level* drafting can be viewed as a variant of a Computer-Aided Design process, whereby a document is constructed from its constituent parts with emphasis on structure and overall syntactic coherence. At the *micro-level* drafting is the formulation of individual detailed provisions so that the whole is complete, that is, it caters for all the eventualities that the drafters intend to cover, and consistent, that is, it does not give rise to conflicting provisions that apply in the same circumstances.

As regards macro-level drafting, the idea that a document can be assembled from its parts is not itself novel. It had been expressed by other researchers (for example (Fiedler 1985); (Gordon 1992)) before this work. However it had not been exploited and developed into a practical possibility and this is where the first specific contribution of this research lies.

Previous work on the drafting of legal documents since 1980 had given rise to a number of systems, some of which have evolved over the years and are available commercially today. Our review of the major ones (such as the ABF processor, Scrivener, CAPS and its variants) in chapter 4 showed that they rely on encoding standardised text along with the mechanisms that are used to manipulate it in a procedural manner. Although these systems have proven useful for relatively short and not very complex documents (most notably for wills) we argued that they are not satisfactory for the long and complex documents that we are interested in. Although the most modern of those systems model explicitly the syntactic structure (in terms of containment relations between document components) of the document types that they support, they do not model functional dependencies between document components, such as necessary inclusion, exclusion and so on. The drafter's initiative in constructing a new document is restricted to the choices already supported by the system. If the available document samples do not suit the circumstances at hand, the drafter must familiarize himself with the underlying representation scheme of the system to encode a new document type or have the system upgraded by its programmers or merely not use the system and resort to drafting a new document from scratch using a conventional word-processor. We argued that, apart from modelling explicitly the organisation of a document in terms of containment relations between its components, we also need to model explicitly functional dependencies between such components, which can be used to constrain the CAD-like assembly of a new document so that the whole is syntactically coherent.

Drawing upon work into the syntactic structure of text from the Electronic Publishing Community and the functional organisation of text from the Computational Linguistics community we constructed a prototype system, MODELLER, that is intended to support contract assembly at the macro-level.

The main features of MODELLER, which were noted earlier in the dissertation, are summarised again here:

(i) Generic documents, that is, document types rather than individual instances, are represented as collections of re-usable components. Each component corresponds to a portion of text and may have multiple versions, that is, it may correspond to multiple variants of portions of text. Generic documents are dynamic: they are initially defined based on model forms or previous examples for their corresponding document type, but drafters can create new versions of components during the drafting session and these are automatically incorporated in the associated generic document.

(ii) Syntactic relations between document components (such as containment relations) and functional dependencies between them (such as necessary inclusion, exclusion, choice and so on) are represented explicitly and constrain the drafting process.

(iii) Individual documents of a given type are created by instantiating a generic document for that type, interactively with the drafter, who chooses what components to include in the document instance and what version of such components. This renders the drafting process precedent-based. The drafter's choices are constrained by the syntactic relations and functional dependencies between document components. The drafter negotiates a document instance with the system and his choices are critiqued by the system in the spirit of Hammond and his associates' work on the design of dental prostheses (Hammond *et al.* 1993). The drafter however is not limited to merely choosing one of the available versions for each component. Rather, if none of the available versions is appropriate for the circumstances at hand, he may create new ones, which become incorporated in the generic document and are available for future use. A useful feature of MODELLER in this respect is that it offers drafters the facility to record comments that are associated with individual versions of document components. Such commentary is helpful in determining why a particular component was included in a given document instance or why a particular version of a document component was preferred over other available versions.

(iv) As document instances for a given type may share versions of components, they are not represented and stored explicitly as text. Rather document instances are represented and stored as collections of indices to the text files of their components. Apart from avoiding unnecessary replication of data in memory, this allows for greater flexibility in document instance maintenance and administration. If, because of organizational or legislative changes, a portion of contractual text needs to be re-worded, only the corresponding document component needs to be altered and all document instances sharing it are ensured to be updated. This form of representing and storing document instances also allows for their full text to be re-constructed in a variety of forms (for example as plain text, or SGML, or HTML marked up) through separate mechanisms that manipulate the representation.

MODELLER is thus more flexible and general a framework than previous systems for legal document drafting, as it enables drafters to have more control over the document instance that they are creating rather than restricting them to merely using boilerplate text. Although it was primarily constructed with complex engineering contractual documents in mind, the approach can be applied to the relatively shorter and less complex documents that such previous systems addressed—as Appendix B illustrates MODELLER was tried on documents of variable size and complexity. Moreover, in chapter 4 we showed how the MODELLER framework for representing contractual documents for the purposes of drafting can be extended with database retrieval mechanisms to support contract management.

As was noted in chapter 4, researchers such as Bench-Capon & Dunne (1989) and Koo (1989) have argued in favour of computer document models that capture explicitly the structural conventions introduced by authors because this enhances the readers' understanding of document content. Koo in particular noted that in this way production of a document is facilitated as the functionality and extensibility of a document management system is enhanced and problems of collaborative authorship can be solved. The approach employed in MODELLER is in response to and supports this argument.

The successful evaluation of the prototype version of MODELLER by our industrial collaborators and the favourable reaction of other researchers in the area (most notably Branting and his associates) to its underlying assumptions and principle were particularly welcome.

We now turn to the representation of contracts at the micro-level, that is, the detailed representation of individual provisions. Micro-level representation of contractual content is (not surprisingly) more complicated than macro-level representation of contractual documents and requires examining and drawing upon different areas of research. The dissertation noted several reasons that render this enterprise complex, which are summarised here:

(i) In addressing the representation of contractual content it is difficult to separate issues relating to the representation of agreements, from issues relating to the representation of the (textual) contents of documents that record such agreements. It is convenient some times for the sake of clarity to view contract formation as conducted at two phases: negotiation, where the agreement is designed, and drafting, where the designed agreement is recorded. These two phases have been viewed as parallel to software specification and software implementation (Bench-Capon 1987). However, in practice, negotiation of long-term agreements, such as the ones in which we are interested in here, is carried out on the basis of draft documents and is often about the documents (for example, the particular wording employed in certain provisions).

(ii) As our discussion on Contract Law in chapter 2 noted, it is not necessarily the case that contractual documents contain all of the terms of an agreement. In areas that are regulated by legislative Acts, additional terms are implied into agreements. A micro-level representation of contractual content based on analysing the contents of documents is therefore not ensured to be complete unless such relevant legislation is represented as well. Moreover, it is a common assumption that contractual documents contain all that the parties agreed (and, as we saw in chapter 2, Contract Law encourages this view, since in the case of written agreements the courts apply the *parol evidence rule* when disputes arise between the parties). In practice however, as our examination of sample contracts and the comments of our industrial collaborators indicated, contractual documents often leave many details only vaguely specified, especially when the transaction between parties is long-term and it is practically impossible to foresee all eventualities that might arise. There are also cases, where although some eventualities can be foreseen at the time of forming an agreement, parties prefer not to commit themselves to any particular course of action and defer resolving such issues as and when they arise.

(iii) As our initial discussion on engineering contracts in chapter 2 noted, although Contract Law concentrates on obligations as the essential characteristic of contractual content, our investigation highlighted diverse contractual provisions that serve different purposes. Apart from prescriptive provisions that specify obligations for contracting parties, we noted that some provisions supply definitions for various terms or formulae that assist in establishing values for various parameters, some specify procedures that are to be followed when the exchange is in operation, and some determine the conditions under which other provisions apply. The difference in nature and function served by different provisions suggests that not all of them need to be represented explicitly for all

> kinds of contractual activity. Moreover, they cannot be represented in the same manner. For example, logic-programming representations, which have proved useful in the domain of legislation, are appropriate for the representation of definitional provisions. However, they are not satisfactory for the representation of prescriptive provisions, for they conceal the distinction between the actual ("what is the case") with the ideal ("what ought to be the case") and do not account for possible violations of norms. We found that definitional provisions in contracts are themselves not worth representing explicitly. This is because, unlike classification norms that are frequently encountered in legislation, they are not sufficiently complex to warrant the effort of constructing representations and questions about open-textured concepts tend to dominate.

The dissertation concentrated on logic-based techniques that have been used successfully for the detailed representation of legislation and regulations, in order to establish the extent to which they can be transferred to the domain of legal contracts. In chapter 3 we considered first symbolic logic in its general form, and Allen's proposal that it be used as the basis for a legal drafting language in order to avoid inadvertent ambiguity that is inherent in natural language. Allen's work concentrates on syntactic ambiguity that arises from the use of connectives in compound sentences. We did not encounter many such problems in the text of the sample contracts that we examined. Rather we encountered vagueness arising from open-textured concepts. Moreover, as was noted in chapter 3, logic programming can be used for the purpose of syntactic disambiguation. Kowalski's (1995) version of the Michigan lease termination clause—an example used by Allen—illustrated that logic programming is computationally more powerful (albeit less expressive) than Allen's proposal as logically implausible interpretations are omitted from the space of possible interpretations. However as was noted earlier, logic programming, although very useful for the representation of complex legislative and regulatory provisions, is not appropriate for the detailed representation of contractual provisions. The definitional provisions found in our sample contracts are not complex enough to warrant the effort of constructing logic programming representations and questions of vagueness and open texture dominate. Moreover, although logic programming is sufficient for the representation of prescriptive provisions as normative propositions (that is, as statements defining prescriptions), it does not allow reasoning about and with prescriptions. Since legal contracts abound with notions such as "duty", "right", "power" and so on we examined deontic logic. Chapter 2 presented a semi-formal account of Hohfeld's (1913) explication of these notions and the discussion was continued in chapter 3, with Sergot's extension of the Kanger-Lindahl theory of Normative Positions. This theory combines deontic logic and the logic of action/agency to account for Hohfeld's "fundamental legal conceptions". In its generalised and automated form, it is intended to assist in clarifying and expanding an incomplete and imprecise statement of requirements into a precise formal specification at some desired level of detail. Its computerised version is a powerful tool for the interactive design of sets of norms as questions about intended interpretation, structural ambiguity, and the precise nature of legal notions are brought to the foreground while ensuring overall consistency of the norms under construction. Moreover it can be integrated with the theoretical framework developed by Alchourrón & Bulygin (1971) to address questions about the completeness and independence of sets of norms, as was shown in chapter 3. Most of the prescriptive provisions found in our sample contracts are temporally qualified and checking for consistency and completeness along the temporal dimension requires the employment of some suitable temporal logic.

Chapter 5 examined the representation of contracts at the micro-level from three perspectives and contained three related discussions.

The first discussion concentrated on contractual relations as the outcome of promissory exchanges between parties. It was inspired by recent research (most notably by Kimbrough and his associates), which has been concerned with the development of formal languages for business communication based on speech act theory. Kimbrough (1998) proposes a framework for representing performative statements such as requests, promises, assertions and so on. In attempting to represent contractual content as the result of such statements exchanged between two parties, we extended his proposal in the following respects (noted in chapter 5 and summarised again here):

(i) We argued for the distinction between actual events and stipulated (future) events to be made explicit. In this way the fact that a promise or a request refers to a future, unrealized at the time of the promise, event is clearer. Whether assertions refer to actual facts (past or present events) and hence have informative nature or future events and hence have promissory nature is also made clearer.

(ii) We argued for the association of events with times, where appropriate, and the extension of Kimbrough's framework with a suitable temporal algebra, such as the event calculus. This is a necessary extension of the framework to accommodate contractual situations realistically, since the notion of time (in relative or absolute terms) is intrinsic to contracts.

(iii) We argued for the explicit representation of similarity between events, where appropriate, and the extension of Kimbrough's framework with suitable definitions for similarity, based on common sense knowledge or information provided by the contract or legislation where it is available. This is more flexible than insisting upon identity between events and enables questions about fulfilment or otherwise of promises (and consequently of requests that gave rise to such promises) to be addressed more realistically.

(iv) We argued for a generic representation of the notion of a fulfilled promise in place of Kimbrough's proposal, which associates each promise with an explicit representation of what would render it fulfilled. Our proposal stems from the observation that the notion of fulfilment can be defined abstractly in terms of the occurrence of the content of a promise independently of the particular type of such content.

(v) We argued for the extension of Kimbrough's event descriptions with preconditions and postconditions. The former may go some way towards expressing conditional performative statements (such as conditional promises). However neither our representation nor Kimbrough's provide a systematic account for such conditionals. The latter enable the association between performative statements and their effects to be come explicit. Hence, obligations that are incurred by promises or assertions of a promissory nature are highlighted and become accessible.

These improvements to Kimbrough's framework constitute the second specific contribution of this dissertation. The extended framework was used to represent extracts from sample contracts as collections of assertions, declarations and promises. However, we noted that most of the prescriptive and procedural provisions that we analysed give rise to conditional promises and as stated previously, the representation (in its original form as proposed by Kimbrough and in its extended form along our suggested improvements) does not accommodate them adequately. In our extended representation, the most that can be expressed is the association of pre- and post-conditions with promises. Further extensions to the framework to accommodate conditional promises are required.

The second discussion in chapter 5 concentrated on contractual obligations and took a process view of contractual content. In this view, at the micro-level contracts are modelled as collections of interrelated obligations using state transition diagrams. In such diagrams, each state corresponds to a (possibly incomplete) view of the status of an agreement during its life-span. We introduced an informal notation for describing paths between states and showed how relative or absolute time can be incorporated in the representation enabling us to determine what obligations hold given an event or a sequence of events brought about by the parties' actions.

The third discussion in chapter 5 concentrated on the temporal verification of contractual provisions. Other researchers in the past (most notably Lee and his associates), have proposed Petri nets as a suitable formalism for modelling contracts as trade protocols between parties (Bons *et al.* 1995) and noted the need for formal verification techniques that establish whether a trade protocol satisfies certain behavioural requirements. The third specific contribution of the dissertation lies in exploring how model checking, a technique used for the temporal verification of hardware specifications, can be successfully applied in the domain of legal contracts.

6.2 Future work

The dissertation raises a number of issues for future work both in practical and in theoretical terms.

There are several aspects in which MODELLER can be improved as was noted in chapter 4. For example, usability issues need to be considered in more detail than was afforded in the development of the prototype version and the system would be more flexible if it was integrated with commercial word-processors and hypertext browsers. Our experience with our industrial collaborators suggested that document drafting is often carried out collaboratively by a team rather than a single individual and a distributed version of the system with appropriate locking and authentication mechanisms would be desirable to support such kind of activity. Moreover, MODELLER would need to be integrated with micro-level representations of contractual content. There are several possibilities for such extensions, which were discussed in the dissertation. One possibility would be to integrate MODELLER with a Petri net composition tool in order to construct process views of the transactions stipulated in the agreement and verify them against behavioural requirements using the techniques outlined in chapter 5. The desired features of a Petri net composition tool itself, in order to model multiple transactions between parties in the course of a long-term trading agreement such as the ones we examined, is a topic worth exploring in more detail. The discussion here was limited to a simple example to illustrate the potential of model checking techniques rather than a systematic exploration of available Petri net kinds.

The recent development of XML and web-based methods for annotating documents open up interesting possibilities for electronic contracting. In particular, it would be interesting to explore the relation between the Document Object Model and other emerging standards to the assumptions and approach employed in MODELLER.

We extended Kimbrough's framework in order to represent explicitly contractual obligations as the result of exchanges of a promissory nature between parties. However, neither Kimbrough's original framework nor our extended version account adequately for conditional promises. Kimbrough (1999) presents a diagrammatic methodology for representing the meaning of performative statements exchanged between communicating agents, which is an extension of UML[55]. In this work, he proposes a schematic way for representing conditional promises and it would be interesting to pursue this further to establish appropriate semantics.

We used an informal notation based on modal languages used to describe transition systems in order to model contracts as processes. Defining semantics for transitions, especially those involving action negation and time constraints is a promising direction for future work.

We did not address contract negotiation itself in this work. One way to view contracts is as a collection of separate but interrelated sub-agreements. The parties involved have a common goal, to realise the business exchange, to co-operate, but each wants this to happen under the 'best' terms for them. What makes a particular arrangement good for a party is relevant to how it affects their broader business goals. Often the goals of the two parties are not mutually satisfiable as they stand, and revision (some mutual compromise) is required. A negotiation tool would therefore be useful if it allowed parties to specify their goals and determined whether these are satisfiable, or would be satisfiable, if certain terms were agreed; if resolution of some conflict were required, then it would be useful for the tool to indicate alternative terms (that entail change in the original set of goals). Obviously in its full generality, this is a huge problem raising a whole range of issues to which various techniques could be applied. One promising approach is to take an argumentation view of negotiation.

[55] **U**nified **M**odelling **L**anguage.

Argumentation has been researched intensively in recent years. Prakken (1997) provides an excellent account of legal argumentation in particular and also an overview of recent advances in argumentation generally. As negotiating parties argue for their own interests, their success in getting the 'best' for them relies on how *persuasive* their arguments are. A representation scheme that can model persuasive argument might prove useful in this context (for a discussion see (Reed 1998)).

Finally, there is scope for pursuing further research into the development of contract administration tools, that is, tools that monitor the parties' compliance with a given agreement. This is not a straightforward matter. It touches on fundamental problems in the field of deontic logic—contrary-to-duty obligations (for example (Prakken & Sergot 1996); (Prakken 1997)), the interplay between time and obligation (for example (van Eck 1982), the proper treatment of legal competence or power. It is an open question whether such issues need to be resolved before practical applications can be attempted.

References

Alchourrón C. E. (1993). Philosophical Foundations of Deontic Logic and the Logic of Defeasible Conditionals. In Meyer & Wieringa (eds.) *Deontic Logic in Computer Science: Normative Systems Specification*, John Wiley & Sons, Chichester.

Alchourrón C. E. & Bulygin E. (1971). *Normative Systems*. Springer-Verlag, New York.

ALDUS (1992). The ALDUS project: Artificial Legal Draftsman for Use in Sales, ESPRIT Commission.

Allen L. E. (1957). Symbolic Logic: A Razor-Edged Tool for Drafting and Interpreting Legal Documents. *The Yale Law Journal,* 66, pp. 833–879.

Allen L. E. (1968). A Language-Normalization Approach to Information Retrieval in Law. *Jurimetrics Journal,* September, pp. 41–57.

Allen L. E. (1978). Normalised Legal Drafting and the Query Method. *Journal of Legal Education*, 29, pp. 380–412.

Allen L. E. (1980). Language, Law and Logic: Plain Legal Drafting for the Electronic Age. In Niblett (ed.) *Computer Science and Law,* Cambridge University Press, Cambridge.

Allen, L. E. (1982). Towards a Normalised Language to Clarify the Structure of Legal Discourse. In Martino (ed.) *Deontic Logic, Computational Linguistics and Legal Information Systems,* North-Holland.

Allen, L. E. & C. S. Saxon (1984). Computer Aided Normalizing and Unpacking: Some Interesting Machine-Processable Transformation of Legal Rules. In Walter (Ed.) *Computer Power and Legal Reasoning*, West Publishing Company, 495–572.

Allen L. E. & Saxon C. E. (1993). A-Hohfeld: A Language for Robust Structural Representation of Knowledge in the Legal Domain to Build Interpretation–Assistance Expert Systems. In Meyer & Wieringa (Eds.) *Deontic Logic in Computer Science: Normative System Specification.* John Wiley & Sons, Chichester.

Åqvist L. & Hoepelman J. (1981). Some Theorems about a “Tree” System of Deontic Tense Logic. In Hilpinen (ed.) *New Studies in Deontic Logic*, D. Reidel, Dordrecht.

Atiyah P. S. (1979). *The Rise and Fall of Freedom of Contract*. Clarendon Press, Oxford.

Atiyah P. S. (1989). *An Introduction to the Law of Contract*. Clarendon Law Series, 4th edition, Clarendon Press, Oxford.

Austin J. L. (1962). *How to Do Things with Words*. (1975 edition), Clarendon Press, Oxford.

Bench-Capon T. J. M. (1987). Support for Policy Makers: Formulating Legislation with the Aid of Logical Models. In *Proceedings of the First International Conference on Artificial Intelligence and Law*, Boston Massachusetts, ACM Press, 181–189.

Bench-Capon T. J. M. & Dunne P. E. (1989). Some computational properties of a model for electronic documents. *Electronic Publishing,* 2:4, pp. 231–256.

Bench-Capon T. J. M. & Sergot M. J. (1988). Towards a rule-based representation of open texture in law. In Walter (ed.) *Computer Power and Legal Language,* Quorum Books, New York.

Bons R, Lee R. M., Wagenaar R. & Wrigley C. D. (1995). Modelling Inter-organizational Trade Procedures Using Documentary Petri Nets. In *Proceedings of the Hawaii International Conference on Systems Sciences (HICCS-95)*, Hawaii, IEEE Computer Society Press.

Branting L. K. (1998). Techniques for Automated Drafting of Judicial Documents. *International Journal of Law and Information Technology*, 6 (2), pp. 214-229.

Branting L. K. & Lester J. C. (1996). Justification Structures for Document Reuse. In *Proceedings of the Third European Workshop on Case-Based Reasoning (EWCBR-96)*, Lausanne, Switzerland, *Lecture Notes in Artificial Intelligence 1168*, pp. 76-90.

Branting L. K., Lester J. C. & Callaway C. B. (1997). Automated Drafting of Self-Explaining Documents. In *Proceedings of the 6th International Conference on Artificial Intelligence and Law*, Melbourne, Australia, ACM Press, pp. 72-81.

Branting L. K., Lester J. C. & Callaway C. B. (1998). Automated Judicial Document Drafting: A Discourse-Based Approach. *Artificial Intelligence and Law*, 6 (2/4), pp. 111-149.

Bryan M. (1988*). SGML: An Author's Guide to the Standard Generalized Markup Language*. Addison-Wesley.

Buchanan B. G. & Headrick T. E. (1970). Some speculation about artificial intelligence and legal reasoning. *Stanford Law Review*, 23.

Burch J. R., Clarke E. M., McMillan K. L., Dill D. L. & Hwang L. J. (1992). Symbolic Model Checking: 10^{20} States and Beyond. *Information and Computation*, 98 (2), pp. 142–170.

Capper P. N. & Susskind R. E. (1988). *Latent Damage Law–The Expert System*. Butterworths, London.

Chellas B. F. (1980). *Modal Logic: An Introduction*. Cambridge University Press.

Collins (1994). *Collins English Dictionary*. HarperCollins Publishers, Glasgow.

Clarke E. M., Emerson E. A. & Sistla A. P. (1986). Automatic Verification of Finite State Concurrent Systems using Temporal Logic Specifications. *ACM Transactions on Programming Languages*, 8 (2), pp. 244–263.

Daskalopulu A. & Sergot M. J. (1995). A Constraint-Driven System for Contract Assembly. In *Proceedings of the 5th International Conference on Artificial Intelligence and Law*, University of Maryland, College Park, ACM Press, pp. 62-70.

Daskalopulu A. & Sergot M. J. (1997). The Representation of Legal Contracts. *AI & Society*, 11 (1& 2), pp. 6–17.

Daskalopulu A. & Reed C. A. (1998). Modelling the Structure and Functionality of Contractual Text. In *Law in the Information Society, Proceedings of the Fifth International Conference of the Instituto per la Documentazione Giuridica of the Italian National Research Council*, (CD-Rom), IDG-CNR, December 2-5, Florence, Italy.

van Eck J. A. (1982). A System of Temporally Relative Modal and Deontic Predicate Logic and its Philosophical Applications. *Logique et Analyse*, 100. 249-381.

Evans D. B. (1990). Artificial Intelligence and Document Assembly. *Law Practice Management*, May/June, pp. 18–22.

Fiedler H. (1985). Expert Systems as a Tool for Drafting Legal Decisions. In Martino (ed.) *Pre-proceedings of the 2nd International Conference on Logic, Informatics and Law*, Florence, pp 265–274.

Fox J. & Thomson R. (1998). Decision Support and Disease Management: A Logic Engineering Approach. *IEEE Transactions on Information Technology in Biomedicine*, 2 (4).

Furuta R. (1989). Concepts and models for structured documents. In André, Furuta & Quint (eds.) *Structured Documents*, Cambridge University Press.

Gardner A. v. d. L. (1987). *An Artificial Intelligence Approach to Legal Reasoning.* MIT Press.

Geach P. & Black M. (1977) (eds.). *Translations from the Philosophical Writings of Gottlob Frege*, Basil Blackwell, Oxford.

Genrich H. J. & Lautenbach K. (1981). System Modelling with High Level Petri Nets. *Theoretical Computer Science*, 13, pp. 109–136.

Gordon T. F. (1992). A Theory Construction Approach to Legal Document Assembly. In Martino (ed.) *Expert Systems in Law*, Elsevier Publishers.

Greenleaf G., Mowbray A. & Tyree A. L. (1987). Expert Systems in Law: The Datalex Project. In *Proceedings of the first International Conference on Artificial Intelligence and Law*, Boston, ACM Press, pp. 9–17.

Hafner C. D. (1987). Conceptual Organization of Case Law Knowledge Bases. In *Proceedings of the first International Conference on Artificial Intelligence and Law*, Boston, ACM Press, pp. 35–42.

Hammond P. (1983). Representation of DHSS regulations as a Logic Program. In *Proceedings BCS Expert Systems '83*, Cambridge, British Computer Society.

Hammond P., Davenport J. C. & Fitzpatrick F. J. (1993). Logic-Based Integrity Constraints and the Design of Dental Prostheses. *Artificial Intelligence in Medicine*, 5, pp. 431–446.

Hammond P., Wyatt J. C. & Harris A. L. (1994). Drafting Protocols Certifying Clinical Trial Design and Monitoring Compliance. In *Proceedings of the European Conference on Artificial Intelligence (ECAI-94)*, Workshop on Normative Reasoning, Amsterdam, pp. 124–131.

Hellawell R. (1980). A Computer Program for Legal Planning and Analysis: Taxation of stock redemptions. *Columbia Law Review,* 80:70, pp. 1363–1398.

Hellawell R. (1981). CHOOSE: A computer program for legal planning and analysis. *Columbia Journal of Transnational Law*, 19, pp. 339–357.

Hellawell R. (1982). SEARCH: A computer program for legal problem solving. *Akron Law Review*, 15, pp. 635–653.

Hilpinen R. (1993). Actions in Deontic Logic. In Meyer & Wieringa (eds.) *Deontic Logic in Computer Science: Normative Systems Specification*, John Wiley & Sons, Chichester.

Hohfeld W. N. (1913). Some Fundamental Legal Conceptions as Applied in Judicial Reasoning. *Yale Law Journal*. 23.

Holzmann G. J. (1997). The Model Checker Spin. *IEEE Transactions in Software Engineering*, 23 (5), pp. 279–295.

Holzmann G. J. (1998). Seminar on Model Cheking at Royal Holloway and Bedford New College.

Hovy E. H. (1993). Automated Discourse Generation Using Discourse Structure Relations. *Artificial Intelligence*, 63, pp. 341–385.

IEE (1988). Model Form of General Conditions of Contract—Home or Overseas Contracts with Erection (MF/1). The Institution of Electrical Engineers, London.

IEE (1991). Model Form of General Conditions of Contract—Home or Overseas Contracts for the Supply of Electrical or Mechanical Plant (MF/2). The Institution of Electrical Engineers, London.

Jones A. J. I. (1990). Deontic Knowledge and Legal Knowledge Representation. *Ratio Juris*. 3:2. 237-244.

Jones A. J. I. & Pörn I. (1985). Ideality, Sub-Ideality and Deontic Logic. *Synthese*, 65, pp. 275–290.

Jones A. J. I. & Sergot M. J. (1992). Deontic Logic in the Representation of Law: Towards a Methodology. *Artificial Intelligence and Law*. 1:1. 45-64.

Jones A. J. I. & Sergot M. J. (1993). On the Characterisation of Law and Computer Systems: The Normative Systems Perspective. In Meyer & Wieringa (eds.) *Deontic Logic in Computer Science: Normative Systems Specification*, John Wiley & Sons, Chichester.

Jones A. J. I. & Sergot M. J. (1996). A Formal Characterisation of Institutionalized Power. *Journal of the Interest Group in Pure and Applied Logics*, 4:3, pp. 429–445. Also in: Valdés, Krawietz, von Wright & Zimmerling (eds.), *Normative Systems in Legal and Moral Theory*, Duncker & Humblot, Berlin, 1997.

Kanger S. & Kanger H. (1966). Rights and Parliamentarism. *Theoria*, 32, pp. 85–115.

Kanger S. (1971). New Foundations for Ethical Theory. In Hilpinen (ed.) *Deontic Logic: Introductory and Systematic Readings*, Dordrecht, pp. 36–58.

Kanger S. (1985). On Realization of Human Rights. In Holmstrom & Jones (eds.) *Action, Logic and Social Theory*. Acta Philosophica Fennica, 38.

Kimbrough S. O. (1997). On Electronic Commerce, Subatomic Semantics and the Cæsar's Stabbing. In Sprague (ed.) *Proceedings of the 30th Hawaii International Conference on Systems Sciences*, Los Alamitos CA, IEEE Computer Society Press.

Kimbrough S. O. (1998). Formal Language for Business Communication: Sketch of a Basic Theory". *International Journal of Electronic Commerce*, 3 (2), pp. 23–44.

Kimbrough S. O. (1999). Towards a Methodology for Analysis and Design of Messages for Communicating Agents. *Proceedings of the 12th International Bled Electronic Commerce Conference*, June 7–9, Bled, Slovenia.

Kimbrough S. O. & Lee R. M. (1986). On Illocutionary Logic as a Telecommunications Language. *Proceedings of the 7th International Conference on Information Systems*, San Diego, December 15–17, pp. 15–26.

Kimbrough S. O. & Lee R. M. (1997). Formal Aspects of Electronic Commerce: Research Issues and Challenges. *International Journal of Electronic Commerce*, 1 (4), pp. 11–30.

Kimbrough S. O. & Moore S. A. (1997). On Automated Message Processing in Electronic Commerce and Work Support Systems: Speech Act Theory and Expressive Felicity. *Transactions on Information Systems*, 15(4), pp. 321–367.

Koo R. (1989). A model for electronic documents. *ACM SIGOIS* Bulletin, 10 (1), pp. 23–33.

Kowalski, R. A. (1995). Legislation as Logic Programs. In Bankowski *et al.* (Eds.) *Informatics and the Foundations of Legal Reasoning*, Kluwer Academic Publishers, the Netherlands.

Kowalski R. A. & Sergot M. J. (1986). A logic-based calculus of events. *New Generation Computing*, 4, pp. 67–95.

Kowalski R. A. & Sergot M. J. (1990). The Use of Logical Models in Legal Problem Solving. *Ratio Juris*, 3(2), pp. 201–218.

Lauritsen M. (1992). Technology Report: Building Legal Practice Systems with Today's Commercial Authoring Tools. In *Artificial Intelligence and Law*, 1:1, pp. 87–102.

Lee R. M. & Dewitz S. (1992). Facilitating International Contracting: AI extensions to EDI. *International Information Systems*, January 1992.

Lee R. M. (1999). Automated Generation of Electronic Procedures: Procedure Constraint Grammars. *Workshop on Formal Models of Electronic Commerce*, June 2–3, ERASMUS University Research Centre for Electronic Commerce, Rotterdam, The Netherlands.

Lindahl, L. (1977). *Position and Change—A Study in Law and Logic*. Synthese Library, 112, D. Reidel, Dordrecht.

LPA (1992). *MacProlog 4.1 Reference Manual, and MacProlog 4.5 Update Notes*, Logic Programming Associates Ltd.

Loemker L. (1969). (ed.) *Philosophical Papers and Letters of G. W. Leibniz*, D. Reidel, Dordrecht.

McMillan K. L. (1993). *Symbolic Model Checking*. Kluwer.

Makinson D. (1986). On the Formal Representation of Rights Relations: Remarks on the Work of Stig Kanger and Lars Lindahl. *Journal of Philosophical Logic*, 15, pp. 403–425.

Mally E. (1926). *Grundgesetze des Sollens; Elemente der Logik des Willens*. Leuschner & Lubensky, Graz.

Mann W. C. & Thompson S. A. (1987). Rhetorical Structure Theory: A Theory of Text Organization. In Polanyi (ed.) *The Structure of Discourse*, Ablex Publishing Corporation, Norwood, N. J.

Mann W. C. & Thompson S. A. (1988). Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text*, 8, pp. 243–281.

McCarthy J. (1968). Programs with Common Sense. In Minsky (ed.) *Semantic Information Processing*, MIT Press, pp. 403–418.

McCarthy J. & Hayes P. J. (1969). Some Philosophical Problems from the Standpoint of Artificial Intelligence. In Meltzer & Michie (eds.) *Machine Intelligence* 4, Edinburgh University Press, pp. 463–502.

McCarty L. T. (1980). The TAXMAN Project: Towards a Cognitive Theory of Legal Argument. In Niblett (ed.) *Computer Science and Law*, Cambridge University Press, New York.

McCarty L. T. (1983). Intelligent Legal Information Systems: Problems and Prospects. *Rutgers Computers and Technology Law Journal*, 9:2, pp. 265–294. Also in Campbell C. (ed.) *Data Processing and the Law*, Sweet and Maxwell, London, pp. 125–151.

McCarty L. T. (1989). A Language for Legal Discourse I. Basic features. In *Proceedings of the Second International Conference on Artificial Intelligence and Law*, Vancoover, ACM Press, pp. 180–189.

McCarty L. T., Sridharan N. S. & Sangster B. C. (1979). The implementation of TAXMAN II: An experiment in artificial intelligence and legal reasoning. Technical Report LRP-

TR-2, Laboratory for Computer Science Research, Rutgers University, New Brunswick, New Jersey.

McCarty L. T. & Sridharan N. S. (1980). The representation conceptual structures in TAXMAN II. Part One: Logical Templates. Technical Report LRP-TR-4, Laboratory for Computer Science Research, Rutgers University, New Brunswick, New Jersey.

McCarty L. T. & Sridharan N. S. (1981). The representation of an evolving system of legal concepts. II. Prototypes and deformations. Technical Report LRP-TR-11, Laboratory for Computer Science Research, Rutgers University, New Brunswick, New Jersey.

McCarty L. T. & Sridharan N. S. (1982). A computational theory of legal argument. Technical Report LRP-TR-13, Laboratory for Computer Science Research, Rutgers University, New Brunswick, New Jersey.

McKeown K. (1985). *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press.

Mehl L. (1958). Automation in the Legal World. In *Proceedings of a Symposium on 'Mechanisation of Thought Processes'*, National Physical Laboratory, Teddington, pp. 755–779.

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

Meyer J.-J. Ch. & Wieringa R. J. (1993). Deontic Logic: A Concise Overview. In Meyer & Wieringa (eds.) *Deontic Logic in Computer Science: Normative Systems Specification*, John Wiley & Sons.

Michaelsen R. H. (1984). An Expert System for Federal Tax Planning. *Expert Systems*, 1:2.

Niblett B (1980) (ed.). *Computer Science and Law*. Cambridge University Press.

Parsons T. (1990). *Events in the Semantics of English: A Study in Subatomic Semantics*. Current Studies in Linguistics, MIT Press, Cambridge MA.

Peterson J. L. (1981). *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, Englewood Cliffs N. J.

Pörn I (1977). *Action Theory and Social Science: Some Formal Models*. Synthese Library120. D. Reidel, Dordrecht.

Prakken H. (1997). *Logical Tools for Modelling Legal Argument: A Study in Defeasible Reasoning in Law*. Kluwer, Dordrecht.

Prakken H. & Sergot M. J. (1996). Contrary-to-duty Obligations. *Studia Logica*. 57 (1/2). 91-115.

Prakken H. & Sergot M. J. (1997). Dyadic Deontic Logic and Contrary-to-Duty Obligations. In Nute (ed.) *Defeasible Deontic Logic*, Kluwer, The Netherlands.

Quine W. V. (1960). *Word and Object*, MIT Press, Cambridge MA.

Reed C. A. (1997). Representing and Applying Knowledge for Argumentation in a Social Context. *AI & Society*, 11 (1 & 2), pp. 138–154.

Reed C. A. (1998). *Generating Arguments in Natural Language*. PhD Thesis, University College London.

Rissland E. L. & Ashley K. D. (1987). A Case-Based System for Trade Secrets Law. In *Proceedings of the First International Conference on Artificial Intelligence and Law*, Boston, ACM Press, pp. 60–66.

Rissland E. L., Valcarce E, M. & Ashley K. D. (1984). Explaining and Arguing with Examples. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-84).*

Sanders K. E. (1991). Representing and reasoning about open-textured predicates. In *Proceedings of the Third International Conference on Artificial Intelligence and Law,* Oxford, ACM Press.

Santos F. & Carmo J. (1993). A Deontic Logic Representation of Contractual Obligations. In Meyer & Wieringa (eds.) *Deontic Logic in Computer Science: Normative System Specification*. John Wiley & Sons, Chichester.

Santos F. & Carmo J. (1996). Indirect Action, Influence and Responsibility. In Brown & Carmo (eds.) *Deontic Logic, Agency and Normative Systems*. Proceedings of the Third International Workshop on Deontic Logic in Computer Science (DEON-96), Springer-Verlag, Berlin-Heidelberg, pp. 194–215.

Sartor G. (1998). Logic and Argumentation in Legal Reasoning. *Current Legal Theory*, pp. 25–63.

Searle J. R. (1969). *Speech Acts.* Cambridge University Press, Cambridge.

Searle J. R. & Vanderveken D. (1985). *Foundations of Illocutionary Logic*. Cambridge University Press, Cambridge England.

Sergot M. J. (1985). Logic Programming and its Applications in Law: A brief tutorial introduction. In Walter (ed.) *Computer Power and Legal Language*, Greenwood/Quorum Press, New York.

Sergot M. J. (1988). Representing Legislation as Logic Programs. In Hayes, Michie, Richards (Eds.) *Machine Intelligence*, 11, Clarendon Press, Oxford.

Sergot M. J. (1991). The Representation of Law in Computer Programs. In Bench-Capon T. J. M. (ed.) *Knowledge-Based Systems and Legal Applications*, Academic Press.

Sergot M. J. (1997). A Computational Theory of Normative Positions. Technical Report, Department of Computing, Imperial College London. Submitted for publication.

Sergot M. J. (1998a). A Method for Automating the Analysis of Normative Positions. In *Proceedings of the Fourth International Workshop on Deontic Logic in Computer Science*, Bologna, Italy (January 1998).

Sergot M. J. (1998b). Normative Positions. In McNamara & Prakken (Eds.) *Norms, Logics and Information Systems*, IOS Press, pp. 289–308.

Sergot M. J., Sadri F., Kowalski R. A., Kriwaczek F., Hammond P.& Cory H. T. (1986). The British Nationality Act as a Logic Program. *Communications of the ACM.* 29 (5), pp. 370–386.

Sergot, M. J., Kamble, A. S. & Bajaj, K. K. (1991). Indian Central Civil Service Pension Rules: A case study in Logic Programming Applied to Regulations. In *Proceedings of the Third International Conference on Artificial Intelligence and Law*, Oxford, ACM Press, 118–127.

Smith J. M. & Stutely R. (1988). *SGML: the user's guide to ISO 8879*, Ellis Horwood Ltd.

Sprowl J. (1980). Automated Assembly of Legal Documents. In Niblett (ed.) *Computer Science and Law*, Cambridge University Press.

Stamper R. (1980). LEGOL: Modelling Legal Rules by Computer. In Niblett (ed.) *Computer Science and Law*, Cambridge University Press, New York.

Stamper R., Tagg C., Mason P., Cook S. & Marks J. (1982). Developing the LEGOL Semantic Grammar. In Martino & Socci (eds.) *Automated Analysis of Legal Texts*, North-Holland, Amsterdam.

Stone R. (1994). *Contract Law*. Cavendish Publishing Ltd, London.

Susskind R. E. (1990). Review of Gardner (1987) *An Artificial Intelligence Approach to Legal Reasoning.* In *Yearbook of Law Computers and Technology*, 4, Butterworths, London, pp. 221–228

Susskind R. E. (1998). *The Future of Law–Facing the Challenges of Information Technology*, Clarendon Press, Oxford.

Walton D. N. & Krabbe E. C. W. (1995). *Commitment in Dialogue*. State University of New York Press.

Waterman D. A. & Peterson M. A. (1980). Rule-based Models of Legal Expertise. *Proceedings of the National Conference on Artificial Intelligence (AAAI-80)*, pp. 272–275.

Wittgenstein L. (1961). *Tractatus Logico-Philosophicus*, Kegan Paul, London.

Wright G. H. von (1951). Deontic Logic. *Mind*, 60, pp. 1–15.

Wrigley C. D. (1992). EDI Transaction Protocols in International Trade. In *Proceedings of the Conference on Interorganizational Systems in the Global Environment*, Bled, Slovenia.

Yoshino H. (1997). “Legal Expert” Project. *Journal of Advanced Computational Intelligence*, 1:2, pp. 83–85.

Yoshino H. (1998). Logical Structure of Contract Law System for Constructing a Knowledge Base of the United Nations Convention on Contracts for the International Sale of Goods. *Journal of Advanced Computational Intelligence*, 2:1, pp. 2–11.

Zimmer R. M., Daskalopulu A. & Hunter A. (1999). Verifying Inter-organisational Trade Procedures by Model Checking Synchronised Petri Nets. In *Proceedings of the Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT-99)*, November 17–19, Mount Amanzi Lodge, Hartebeespoort, South Africa.

Zimmer R. M. & McDonald A. (1993). The Algebra of System Design: A Petri Net Model of Modular Composition. *IEEE International Symposium on Circuits and Systems*.

Zimmer R. M., McDonald A. & Holte R. (1992). Representation Engineering and Electronic Design: CAD as Automated Representation Change. PATRICIA European Community ESPRIT Project Deliverable.

A

REPRESENTATION OF GENERIC DOCUMENT IN MODELLER

This is an example of generic document representation in MODELLER. A simple tool was developed that facilitates entering such clauses. The tool enables the user to define the structure of a new generic document graphically, by setting up the nodes that correspond to document components and by attaching the required information to them (for example, what the title of a part is, whether it is compulsory or optional and so on). The user specifies constraints between document sub-components by linking them with separate types of line (one type of line is defined for each of the constraints we have discussed).

The representation of (IEE 1991)

```
generic(‘IEE/MF2’).
document_parameters(‘IEE/MF2’,[$Engineer, $Plant, $Contract_Date,
$Contractor, $Purchaser, $Contract_Price]).
%-------------------------
%Document Parts
%part(DocType, PartHeading, OptionalOrCompulsoryFlag
%-------------------------
part(‘IEE/MF2’, ‘Definitions and Interpretations’, c).
part(‘IEE/MF2’, ‘Engineer and Engineer Representative’, c).
part(‘IEE/MF2’, ‘Assignment and Subcontracting’, o).
part(‘IEE/MF2’, ‘Precedence of Documents’, o).
part(‘IEE/MF2’, ‘Basis of Tender and Contract Price’, c).
part(‘IEE/MF2’, ‘Changes in Costs’, o).
part(‘IEE/MF2’, ‘Purchaser General Obligations’, c).
part(‘IEE/MF2’, ‘Contractor Obligations, c).
part('IEE/MF2', 'Variations', o).
part('IEE/MF2', 'Defects Liability', o).
part('IEE/MF2', 'Tests on Completion', c).
part('IEE/MF2', 'Certificates and Payment', c).
part('IEE/MF2', 'Accidents and Damage', o).
part('IEE/MF2', 'Insurance', o).
part('IEE/MF2', 'Disputes and Arbitration', o).
part('IEE/MF2', 'Contract Documents', c).
part('IEE/MF2', 'Performance Bond or Guarantee', o).
part('IEE/MF2', 'Details Confidential', o).
part('IEE/MF2', 'Notices', o).
part('IEE/MF2', 'Inspection and Testing of Plant before Delivery', o).
part('IEE/MF2', 'Time for Delivery', c).
part('IEE/MF2', 'Delivery', c).
part('IEE/MF2', 'Suspension of Contract', c).
part('IEE/MF2', 'Delay', o).
part('IEE/MF2', 'Vesting of Plant', o).
part('IEE/MF2', 'Claims', o).
part('IEE/MF2', 'Patent Rights', c).
part('IEE/MF2', 'Limitations of Liability', o).
part('IEE/MF2', 'Contractor Default', c).
part('IEE/MF2', 'Purchaser Default', c).
%-------------------------
%Part Components
%section(DocType, PartHeading, SectionHeading, NumberInPart, ListofParameters
%-------------------------
section('IEE/MF2', 'Definitions and Interpretations', 'Purchaser', 1-a,
[$Purchaser]).
section('IEE/MF2', 'Definitions and Interpretations', 'Contractor', 1-b,
[$Contractor]).
section('IEE/MF2', 'Definitions and Interpretations', 'SubContractor', 1-c,
[]).
section('IEE/MF2', 'Definitions and Interpretations', 'Engineer', 1-d,
[$Engineer]).
section('IEE/MF2', 'Definitions and Interpretations', 'Engineer
Representative', 1-e, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Conditions', 1-f, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Contract', 1-g, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Contract Price', 1-h,
[$Contract_Price]).
section('IEE/MF2', 'Definitions and Interpretations', 'Contract Value', 1-i,
[]).
section('IEE/MF2', 'Definitions and Interpretations', 'Cost', 1-j, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Tender', 1-k, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Letter of Acceptance',
1-l, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Delivery', 1-m, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Time for Delivery', 1-
n, [$Date_Letter_Acceptance, $Date_Advance_Payment, $Contract_Date]).
section('IEE/MF2', 'Definitions and Interpretations', 'Plant', 1-o, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Section of Plant', 1-
p, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Programme', 1-q, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Specification', 1-r,
[]).
section('IEE/MF2', 'Definitions and Interpretations', 'Special Conditions',
1-s, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Tests', 1-t, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Defects Liability
Period', 1-u, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Appendix', 1-v, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Writing', 1-w, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Day', 1-x, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Week', 1-y, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Month', 1-z, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Interpretation', 2,
[]).
section('IEE/MF2', 'Definitions and Interpretations', 'Singular and Plural',
3, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Notices and Consents',
4, []).
section('IEE/MF2', 'Definitions and Interpretations', 'Headings and Marginal
Notes', 5, []).
%-------------------------
section('IEE/MF2', 'Engineer and Engineer Representative', 'Engineer Duties',
1, []).
section('IEE/MF2', 'Engineer and Engineer Representative', 'Engineer
Representative', 2, []).
section('IEE/MF2', 'Engineer and Engineer Representative', 'Engineer Power to
Delegate', 3, []).
section('IEE/MF2', 'Engineer and Engineer Representative', 'Engineer
Decisions Instructions and Orders', 4, []).
section('IEE/MF2', 'Engineer and Engineer Representative', 'Confirmation in
Writing', 5, []).
section('IEE/MF2', 'Engineer and Engineer Representative', 'Disputing
Engineer Decisions Instructions and Orders', 6, [$First_Notice_Period,
$Second_Notice_Period]).
section('IEE/MF2', 'Engineer and Engineer Representative', 'Engineer to Act
Fairly', 7, []).
section('IEE/MF2', 'Engineer and Engineer Representative', 'Replacement of
Engineer', 8, []).
%-------------------------
section('IEE/MF2', 'Assignment and Subcontracting', 'Assignment', 1, []).
section('IEE/MF2', 'Assignment and Subcontracting', 'Sub-Contracting', 2,
[]).
%-------------------------
section('IEE/MF2', 'Precedence of Documents', 'Precedence of Documents', 1,
[]).
%-------------------------
```

```
section(‘IEE/MF2’, ‘Basis of Tender and Contract Price’, ‘Contractor to
inform himself fully’, 1, []).
section(‘IEE/MF2’, ‘Basis of Tender and Contract Price’, ‘Site Data’, 2, []).
section(‘IEE/MF2’, ‘Basis of Tender and Contract Price’, ‘Incorrect
Information ’, 3, []).
section(‘IEE/MF2’, ‘Basis of Tender and Contract Price’, ‘Provisional Sums’,
4, []).
section(‘IEE/MF2’, ‘Basis of Tender and Contract Price’, ‘Prime Cost Items’,
5, []).
section(‘IEE/MF2’, ‘Basis of Tender and Contract Price’, ‘Work Carried out by
Others’, 6, []).
%-------------------------
section(‘IEE/MF2’, ‘Changes in Costs’, ‘Statutory and Other Regulations’, 1,
[]).
section(‘IEE/MF2’, ‘Changes in Costs’, ‘Labour Materials and Transport’, 2,
[]).
%-------------------------
section(‘IEE/MF2’, ‘Contract Documents’, ‘Agreement’, 1, []).
%-------------------------
section(‘IEE/MF2’, ‘Performance Bond or Guarantee’, ‘Provision of Bond of
Guarantee’, 1, []).
section(‘IEE/MF2’, ‘Performance Bond or Guarantee’, ‘Failure to Provide Bond
of Guarantee’, 2, [$Notice_Period]).
%-------------------------
section(‘IEE/MF2’, ‘Details Confidential’, ‘Details Confidential’, 1, []).
%-------------------------
section(‘IEE/MF2’, ‘Notices’, ‘Notices to Purchaser and Engineer’, 1, []).
section(‘IEE/MF2’, ‘Notices’, ‘Notices to Contractor’, 2, []).
section(‘IEE/MF2’, ‘Notices’, ‘Service of Notices’, 3, []).
%-------------------------
section(‘IEE/MF2’, ‘Purchaser General Obligations’, ’Import Permits Licences
and Duties’, 1, []).
section(‘IEE/MF2’, ‘Purchaser General Obligations’, ‘Assistance with Laws and
Regulations’, 2, []).
%-------------------------
section(‘IEE/MF2’, ‘Contractor Obligations’, ’General’, 1, []).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Manner of Execution’, 2, []).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Contractor Design’, 3, []).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Packing’, 4, []).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Programme’, 5,
[$Programme_Date]).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Form of Programme’, 6, []).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Approval of Programme’, 7, []).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Alterations to Programme’, 8,
[]).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Revisions of Programme’, 9,
[]).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Rate of Progress’, 10, []).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Drawings’, 11,
[$Approval_Date]).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Consequences of Disapproval of
Drawings’, 12, []).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Approved Drawings’, 13, []).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Inspection of Drawings’, 14,
[]).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Foundation etc Drawings’, 15,
[]).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Operating and Maintenance
Instructions’, 16, []).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Purchaser Use of Drawings
Supplied by Contractor’, 17, []).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Contractor Use of Drawings
Supplied by Purchaser or Engineer’, 18, []).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Manufacturing Drawings’, 19,
[]).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Errors in Drawings Supplied by
Contractor’, 20, []).
section(‘IEE/MF2’, ‘Contractor Obligations’, ’Errors in Drawings Supplied by
Engineer’, 21, []).
%-------------------------
section(‘IEE/MF2’, ‘Inspection and Testing of Plant before Delivery’,
’Engineer Entitlement to Test’, 1, []).
section(‘IEE/MF2’, ‘Inspection and Testing of Plant before Delivery’, ’Date
for Tests of Inspection’, 2, [$Notice_Period]).
section(‘IEE/MF2’, ‘Inspection and Testing of Plant before Delivery’,
’Services for Tests or Inspection’, 3, []).
section(‘IEE/MF2’, ‘Inspection and Testing of Plant before Delivery’,
’Certificate of Tests or Inspection’, 4, []).
section(‘IEE/MF2’, ‘Inspection and Testing of Plant before Delivery’,
’Failure on Tests or Inspection’, 5, [$Rejection_Notice]).
%-------------------------
section(‘IEE/MF2’, ‘Delivery’, ’Permission to Deliver’, 1, []).
section(‘IEE/MF2’, ‘Delivery’, ’Delivery Terms’, 2, [$Regulatory_Terms]).
section(‘IEE/MF2’, ‘Delivery’, ’Notice to Insure’, 3, [$Sale_of_Goods_Act]).
%-------------------------
section(‘IEE/MF2’, ‘Suspension of Contract’, ‘Instructions to Suspend’, 1,
[]).
section(‘IEE/MF2’, ‘Suspension of Contract’, ’Additional Cost caused by
Suspension’, 2, []).
section(‘IEE/MF2’, ‘Suspension of Contract’, ’Payment for Plant Affected by
Suspension’, 3, []).
section(‘IEE/MF2’, ‘Suspension of Contract’, ’Disallowance of Additional Cost
or Payment’, 4, []).
section(‘IEE/MF2’, ‘Suspension of Contract’, ’Resumption of Manufacture or
Delivery’, 5, []).
section(‘IEE/MF2’, ‘Suspension of Contract’, ’Effects of Suspension on
Defects Liability’, 6, []).
```

```
section(‘IEE/MF2’, ‘Suspension of Contract’, ’Defects before Delivery’, 7,
[]).
%-------------------------
section(‘IEE/MF2’, ‘Variations’, ’Meaning of Variation’, 1, []).
section(‘IEE/MF2’, ‘Variations’, ‘Engineer Power to Vary’, 2,
[$Proportion_Contract_Price]).
section(‘IEE/MF2’, ‘Variations’, ’Valuation of Variations’, 3, []).
section(‘IEE/MF2’, ‘Variations’, ’Contractor Record of Costs’, 4, []).
section(‘IEE/MF2’, ‘Variations’, ’Notice and Confirmation of Variations’, 5,
[]).
section(‘IEE/MF2’, ‘Variations’, ’Progress with Variations’, 6,
[$Proportion_Contract_Price]).
%-------------------------
section(‘IEE/MF2’, ‘Time for Delivery’, ‘Time for Delivery’, 1, []).
section(‘IEE/MF2’, ‘Time for Delivery’, ‘Extension of Time for Delivery’, 2,
[]).
section(‘IEE/MF2’, ‘Time for Delivery’, ‘Delays by Subcontractors’, 3, []).
section(‘IEE/MF2’, ‘Time for Delivery’, ‘Mitigation of Consequences of
Delay’, 4, []).
%-------------------------
section(‘IEE/MF2’, ‘Delay’, ‘Delay in Delivery’, 1, []).
section(‘IEE/MF2’, ‘Delay’, ‘Prolonged Delay’, 2, [$Minimum_Extension]).
%-------------------------
section(‘IEE/MF2’, ‘Defects Liability’, ‘Defects after Delivery’, 1,
[$Liability_Period]).
section(‘IEE/MF2’, ‘Defects Liability’, ‘Making Good Defects’, 2, []).
section(‘IEE/MF2’, ‘Defects Liability’, ‘Notice of Defects’, 3,
[$Repair_Liability_Period]).
section(‘IEE/MF2’, ‘Defects Liability’, ‘Extension of Defects Liability’, 4,
[]).
section(‘IEE/MF2’, ‘Defects Liability’, ‘Delay in Remedying Defects’, 5, []).
section(‘IEE/MF2’, ‘Defects Liability’, ‘Further Tests’, 6, []).
section(‘IEE/MF2’, ‘Defects Liability’, ‘Limitation of Liability for
Defects’, 7, []).
section(‘IEE/MF2’, ‘Defects Liability’, ‘Latent Defects’, 8, []).
%-------------------------
section(‘IEE/MF2’, ‘Vesting of Plant’, ‘Ownership of Plant’, 1, []).
section(‘IEE/MF2’, ‘Vesting of Plant’, ‘Marking of Plant’, 2, []).
%-------------------------
section(‘IEE/MF2’, ‘Certificates and Payment’, ‘Application for Payment, 1,
[]).
section(‘IEE/MF2’, ‘Certificates and Payment’, ‘Form of Application’, 2, []).
section(‘IEE/MF2’, ‘Certificates and Payment’, ‘Issue of Payment
Certificate’, 3, [$Invoice_Date]).
section(‘IEE/MF2’, ‘Certificates and Payment’, ‘Value Included in
Certificates of Payment’, 4, []).
section(‘IEE/MF2’, ‘Certificates and Payment’, ‘Adjustment to Certificates’,
5, []).
```

```
section('IEE/MF2', 'Certificates and Payment', 'Corrections to Certificates',
6, []).
section('IEE/MF2', 'Certificates and Payment', 'Withholding Certificate of
Payment', 7, []).
section('IEE/MF2', 'Certificates and Payment', 'Effects of Certificate of
Payment', 8, []).
section('IEE/MF2', 'Certificates and Payment', 'Application for Final
Certificate of Payment', 9, []).
section('IEE/MF2', 'Certificates and Payment', 'Value of Final Certificate of
Payment', 10, []).
section('IEE/MF2', 'Certificates and Payment', 'Issue of Final Certificate of
Payment', 11, [$Date_Final_Certificate_Payment]).
section('IEE/MF2', 'Certificates and Payment', 'Effect of Final Certificate
of Payment', 12, []).
section('IEE/MF2', 'Certificates and Payment', 'No Effect in Case of Gross
Misconduct', 13, []).
section('IEE/MF2', 'Certificates and Payment', 'Payment', 14,
[$Payment_Date]).
section('IEE/MF2', 'Certificates and Payment', 'Delayed Payment', 15, []).
section('IEE/MF2', 'Certificates and Payment', 'Remedies on Failure to
Certify or Make Payment', 15, [$Notice_Suspension_Work,
$Notice_Termination_Contract]).
%-------------------------
section('IEE/MF2', 'Claims', 'Notification of Claims', 1, []).
section('IEE/MF2', 'Claims', 'Allowance for Profit on Claims', 2, []).
section('IEE/MF2', 'Claims', 'Purchaser Liability to Pay Claims', 3, []).
%-------------------------
section('IEE/MF2', 'Patents Rights', 'Indemnity Against Infringement', 1,
[]).
section('IEE/MF2', 'Patents Rights', 'Conduct of Proceedings', 2, []).
section('IEE/MF2', 'Patents Rights', 'Purchaser Indemnity Against
Infringement', 3, []).
section('IEE/MF2', 'Patents Rights', 'Infringement affecting Performance', 4,
[$Notice_Period]).
%-------------------------
section('IEE/MF2', 'Accidents and Damage', 'Care of Plant', 1, []).
section('IEE/MF2', 'Accidents and Damage', 'Making Good Loss or Damage to
Plant', 2, []).
%-------------------------
section('IEE/MF2', 'Limitations of Liability', 'Mitigation of Loss', 1, []).
section('IEE/MF2', 'Limitations of Liability', 'Indirect or Consequential
Damage', 2, []).
section('IEE/MF2', 'Limitations of Liability', 'Limitations of Contractor
Liability', 3, []).
section('IEE/MF2', 'Limitations of Liability', 'Exclusive Remedies', 4, []).
%-------------------------
section('IEE/MF2', 'Insurance', 'Insurance of Plant', 1, []).
section('IEE/MF2', 'Insurance', 'Application of Insurance Monies', 2, []).
section('IEE/MF2', 'Insurance', 'General Insurance Requirements', 3, []).
section('IEE/MF2', 'Insurance', 'Exclusion from Insurance Cover', 4, []).
section('IEE/MF2', 'Insurance', 'Remedy on Failure to Insure', 5, []).
%-------------------------
section('IEE/MF2', 'Contractor Default', 'Default', 1, [$Notice_Period]).
section('IEE/MF2', 'Contractor Default', 'Valuation at Date of Termination',
2, []).
section('IEE/MF2', 'Contractor Default', 'Payment after Termination', 3, []).
section('IEE/MF2', 'Contractor Default', 'Bankruptcy and Insolvency', 4, []).
%-------------------------
section('IEE/MF2', 'Purchaser Default', 'Notice of Termination due to
Purchaser Default', 1, [$Notice_Period]).
section('IEE/MF2', 'Purchaser Default', 'Payment on Termination due to
Purchaser Default', 2, []).
%-------------------------
section('IEE/MF2', 'Disputes and Arbitration', 'Notice of Arbitration', 1,
[$Notice Period]).
section('IEE/MF2', 'Disputes and Arbitration', 'Performance to Continue
during Arbitration', 2, []).
section('IEE/MF2', 'Disputes and Arbitration', 'Applicable Law', 3,
[$Applicable_Law]).

%-------------------------
%Constraints
%-------------------------
refers('IEE/MF2', sect('Definitions and Interpretations', 1-e),
sect('Engineer and Engineer Representative', _)).
refers('IEE/MF2', sect('Definitions and Interpretations', 1-
h),sect('Variations', _)).
refers('IEE/MF2', sect('Definitions and Interpretations', 1-n),sect('Time for
Delivery',2)).
refers('IEE/MF2', sect('Definitions and Interpretations', 1-
q),sect('Contractor Obligations', 5)).
refers('IEE/MF2', sect('Definitions and Interpretations', 1-
r),sect('Variations', _)).
refers('IEE/MF2', sect('Definitions and Interpretations', 1-u),sect('Defects
Liability', _)).
refers('IEE/MF2', sect('Engineer and Engineer Representative',
2),sect('Engineer and Engineer Representative', 3)).
refers('IEE/MF2', sect('Engineer and Engineer Representative',
3),sect('Engineer and Engineer Representative', 6)).
refers('IEE/MF2', sect('Engineer and Engineer Representative',
6),sect('Engineer and Engineer Representative', 5)).
refers('IEE/MF2', sect('Contractor Obligations, 10),sect('Time for Delivery',
2)).
refers('IEE/MF2', sect('Contractor Obligations', 16),sect('Delay', _)).
refers('IEE/MF2', sect('Contractor Obligations', 20),sect('Delay', _)).
refers('IEE/MF2', sect('Suspension of Contract, 2),sect('Suspension of
Contract', 1)).
```

```
refers('IEE/MF2', sect('Suspension of Contract', 3),sect('Certificates and
Payment', 3)).
refers('IEE/MF2', sect('Suspension of Contract', 3),sect('Vesting of Plant',
2)).
refers('IEE/MF2', sect('Suspension of Contract', 3),sect('Suspension of
Contract', 1)).
refers('IEE/MF2', sect('Suspension of Contract', 4),sect('Suspension of
Contract', 2)).
refers('IEE/MF2', sect('Suspension of Contract', 4),sect('Suspension of
Contract', 3)).
refers('IEE/MF2', sect('Suspension of Contract', 5),sect('Suspension of
Contract', 1)).
refers('IEE/MF2', sect('Suspension of Contract', 5),sect('Suspension of
Contract', 4)).
refers('IEE/MF2', sect('Suspension of Contract', 5),sect('Variations', _)).
refers('IEE/MF2', sect('Suspension of Contract', 5),sect('Purchaser
Default',1)).
refers('IEE/MF2', sect('Suspension of Contract', 5),sect('Purchaser
Default',2)).
refers('IEE/MF2', sect('Suspension of Contract', 6),sect('Defects Liability',
_)).
refers('IEE/MF2', sect('Suspension of Contract', 7),sect('Inspection and
Testing of Plant before Delivery', 5)).
refers('IEE/MF2', sect('Suspension of Contract', 7),sect('Delay', _)).
refers('IEE/MF2', sect('Variations', 4),sect('Variations', 3)).
refers('IEE/MF2', sect('Variations', 6),sect('Variations', 2)).
refers('IEE/MF2', sect('Variations', 6),sect('Variations', 5)).
refers('IEE/MF2', sect('Time for Delivery', 2),sect('Variations', _)).
refers('IEE/MF2', sect('Time for Delivery', 3),sect('Time for Delivery', 1)).
refers('IEE/MF2', sect('Time for Delivery', 4),sect('Time for Delivery', 1)).
refers('IEE/MF2', sect('Delay', 1),sect('Delay', 2)).
refers('IEE/MF2', sect('Delay', 2),sect('Delay', 1)).
refers('IEE/MF2', sect('Defects Liability', 2),sect('Contractor Obligations',
3)).
refers('IEE/MF2', sect('Defects Liability', 3),sect('Defects Liability', 2)).
refers('IEE/MF2', sect('Defects Liability', 7),sect('Defects Liability', 8)).
refers('IEE/MF2', sect('Defects Liability', 8),sect('Defects Liability', 2)).
refers('IEE/MF2', sect('Certificates and Payment', 2),sect('Claims', 1)).
refers('IEE/MF2', sect('Certificates and Payment', 2),sect('Suspension of
Contract', 3)).
refers('IEE/MF2', sect('Certificates and Payment', 9),sect('Defects
Liability', _)).
refers('IEE/MF2', sect('Certificates and Payment', 11),sect('Certificates and
Payment', 9)).
refers('IEE/MF2', sect('Certificates and Payment', 13),sect('Defects
Liability', 8)).
refers('IEE/MF2', sect('Certificates and Payment', 15),sect('Certificates and
Payment', 14)).
refers('IEE/MF2', sect('Claims', 2),sect('Purchaser General Obligations',
1)).
```

```
refers('IEE/MF2', sect('Claims', 2),sect('Contractor Obligations', 9)).
refers('IEE/MF2', sect('Claims', 2),sect('Contractor Obligations', 21)).
refers('IEE/MF2', sect('Claims', 2),sect('Suspension of Contract', 2)).
refers('IEE/MF2', sect('Claims', 2),sect('Suspension of Contract', 5)).
refers('IEE/MF2', sect('Claims', 2),sect('Suspension of Contract', 6)).
refers('IEE/MF2', sect('Claims', 2),sect('Time for Delivery', 3)).
refers('IEE/MF2', sect('Claims', 2),sect('Certificates and Payment', 16)).
refers('IEE/MF2', sect('Claims', 2),sect('Disputes and Arbitration', 2)).
refers('IEE/MF2', sect('Patent Rights', 3),sect('Patent Rights', 1)).
refers('IEE/MF2', sect('Patent Rights', 3),sect('Patent Rights', 2)).
refers('IEE/MF2', sect('Patent Rights', 4),sect('Patent Rights', 1)).
refers('IEE/MF2', sect('Patent Rights', 4),sect('Patent Rights', 3)).
refers('IEE/MF2', sect('Limitations of Liability', 2),sect('Delay', 1)).
refers('IEE/MF2', sect('Limitations of Liability', 2),sect('Claims', 2)).
refers('IEE/MF2', sect('Limitations of Liability', 3),sect('Defects
Liability', 8)).
refers('IEE/MF2', sect('Insurance, 1),sect('Vesting of Plant', 1)).
refers('IEE/MF2', sect('Contractor Default', 1),sect('Inspection and Testing
of Plant before Delivery', 5)).
refers('IEE/MF2', sect('Contractor Default', 2),sect('Certificates and
Payment', _)).
refers('IEE/MF2', sect('Purchaser Default', 1),sect('Certificates and
Payment', 16)).
refers('IEE/MF2', sect('Purchaser Default', 2),sect('Contractor Default',
2)).
refers('IEE/MF2', sect('Purchaser Default', 2),sect('Purchaser Default', 1)).
refers('IEE/MF2', sect('Purchaser Default', 2),sect('Claims', 2)).
refers('IEE/MF2', sect('Disputes and Arbitration', 1),sect('Engineer and
Engineer Representative', 6)).
forces('IEE/MF2', part('Assignment and Subcontracting'), part('Changes in
Costs')).
forces('IEE/MF2', part('Assignment and Subcontracting'), part('Variations')).
forces('IEE/MF2', part('Assignment and Subcontracting'), part('Defects
Liability')).
forces('IEE/MF2', part('Defects Liability'), part('Claims')).
forces('IEE/MF2', part('Defects Liability'), part('Performance Bond or
Guarantee')).
forces('IEE/MF2', part('Limitations of Liability'), part('Defects
Liability')).
forces('IEE/MF2', part('Defects Liability'), part('Disputes and
Arbitration')).
forces('IEE/MF2', part('Claims'), part('Disputes and Arbitration')).
forces('IEE/MF2', part('Insurance'), part('Disputes and Arbitration')).
forces('IEE/MF2', part('Delay'), part('Disputes and Arbitration')).
forces('IEE/MF2', part('Delay'), part('Claims')).
forces('IEE/MF2', part('Limitations of Liability'), part('Disputes and
Arbitration')).
forces('IEE/MF2', part('Vesting of Plant'), part('Disputes and
Arbitration')).
```

```
forces(‘IEE/MF2’, part(‘Accidents and Damage’), part(‘Disputes and
Arbitration’)).
forces(‘IEE/MF2’, part(‘Inspection and Testing of Plant before Delivery’),
part(‘Disputes and Arbitration’)).
forces(‘IEE/MF2’, data(‘Delivery’, [$Regulatory_Terms=’English Law’),
part(‘Disputes and Arbitration’, [$Applicable_Law=’English Law’)).
forces(‘IEE/MF2’, data(‘Delivery’, [$Regulatory_Terms=’English Law’),
part(‘Vesting of Plant’)).
```

B

SAMPLE DOCUMENTS FOR MODELLER

The following is a list of sample contracts that were used to construct generic documents for MODELLER. Sample documents supplied by British Gas are confidential.

1. Model Form of General Conditions of Contract—Home or Overseas Contracts with Erection (MF/1). The Institution of Electrical Engineers, London, 1988.
   Size (Pages): 36; Atomic Components (Number): 184; Constraints (Number): 137
2. Model Form of General Conditions of Contract—Home or Overseas Contracts for the Supply of Electrical or Mechanical Plant (MF/2). The Institution of Electrical Engineers, London, 1991. Size: 26 pages, Number of Constraints
   Size (Pages): 26; Atomic Components (Number): 159; Constraints (Number): 98
3. General Conditions of Contract for Gas Purchase. British Gas Plc, 1991.
   Size (Pages): 123; Atomic Components (Number): 464; Constraints (Number): 235
4. Conditions of Contract for Consultancy and other Professional Services. British Gas Plc, December 1992.
   Size (Pages): 18; Atomic Components (Number): 70; Constraints (Number): 21
5. Conditions of Contract for Research. British Gas Plc, June 1993.
   Size (Pages): 15; Atomic Components (Number): 62; Constraints (Number): 15
6. Commission of the European Communities Model Contracts for Community Activities in the Field of Research and Technological Development. Directorate General XII, October 1988.

—Actual Cost Contract

Size (Pages): 6; Atomic Components (Number): 30; Constraints (Number): 18

—Fixed Contribution Contract

Size (Pages): 6; Atomic Components (Number): 28; Constraints (Number): 15

—General Conditions

Size (Pages): 12; Atomic Components (Number): 96; Constraints (Number): 74

The following documents were consulted to determine whether the modeller approach would be applicable, without however constructing actual representations:

1. Assured Shorthold Tenancy Agreement.
2. Confidentiality Agreement. British Gas Plc, October 1990.
3. Novation Agreements (seller to seller, contractor to new contractor). British Gas Plc, October 1990.
4. Performance Bond. British Gas Plc, October 1990.
5. Bank Retention Bond. British Gas Plc, October 1990.
6. Law Temperature Recuperative Burner Development. British Gas Plc, October 1987.
7. General Conditions for Consultancy Services Onshore. British Gas Plc, July 1989.

C

NOTES ON PERFORMATIVE STATEMENTS

C.1 Classification of Performative Statements

Searle (1969) identified five distinct classes in which performative statements fall. These are summarised by Kimbrough & Moore (1997) as follows:

(i) Assertives—used to describe states of affairs in the world, used to make statements.

(ii) Commissives—used to commit the speaker to an action; used to make promises.

(iii) Directives—used to commit the hearer to an action; used to give orders.

(iv) Declaratives—used to make changes in virtue of speaking; in this case, “saying so makes it so”.

(v) Expressives—used to express the speaker’s attitude; as in “Oh, to be in England”.

Austin too, distinguished five general classes, albeit he noted that he was “far from equally happy about all of them” (Austin 1962 p. 151):

Performative Class	**Explication**	**Performative verbs (Examples)**
Verdictive	Giving a verdict by a jury arbitrator or umpire, or giving an estimate, reckoning or appraisal. An exercise of speaker's judgement. Committing hearer/others to certain future conduct.	Acquit, convict, grade, assess, locate, measure, find (as a matter of fact), rule, diagnose etc.
Exercitives	Assertion of influence or exercising of powers or rights. Giving a decision that something is to be so. Creating obligations, permissions or prohibitions for hearer/others.	Appoint, vote, order, urge, advise, warn, dismiss, demote, name, bequeath, proclaim, resign, nominate, recommend etc.
Commissives	Committing speaker to a course of action. Assuming of an obligation or declaring of intention.	Promise, undertake, intend, plan, shall, adopt, oppose, guarantee, consent etc.
Behabitives	Describing speaker's reaction to other people's behaviour or states of affairs. Adopting an attitude.	Apologize, thank, commiserate, resent, welcome, protest, challenge etc.
Expositives	Describing views, clarifying reasons, arguments and communications.	Affirm, deny, state, identify, inform, postulate, interpret, agree etc.

Table C-1 Austin's (1962) classification of performative utterances

As Austin points out a verb in itself is not sufficient to establish the class of its associated performative utterance. The verb "agree" may be taken as expositive (as in "I agree with your view on this matter") but also as commissive (as in "I agree to deliver the parcel to your doorstep") depending on the context in which the utterance takes place. Similarly, the verb "recommend" may be behabitive (as in "I recommend Rioja 1986 to you") or exercitive or verdictive (as in "The Board of Examiners recommends this student for the award of an Honours degree"). The classification of any particular illocutionary force will therefore (not surprisingly) require interpretation.

The distinction between classes within each classification seems to be somewhat blurred. For instance, in Searle's classification the distinction between declaratives and directives is not altogether clear-cut. "You are out" (as said by an umpire to a player) can be construed as a declarative (the umpire's saying that the player is out makes it so that the player is out, by virtue of the umpire's power) but it can also be construed as an order (the umpire orders the player to leave the game). Perhaps the underlying distinction between declarative and directive lies in the deontic realm. Issuing an order may be construed as issuing an obligation for the player to leave the game, which he may satisfy or violate; if the player violates it, then he is technically still in the game, albeit undesirably so. In its declarative guise the statement may be construed as asserting that the player is considered out of the game whether he actually leaves or not, that is, any subsequent actions of the player are not considered relevant to the game.

The distinction between classes in Austin's classification is also blurred. "The Board of Examiners recommends this student for the award of an Honours degree" can be construed as exercitive (creating an obligation for the Registrar of the University to issue the recommended degree, creating a right for the student to obtain the recommended degree), as we saw earlier, or verdictive. The explication of verdictives by Austin himself refers to committing the hearer or others to future conduct, as does the explication of exercitives. Similarly, "I deny the charges brought against me" can be construed as expositive but also as behabitive (the speaker is protesting against the charges brought against him).

C.2 Searle on the Structure of Illocutionary Acts

In his (1969) Searle explores the structure of illocutionary acts. In his exposition he concentrates on promises, and provides the following necessary and sufficient conditions for a sincere and non-defective[56] promise:

Consider a speaker *S* who utters a sentence *T* in the presence of a hearer *H*. Then in the literal utterance of *T*, *S* sincerely and non-defectively promises that *p* to *H* if and only if the following conditions hold:

1. Normal input and output conditions obtain. As Searle notes "input" refers to conditions of understanding and "output" refers to conditions for intelligible speaking. The condition generally means that speaker and hearer both know how to speak the language, are conscious of what they are doing, suffer from no physical impediments to communication, engage in communication literally (i.e., they are not acting or exchanging jokes) and so on.

2. *S* expresses the proposition that *p* in the utterance of *T*. Searle here concentrates on explicit promises and leaves aside issues of ellipsis, hints and so on.

3. In expressing that *p, S* predicates a future act *A* of *S.* This is to capture the intuition that one cannot make promises about the past. As Searle notes, *A*, the promised act, includes

[56] The notion of an utterance being non-defective is closely linked to Austin's (1962) notion of felicity, which we saw briefly in section 2.4.2

refraining from acts, performing series of acts, and may include states and conditions (as in promising to do something, promising not to do something, promising to do something repeatedly or sequentially, and promising to remain in a certain state or condition).

4. *H* would prefer *S*'s doing *A* to his not doing *A*, and *S* believes *H* would prefer his doing *A* to his not doing *A*. The first conjunct distinguishes between promises and threats. A promise is defective if what is promised is not what the promisee wants. The second conjunct distinguishes between promises and invitations (perhaps it is more accurate to say "volunteering" rather than invitation). A promise, as Searle notes, requires some situation or occasion that calls for it, unlike an invitation. At this point Searle alludes to the notion of beneficiary of a promise:

   > "In general, the point stated in condition 4 is that if the purported promise is to be non-defective, the thing promised must be something the hearer wants done, or considers to be in his interest, or would prefer being done to not being done, etc.; and the speaker must be aware of or believe or know, etc., that this is the case. I think a more elegant and exact formulation of this condition would probably require the introduction of technical terminology of the welfare economics sort". (Searle 1969, p. 59).

5. It is not obvious to both *S* and *H* that *S* will do *A* in the normal course of events. This expresses the intuition that the illocutionary act (a promise in this case) must have a point. As promises incur obligations, this is in line with an intuition about non-defective agreements: There is no valid contract between me and a policeman that promises me to patrol my street in exchange for some reward, if in the normal course of events, the policeman would be patrolling my street anyway, while performing the duties associated with his function[57].

6. *S* intends to do *A*. This is essential to distinguish between sincere and insincere promises. Moreover, as Searle notes, in sincere promises the speaker believes that it is possible for him to do *A* (whether *A* refers to performing an act or refraining from performing an act).

7. *S* intends that the utterance of *T* will place him under an obligation to do *A*. As Searle notes this is a necessary condition for making a promise. If the speaker can demonstrate that he did not have such intention, he can prove that the utterance was not a promise. Much of the litigation that arises when contractual relations between parties are disputed,

[57] Thanks to Marek Sergot for raising this example.

relies on establishing the parties' intentions in order to establish the nature of their relation.

8. *S* intends to produce in *H* the knowledge *K* that the utterance of *T* is to count as placing *S* under an obligation to do *A*. This captures the intuition that when the speaker means the utterance as a promise he intends to produce a certain illocutionary effect by getting the hearer to recognize his intention to produce such effect (the hearer's recognition of such effect is the perlocutionary aspect of the utterance). As Searle puts it "the speaker assumes that the semantic rules (which determine the meaning) of the expression uttered are such that the utterance counts as the undertaking of an obligation".

9. The semantical rules of the dialect spoken by *S* and *H* are such that *T* is correctly and sincerely uttered if and only if conditions 1–8 obtain.

Searle calls conditions 2 and 3 the *propositional content conditions*, 4 and 5 the *preparatory conditions*, 6 the *sincerity condition*, and 7 the *essential condition.* The following tables (adapted from Searle) summarise some other illocutionary acts in terms of their necessary and sufficient conditions, namely requests and assertions, which are mentioned in chapter 5:

Conditions	**Request**
Propositional Content	Future act *A* of *H*
Preparatory	1. *H* is able to do *A*. *S* believes *H* is able to do *A*. 2. It is not obvious to both *S* and *H* that *H* will do *A* in the normal course of events of his own accord.
Sincerity	*S* wants *H* to do *A*
Essential	Counts as an attempt to get *H* to do *A*
Comment	Order and command have the additional preparatory rule that *S* must be in a position of authority over *H*. Command may not have the 'pragmatic' condition requiring non-obviousness. The essential condition in both includes the authority relationship because the utterance counts as an attempt to get *H* to do *A* in virtue of the authority of *S* over *H*.

Table C-2 The structure of requests (Searle 1969)

Conditions	**Assert, state, affirm**
Propositional Content	Any proposition *p*.
Preparatory	1. *S* has evidence for the truth of *p*. 2. It is not obvious to both *S* and *H* that *H* knows *p*.
Sincerity	*S* believes *p*.
Essential	Counts as an undertaking to the effect that *p* represents an actual state of affairs.
Comment	Unlike *argue* these do not necessarily try to convince.

Table C-3 The structure of assertions (Searle 1969)