Argumentation

Trevor Bench-Capon, Department of Computer Science, The University of Liverpool, UK
Henry Prakken, Institute of Information and Computing Sciences, Utrecht University / Faculty of Law, University of Groningen, The Netherlands.

1 Introduction

A popular view of what Artificial Intelligence can do for lawyers is that it can do no more than deduce the consequences from a precisely stated set of facts and legal rules. This immediately makes many lawyers sceptical about the usefulness of such systems: this mechanical approach seems to leave out most of what is important in legal reasoning. A case does not appear as a set of facts, but rather as a story told by a client. For example, a man may come to his lawyer saying that he had developed an innovative product while working for Company A. Now Company B has made him an offer of a job, to develop a similar product for them. Can he do this? The lawyer firstly must interpret this story, in the context, so that it can be made to fit the framework of applicable law. Several interpretations may be possible. In our example it could be seen as being governed by his contract of employment, or as an issue in Trade Secrets law. Next the legal issues must be identified and the pros and cons of the various interpretations considered with respect to them. Does his contract include a non-disclosure agreement? If so, what are its terms? Was he the sole developer of the product? Did Company A support its development? Does the product use commonly known techniques? Did Company A take measures to protect the secret? Some of these will favour the client, some the Company. Each interpretation will require further facts to be obtained. For example, do the facts support a claim that the employee was the sole developer of the product? Was development work carried out in his spare time? What is the precise nature of the agreements entered into? Once an interpretation has been selected, the argument must be organised into the form considered most likely to persuade, both to advocate the client's position and to rebut anticipated objections. Some precedents may point to one result and others to another. In that case, further arguments may be produced to suggest following the favourable precedent and ignoring the unfavourable one. Or the rhetorical presentation of the facts may prompt one interpretation rather than the other. Surely all this requires the skill, experience and judgement of a human being? Granted that this is true, much effort has been made to design computer programs that will help people in these tasks, and it is the purpose of this chapter to describe the progress that has been made in modelling and supporting this kind of sophisticated legal reasoning.

We will review systems that can store conflicting interpretations and that can propose alternative solutions to a case based on these interpretations. We will also describe systems that can use legal precedents to generate arguments by drawing analogies to or distinguishing precedents. We will discuss systems that can argue why a rule should not be applied to a case even though all its conditions are met. Then there are systems that can act as a mediator between disputing parties by structuring and recording their arguments and responses. Finally we look at systems that suggest mechanisms and tactics for forming arguments.

Much of the work described here is still research: the implemented systems are prototypes rather than finished systems, and much work has not yet reached the stage of a computer programme but is stated as a formal theory. Our aim is therefore to give a flavour (certainly not a complete survey) of the variety of research that is going on and the applications that might result in the not too distant future. Also for this reason we will informally paraphrase example inputs and outputs of systems rather than displaying them in their actual, machine-readable format; moreover, because of space limitations the examples have to be kept simple.

2 Proof and Argument

Before proceeding it is worth considering the differences between a proof and an argument. In a proof we have a set of premises which entail a conclusion: if those premises are true then so must the conclusion be. In an argument, in contrast, although the premises give a reason for thinking that the conclusion is true, it remains possible that the falsity of the conclusion co-exists with the truth of the premises. Consider the argument *John is old because he is aged seventy-five*. This may well be a convincing argument, but it is not yet a proof. To turn it into a proof, we would need to add premises such as that *John is a man*, that *men over seventy are old*, and that *seventy five is greater than seventy*. Otherwise it could be the case that John is an adolescent tortoise, or that men cannot be considered old until they are eighty. Even the analytic statement of arithmetic is necessary for the proof. With an argument, however, we can leave many premises implicit since our object is to persuade, rather than compel, our hearer to accept our conclusion. So if the hearer is ready to accept that John is a man, and that men of seventy-five are old (whatever the threshold), our reason will be persuasive. Otherwise we must supply more premises to resolve the doubts. This ability to supply additional information is also characteristic of argument: whereas in a proof all the information is available at the outset, in an argument information may be accumulated gradually. This in turn enables us to see arguments as inherently defeasible: if I am told that John is seventy five, I may argue that he is old, assuming him to be a man. But when I am told that John is a tortoise, I will withdraw my argument.

To summarise: there are four characteristic differences between arguments and proofs:

- the goal of an argument is to persuade, whereas a proof compels acceptance;
- arguments leave things implicit, whereas proofs make everything explicit;
- more information can be added to arguments, whereas proofs begin from complete information;
- in consequence arguments are intrinsically defeasible.

3 Early systems for legal argumentation

In this section we will briefly discuss some of the early landmark systems for legal argumentation. All of them concern the construction of arguments and counterarguments.

3.1 Conflicting Interpretations

Systems to address conflicting interpretations of legal concepts go back to the very beginnings of AI and Law. Thorne McCarty (e.g. McCarty 1977; McCarty & Sridharan 1981) took as his key problem a landmark Supreme Court Case in US tax law which turned on differing interpretations of the concept of ownership, and set himself the ambitious goal of reproducing both the majority and the dissenting opinions expressing these interpretations. This required highly sophisticated reasoning, constructing competing theories and reasoning about the deep structure of legal concepts to map the specific situation onto paradigmatic cases. Although some aspects of the system were prototyped, the aim was perhaps too ambitious to result in a working system, certainly given the then current state of the art. This was not McCarty's goal, however: his motivation was to gain insight into legal reasoning through a computational model. McCarty's main contribution was the recognition that legal argument involves theory construction as well as reasoning with established knowledge. He summarises his position in McCarty (1995): "The task for a lawyer or a judge in a "hard case" is to construct a theory of the disputed rules that produces the desired legal result, and then to persuade the relevant audience that this theory is preferable to any theories offered by an opponent" (p. 285). Note also the emphasis on *persuasion*, indicating that we should expect to see argumentation rather than proof. Both the importance of theory construction and the centrality of persuasive argument are still very much part of current thinking in AI and Law.

Another early system was developed by Anne Gardner (1987) in the field of offer and acceptance in American contract law. The task of the system was "to spot issues": given an input case, it had to determine which legal questions arising in the case were easy and which were hard, and to solve the easy ones. The system was essentially rule based, and this simpler approach offered more possibilities for practical exploitation than did McCarty's system. One set of rules was derived from the Restatement of Contract Law, a set of 385 principles abstracting from thousands of contract cases. These rules were intended to be coherent, and to yield a single answer if applicable. This set of rules was supplemented by a set of interpretation rules derived from case law, common sense and expert opinion, intended to link these other rules to the facts of the case. Gardner's main idea was that easy questions were those where a single answer resulted from applying these two rule sets, and hard questions, or issues, were either those where no answer could be produced, because no interpretation rule linked the facts to the substantive rules, or where conflicting answers were produced by the facts matching with several rules. Some of the issues were resolved by the program with a heuristic that gives priority to rules derived from case law over restatement and commonsense rules. The rationale of this heuristic is that if a precedent conflicts with a rule from another source, this is usually because that rule was set aside for some reason by the court. The remaining issues were left to the user for resolution.

Consider the following example, which is a very much simplified and adapted version of Gardner's own main example[1]. The main restatement rule is

R1: An offer and an acceptance constitute a contract

Suppose further that there are the following commonsense (C) and expert (E) rules on the interpretation of the concepts of offer and acceptance:

C1: A statement "Will supply ..." in reply to a request for offer is an offer.
C2: A statement "Will you supply ..." is a request for offer.
C3: A statement "I accept ..." is an acceptance.
E1: A statement "I accept" followed by terms that do not match the terms of the offer is not an acceptance.

Suppose that Buyer sent a telegram to Seller with "Will you supply carload salt at $2.40 per cwt?" to which Seller replied with "Will supply carload at $2.40, terms cash on delivery", after which Buyer replied with her standard "Purchase Order" indicating "I accept your offer of 12 July" but which also contained a standard provision "payment not due until 30 days following delivery".

Applying the rules to these events, the "offer" antecedent of R1 can be established by C1 combined with C2, since there are no conflicting rules on this issue. However, with respect to the "acceptance" antecedent of R1 two conflicting rules apply, viz. C3 and E1. Since we have no way of giving precedence to C3 or E1, the case will be a hard one, as there are two conflicting notions of "acceptance". If the case is tried and E1 is held to have precedence, E1 will now be a precedent rule, and any subsequent case in which this conflict arises will be easy, since, as a precedent rule, E1 will have priority over C3.

There is evidence that Gardner's approach may lead to useful applications. For example, we can consider the system built by Kees de Vey Mestdagh (1998) in the context of a civil law jurisdiction. He built a system that provides knowledge-based support to officers deciding on environmental permit applications. The system contains provisions from Dutch environmental law as well as possibly conflicting rules on the interpretation of concepts occurring in these provisions. In its output the system provides the user with the various possible decisions on a permit application. The system was fully implemented and evaluated in several controlled experiments in which the system's output was assessed by a number of domain experts. In the main experiment the system was provided with the data of 35 simple and 5 complex actual cases, consisting of in total 430 decisions. The system could ask for additional data. The system improved on the human decision maker for 13% of the decisions, it suggested valid alternatives in addition to the human decision for 18% of the decisions, and took the same decision for the remaining 69% of the decisions.

3.2 Reasoning With Precedents

[1] We in particular abstract from Gardner's refined method for representing knowledge about (speech act) events.

The systems described in the last section do recognise the importance of precedent cases as a source of legal knowledge, but they make use of them by extracting the rationale of the case and encoding it as a rule. To be applicable to a new case, however, the rule extracted may need to be analogised or transformed to match the new facts. Nor is extracting the rationale straightforward: judges often leave their reasoning implicit and in reconstructing the rationales a judge could have had in mind there may be several candidate rationales, and they can be expressed at a variety of levels of abstraction. These problems occur especially in so-called "factor-based domains" (Branting, 2003), i.e., domains where problems are solved by considering a variety of factors that plead for or against a solution. In such domains a rationale of a case often just expresses the resolution of a particular set of factors in a specific case. A main source of conflict in such domains is that a new case often does not exactly match a precedent but will share some features with it, lack some of its other features, and/or have some additional features. Moreover, cases are more than simple rationales: matters such as the context and the procedural setting can influence the way the case should be used. In consequence, some researchers have attempted to avoid using rules and rationales altogether, instead representing the input, often interpreted as a set of factors, and the decisions of cases, and defining separate argument moves for interpreting the relation between the input and decision (e.g. Loui & Norman, 1995; Aleven, 1997, both to be discussed below). This approach is particularly associated with researchers in America, where the common law tradition places a greater stress on precedent cases and their particular features than is the case with the civil law jurisdictions of Europe. None the less cases are also used in civil law jurisdictions and the reasoning techniques are similar. For a discussion of the way in which cases are used in a variety of Civil Law Jurisdictions see (MacCormick and Summers 1997).

The most influential system of this sort is HYPO (Ashley 1990), developed by Edwina Rissland and Kevin Ashley in the domain of US Trade Secrets Law, which can be construed as a factor-based domain[2]. In HYPO cases are represented according to a number of *dimensions*. A dimension is some aspect of the case relevant to the decision. For example, the security measures taken by the plaintiff is one such dimension. One end of the dimension represents the most favourable position for the plaintiff (e.g. specific non-disclosure agreements), while the other end represents the position most favourable to the defendant (e.g. no security measures at all). Typically a case will lie somewhere between the two extremes and will be more or less favourable accordingly. HYPO then uses these dimensions to construct *three-ply arguments*. First one party (say the plaintiff) cites a precedent case decided for that side and offers the dimensions it shares with the current case as a reason to decide the current case for that side. In the second ply the other party responds either by citing a counter example, a case decided for the other side which shares a different set of dimensions with the current case, or distinguishing the precedent by pointing to features which make the precedent more, or the current case less, favourable to the original side. In the third ply the original party attempts to rebut the arguments of the second ply, by distinguishing the counter examples, or by citing additional precedents to emphasise the strengths or discount the weaknesses in the original argument.

[2] HYPO and CATO are described in considerable detail elsewhere in this volume, in section 5.1 of the chapter by Kevin Ashley. Here we will summarise the features that were most important for subsequent developments concerning argumentation in AI and Law.

Subsequently Ashley went on, with Vincent Aleven, to develop CATO (most fully reported in Aleven 1997), a system designed to help law students to learn to reason with precedents. CATO simplifies HYPO in some respects but extends it in others. In CATO the notion of dimensions is simplified to a notion of *factors*. A factor can be seen as a specific point of the dimension: it is simply present or absent from a case, rather than present to some degree, and it always favours *either* the plaintiff or defendant. A new feature of CATO is that these factors are organised into a hierarchy of increasingly abstract factors, so that several different factors can be seen as meaning that the same abstract factor is present. One such abstract factor is that the defendant used questionable means to obtain the information, and two more specific factors indicating the presence of this factor are that the defendant deceived the plaintiff and that the defendant bribed an employee of the plaintiff: both these factors of course favour the plaintiff. The hierarchy allows for argument moves that interpret the relation between a case's input and its decision, such as emphasising or downplaying distinctions. To give an example of downplaying, if in the precedent defendant used deception while in the new case instead defendant bribed an employee, thn a distinction made by the defendant at this point can be downplayed by saying that in both cases the defendant used questionable means to obtain the information. To give an example of emphasising a distinction, if in the new case defendant bribed an employee of plaintiff while in the precedent no factor indicating questionable means was present, then the plaintiff can emphasise the distinction "unlike the precedent, defendant bribed an employee of plaintiff" by adding "and therefore, unlike the precedent defendant used questionable means to obtain the information".

Perhaps the most elaborate representation of cases was produced in Karl Branting's (2000) Grebe system in the domain of industrial injury, where cases were represented as semantic networks. The program matched portions of the network for the new case with parts of the networks of precedents, to identify appropriate analogies. Grebe is described in detail in section 5.2 of the chapter in this volume by Kevin Ashley, and so we will say no more about it here.

HYPO, in particular, was highly influential, both in the explicit stress it put on reasoning with cases as constructing *arguments*, and in providing a dialectical structure in which these arguments could be expressed, anticipating much other work on dialectical procedures.

4 Logical accounts of reasoning under disagreement

The systems discussed in the previous section were (proposals for) implemented systems, based on informal accounts of some underlying theory of reasoning. Other AI & Law research aims at specifying theories of reasoning in a formal way, in order to make general reasoning techniques from logic available for implementations. To some readers this may seem surprising at first sight: it is often thought that in the face of inconsistency logic would be useless, since according to standard deductive logic from a contradiction everything can be derived (*Ex Falso Sequitur Quodlibet*). However, logicians and AI researchers have found ways to cope with this, in the study of so-called nonmonotonic logics. The main idea is that when faced with an inconsistent body of information, attention is paid only to those logical derivations that can be made from a consistent subset of the information. Such derivations can be regarded as arguments, and derivations based on other, perhaps inconsistent, subsets as counterarguments. This idea can be developed in various ways: a detailed discussion of which is beyond the scope of this paper. See e.g. Prakken & Sartor (2002) for a survey.

The first AI & Law proposals in this vein (for example, Gordon, 1991 and Prakken, 1993) can be regarded as formal counterparts of Gardner's ideas on issue spotting. Recall that Gardner allows for the presence in the knowledge base of conflicting rules governing the interpretation of legal concepts and that she defines an issue as a problem to which either no rules apply at all, or conflicting rules apply. Now in logical terms an issue can be defined as a proposition such that either there is no argument about this proposition or there are both arguments for the proposition and for its negation.

Some more recent work in this research strand has utilised a very abstract AI framework for representing systems of arguments and their relations developed by Dung (1995). For Dung, the notion of argument is entirely abstract: all that can be said of an argument is which other arguments it attacks, and which it is attacked by. Given a set of arguments and the attack relations between them, it is possible to determine which arguments are acceptable. Thus an argument which is not attacked will be acceptable, but if an argument has attackers it is acceptable only if it can be defended against these attackers by acceptable arguments which in turn attack those attackers. Variations in the semantics arise: for example according to whether an argument is allowed to defend itself. This framework has proved a fruitful tool for understanding nonmonotonic logics and their computational properties. Dung's framework has also been made use of in AI and Law. It was first applied to the legal domain by Prakken & Sartor (1996), who defined a logic for reasoning with conflicting rules as an instantiation of Dung's framework. In that paper Prakken and Sartor define a structure for arguments (basically a sequence of rule applications), and also define the ways in which arguments may attack one another. They use grounded semantics, where arguments cannot defend themselves, to determine acceptability. Bench-Capon has explored the potential of the fully abstract version of the framework to represent a body of case law in Bench-Capon (2002). Bench-Capon uses preferred semantics, where arguments can defend themselves: in case of mutual attack this gives rise to multiple sets of acceptable arguments, which can explain differences in the application of law in different jurisdictions, or at different times in terms of social choices. Dung's framework has also been extended to include a more formal consideration of social values (discussed in section 4.1 below) in Bench-Capon (2003). This allows an argument to resist an attack if it is founded on a more esteemed value than its attacker. In such a framework, given an ordering on social values, there will be a unique set of acceptable arguments, even when preferred semantics is used.

4.1 Reasoning About Conflicting Rules

Generally speaking, the proposed systems discussed so far attempt to identify conflicting interpretations and arguments, but do not attempt to resolve them, leaving it to the user to choose which argument will be accepted. As we saw above, Gardner's system went somewhat further in that it gave priority to rules derived from case law over restatement and commonsense rules. Thus her system was able to solve some of the cases to which conflicting rules apply. This relates to much logical work in Artificial Intelligence devoted to the resolution of rule conflicts in so-called commonsense reasoning. If we have a rule that birds can fly and another that ostriches cannot fly, we do not want to let the user decide whether Cyril the ostrich can fly or not: we want the system to say that he cannot, since an ostrich is a specific kind of bird. Naturally attempts have been made to apply these ideas to law.

One approach was to identify general principles used in legal systems to establish which of two conflicting rules should be given priority. These principles included preferring the more specific rule (as in the case of the ostrich above, or where a law expresses an exception to a general provision), preferring the more recent rule, or preferring the rule deriving from the higher legislative authority (for instance, ‘federal law precedes state law’). To this end the logics discussed above were extended with the means to express priority relations between rules in terms of these principles so that rule conflicts would be resolved. Researchers soon realised, however, that general priority principles can only solve a minority of cases. Firstly, as for the specificity principle, whether one rule is more specific than another often depends on substantive legal issues such as the goals of the legislator, so that the specificity principle cannot be applied without an intelligent appreciation of the particular issue. Secondly, general priority principles usually only apply to rules from regulations and not to, for instance, case rationales or interpretation rules derived from cases. Accordingly, in many cases the priority of one rule over another can be a matter of debate, especially when the rules that conflict are unwritten rules put forward in the context of a case. For these reasons models of legal argument should allow for arguments about which rule is to be preferred.

As an example of arguments about conflicting case rationales, consider three cases discussed in, amongst others, Berman and Hafner (1993), Bench-Capon and Sartor (2001), Prakken (2002) and Bench-Capon and Sartor (2003), concerning the hunting of wild animals. In all three cases, the plaintiff (P) was chasing wild animals, and the defendant (D) interrupted the chase, preventing P from capturing those animals. The issue to be decided is whether or not P has a legal remedy (a right to be compensated for the loss of the game) against D. In the first case, *Pierson v Post*, P was hunting a fox on open land in the traditional manner using horse and hound, when D killed and carried off the fox. In this case P was held to have no right to the fox because he had gained no possession of it. In the second case, *Keeble v Hickeringill*, P owned a pond and made his living by luring wild ducks there with decoys, shooting them, and selling them for food. Out of malice, D used guns to scare the ducks away from the pond. Here P won. In the third case, *Young v Hitchens*, both parties were commercial fisherman. While P was closing his nets, D sped into the gap, spread his own net and caught the fish. In this case D won. The rules we are concerned with here are the rationales of these cases:

R1. Pierson: If the animal has not been caught, the defendant wins
R2 Keeble: If the plaintiff is pursuing his livelihood, the plaintiff wins
R3 Young: If the defendant is in competition with the plaintiff and the animal is not caught, the defendant wins.

Note that R1 applies in all cases and R2 in both Keeble and Young. In order to explain the outcomes of the cases we need to be able to argue that R3 > R2 > R1. To start with, note that if, as in HYPO, we only look at the factual similarities and differences, none of the three precedents can be used to explain the outcome of one of the other precedents. For instance, if we regard Young as the current case, then both Pierson and Keeble can be distinguished. A way of arguing for the desired priorities, first mooted in Berman and Hafner, 1993, is to refer to the purpose of the rules, in terms of the social values promoted by following the rules.

The logic of Prakken & Sartor (1996) provides the means to formalise such arguments. Consider another case in which only plaintiff was pursuing his livelihood and in which the animal was not caught. In the following (imaginary) dispute the parties reinterpret the precedents in terms of the values promoted by their outcomes, in order to find a controlling precedent (we leave several details implicit for reasons of brevity; a detailed formalisation method can be found in Prakken, 2002; see also and Bench-Capon & Sartor, 2003).

Plaintiff: I was pursuing my livelihood, so (by *Keeble*) I win
Defendant: You had not yet caught the animal, so (by *Pierson*) I win
Plaintiff: following *Keeble* promotes economic activity, which is why *Keeble* takes precedence over *Pierson*, so I win.
Defendant: following *Pierson* protects legal certainty, which is why *Keeble* does not take precedence over *Pierson*, so you do not win.
Plaintiff: but promoting economic activity is more important than protecting legal certainty since economic development, not legal certainty is the basis of this country's prosperity. Therefore, I am right that *Keeble* takes precedence over *Pierson*, so I still win.

This dispute contains priority debates at two levels: first the parties argue about which case rationale should take precedence (by referring to values advanced by following the rationale), and then they argue about which of the conflicting preference rules for the rationales takes precedence (by referring to the relative order of the values). In general, a priority debate could be taken to any level and will be highly dependent on the context and jurisdiction. Various logics proposed in the AI & Law literature are able to formalise such priority debates, such as Gordon (1995), Prakken & Sartor (1996), Hage (1996), Verheij (1996) and Kowalski & Toni (1996). (In fact, Hage and Verheij define a variant of these methods in which the comparison is not between individual conflicting rules but between the sets of all rules pleading for or against a proposition.)

4.2 Other arguments about rules

Besides priority debates in case of conflicting rules, these logics can also model debates about certain properties of rules, such as their legal validity or their applicability to a legal case. The most fully developed logical theory about what it takes to apply a rule is reason-based logic, developed jointly by Jaap Hage and Bart Verheij (e.g. Hage 1996, Verheij, 1996). They claim that applying a legal rule involves much more than subsuming a case under the rule's conditions. Their account of rule application can be briefly summarised as follows. First in three preliminary steps it must be determined whether the rule's conditions are satisfied, whether the rule is legally valid, and whether the rule's applicability is not excluded in the given case by, for instance, a statutory exception. If these questions are answered positively (and all three are open to debate), it must finally be determined that the rule can be applied, i.e., that no conflicting rules or principles apply. On all four questions reason-based logic allows reasons for and against to be provided and then weighed against each other to obtain an answer.

Consider by way of illustration a recent Dutch case (HR 7-12-1990, *NJ* 1991, 593) in which a male nurse aged 37 married a wealthy woman aged 97 whom he had been nursing for several years, and killed her five weeks after the marriage. When the woman's matrimonial estate was divided, the issue arose whether the nurse could retain his share. According to the relevant statutes on Dutch matrimonial law the nurse was entitled to his share since he had been the woman's husband. However, the court refused to apply the law, on the grounds that applying it would be manifestly unjust. Let us assume that this was in turn based on the legal principle that no one shall profit form his own wrongdoing (the court did not explicitly state this). In reason-based logic this case could be formalised as follows (again the full details are suppressed for reasons of brevity).

Claimant: Statutory rule R is a valid rule of Dutch law since it was enacted according to the Dutch constitution and never repealed. All its conditions are satisfied in my case, and so it should be applied to my case. The rule entitles me to my late wife's share in the matrimonial estate. Therefore, I am entitled to my wife's share in the matrimonial estate.
Defendant: Applying rule R would allow you to profit from your own wrongdoing: therefore rule R should not be applied in this case.
Court: The reason against applying this rule is stronger than that for applying the rule, and so the rule does not apply.

Of course, in the great majority of cases the validity or applicability of a statute rule is not at issue but instead silently presumed by the parties (recall the difference between arguments and proofs described in the introduction). The new logical techniques alluded to above can also deal with such presumptions, and they can be incorporated in reason-based logic.

One way to argue about the priority of arguments is to claim that the argument is preferred if it is grounded in the better or more coherent legal theory[3]. While there has been considerable progress in seeing how theories can be constructed on the basis of a body of past cases, evaluation of the resulting theories in terms of their coherence is more problematic, since coherence is a difficult notion to define precisely[4]. Bench-Capon and Sartor (2003) describe some features of a theory which could be used in evaluation, such as simplicity of a theory or the number of precedent cases explained by the theory. As an (admittedly somewhat simplistic) example of the last criterion, consider again the three cases on hunting animals, and imagine two theories that explain the case decisions in terms of the values of promotion of economic activity and protection of legal certainty. A theory that gives precedence to promoting

[3] There is, of course, a debate in legal theory as to how we can provide an epistemology of law, and coherence is only one position. Coherence is discussed here as it is the position which has received most attention in AI and Law.
[4] For fuller discussions of coherence, see Peczenik (1996), and Mommers (2002), chapter 2.

economic activity over protecting legal certainty explains all three precedents while a theory with the reverse value preference fails to explain *Keeble*. The first theory is therefore on this criterion the more coherent one. However, how several coherence criteria are to be combined is a matter for further research. For an attempt to give a metric for coherence, see Bench-Capon and Sartor (2001). Coherence is also discussed in Hage (2001), where coherence is treated mainly in terms of respecting *a fortiori* arguments.

5 Dialogue and Mediation Systems

Implicit in the notion of argument is that there are two parties with opposing views. Already in HYPO there is the dialectical structure of point, counter point and rebuttal, and most logics for argumentation discussed above also have this dialectical flavour. It is therefore a natural step to make this dialogical structure explicit, and to build systems to conduct or mediate dialogues between the opposed parties. Such dialogue systems also provide the opportunity to model the procedure under which a dispute is conducted, and the context in which information is introduced to a dispute. Taking a procedural point of view forces us to think about matters such as burden of proof, admissibility of evidence, agreed and contested points, and the role of a neutral third party to arbitrate the dispute.

One of the first such systems in AI and Law was Tom Gordon's (1995) *Pleadings Game*, which embodies an idealised model of civil pleadings in common law systems. The objective of the system is to extend the issue-spotting task of Gardner's program to a dialogical setting. It is to allow two human parties to state the arguments and facts that they believe to be relevant, so that they can determine where they agree and where they disagree. The residual disagreements will go on to form the issues when the case is tried. The system plays two roles in this process: it acts as a referee to ensure that the proper procedure is followed, and records the facts and arguments that are presented and what points are disputed, so as to identify the issues that require resolution. The Pleadings Game has a built-in proof mechanism for an argumentation logic, which is applied to check the logical well-formedness of the arguments stated by the user, and to compute which of the stated arguments prevail, on the basis of the priority arguments also stated by the user and a built-in specificity checker. The main addition to Gardner's system is that in the Pleadings Game not only the content of the arguments is relevant but also the attitudes of the parties expressed towards the arguments and their premises.

Let us illustrate this with the following simplified dispute, based on the example that we above used to illustrate Gardner's system.

Plaintiff: I claim (1) we have a contract
Defendant: I deny 1
Plaintiff: We have a valid contract since (2) I made an offer and (3) you accepted it, so we have a contract.
Defendant: I concede 2 but I deny 3.
Plaintiff: (4) you said "I accept...", so by C1 you accepted my offer.

Defendant: I concede 4 and C1, but (5) my statement “I accept ...” was followed by terms that do not match the terms of your offer. So by P1 (which takes priority over C1) I (6) did not accept you offer.
Plaintiff: I concede P1 and that P1 takes priority over C1 but I deny 5.
Defendant: (7) you required payment upon delivery while (8) I offered payment 30 days following delivery, so there is a mismatch between our terms.
Plaintiff: I concede (7) and the argument but I deny (8).

At this point, there is one argument for the conclusion that a contract was created, based on the premises 2, 4 and C1 (note that plaintiff left R1 implicit and defendant silently agreed with this). The intermediate conclusion (3) of this argument that there was an acceptance is defeated by a counterargument based on premises 7, 8 and P1. So according to a purely logical analysis of the dispute the case is easy, having as outcome that no contract exists between the parties. This agrees with Gardner’s treatment of the example. However, in the Pleadings Game it also matters that the plaintiff has denied defendant’s claim (8). This is a factual issue making the case hard, and which has to be decided in court.

The Pleadings Game was fully implemented, but purely as an experimental system: in particular the arguments had to be presented in a complicated logical syntax so that they could be handled by the underlying proof mechanism. The trade-off between ease of use and the ability of the system to process the information it receives remains a difficult problem for such systems.

Following Gordon’s work, a number of other systems for dialogue were produced.

Lodder’s (1999) Dialaw is a dialogue game that combines the notion of propositional commitment (see e.g. Walton and Krabbe, 1995) with Hage and Verheij’s Reason Based Logic. The game has two participants, who can use locutions for claiming a proposition and for challenging, conceding and retracting a claimed proposition. Arguments are constructed implicitly, by making a new claim in reply to a challenge. Arguments can also be about the procedural correctness of dialogue moves. Each dialogue begins with a claim of one player, and then the turn usually switches after each move. When the commitments of one player logically imply a claim of the other player, the first player must either concede it or retract one of the implying commitments. A dialogue terminates if no disagreement remains, i.e., if no commitment of one player is not also a commitment of the other. The first player wins if at termination he is still committed to his initial claim, the second player wins otherwise.

Bench-Capon et al.’s (2000) TDG is intended to produce more natural dialogues than the “stilted'” ones produced by systems such as the Pleadings Game and Dialaw. To this end, its speech acts are based on Toulmin’s (1958) well-known argument scheme. In this scheme, a *claim* is supported by *data*, which support is *warranted* by an inference licence, which is *backed* by grounds for its acceptance; finally, a claim can be attacked with a *rebuttal*, which itself is a claim and thus the starting point of a counterargument. Arguments can be chained by regarding data also as claims, for which data can in turn be provided. TDG has speech acts for asking for and providing these elements of an argument; a dialogue starts with a claim and then the protocol supports a dialogue which constructs a Toulmin structure whilst subjecting it to a top-down critical examination.

Finally, Prakken (2001) proposes an idealised formal model of Dutch civil procedure, which aims to model the notion of burden of proof and to give a more realistic account of the role of third parties in a dispute. To this end, a dialogue game is developed that resembles the Pleadings Game but that involves a third party who can use speech acts for, among other things, allocating the burden of proof when a claim is challenged.

6 Tactics for Dispute

Once arguments are placed in a dialogical setting, it becomes apparent that at various points of the dialogue, the parties will have a choice of moves by which to attack their opponent or defend their own arguments. Questions then arise as to which moves are available to construct, attack and defend arguments, and whether there are principles to guide the choice of move. In fact, the implemented dialogue systems of the previous section do not address these questions, because they are intended to act as a mediator between two human players. The responsibility of the system is thus limited to enforcing the rules of the game, while strategy and tactics are the responsibility of the human users.

In their work on the CABARET system, David Skalak and Edwina Rissland (1992) attempted to identify arguments that could be made in a dispute using rules and cases.[5] They begin by identifying a number of forms of argument, and then describe argument strategies to be used according to the context of the dispute. For example, if the current case matches with most but not all the features of some statutory rule that one wishes to use, the rule must be broadened so as to make the rule applicable to the case. Or if a rule is applicable to the case under consideration but would be unfavourable, that rule needs to be discredited. They then identify the moves that can be made to realise the strategies, depending on the disposition of the precedent, and whether the precedent does or does not establish the desired consequent. One move to broaden a rule is to find a precedent that also lacked the missing features but in which the conclusion of the rule was nevertheless drawn. To discredit a rule one can try to find a precedent case in which it was not followed even though all its conditions were satisfied in the case. Finally they identify a number of primitive operations in terms of which the moves can be realised. These operations include all moves that can be made in HYPO with cases. All of this is then brought together in a decision tree which suggests which strategy should be adopted, which moves need to be used to fulfil it and which primitives will enable the required moves.

Ron Loui and Jeff Norman (1995) take this approach a step further in their formal model of the use of rationales in disputes. They allow for a position under attack to be first restated, in order to make the attack more effective. For example if an argument using a rationale *if P then Q* is to be attacked, it may be helpful to restate this as *if P*

[5] For a fuller discussion of CABARET, see section 5.3 of the chapter by Kevin Ashley.

then R and *if R then Q*, and to provide a counter example to *if P then R*. They provide a number of other examples of rationales and tactics for attacking them.

CABARET, by distinguishing different kinds of building materials, and providing different moves and attacks appropriate to each kind, can produce its elegant classification of strategies. The central idea of distinguishing different kinds of premises and different ways of dealing with them is explicitly addressed by work on *argument schemes*, which we discuss in the next section.

7 Argument Schemes

In a logical proof we have a set of premises and a conclusion which is said to follow from them. The premises are considered to be entirely homogenous. Many of the systems discussed so far likewise make no distinctions among their premises. In arguments expressed in natural language in contrast we can typically see the premises as playing different roles in the argument. By identifying these roles, we can present the arguments in a more readily understandable fashion, and also identify the various different ways in which the argument may be attacked. Structuring the argument in this way produces an argument scheme. Analysing legal reasoning in terms of argument schemes produces a taxonomy of arguments, which may provide useful guidance for building implemented argumentation systems, analogous to the guidance provided by domain ontologies for building knowledge-based systems (cf. e.g. Mommers, 2002).

One argument scheme that has been widely used in AI and Law is that devised by Stephen Toulmin (1958). As explained above, this distinguishes between the *data* supporting the argument, the *warrant* which licences the drawing of the conclusion, the *backing* which justifies the warrant, and a *rebuttal* which specifies exceptions to the warrant. This has been mainly used to present arguments to users, as in PLAID (Bench-Capon & Staniford, 1995) and SPLIT UP (Zeleznikow & Stranieri, 1995), but it has also been used as the basis of a dialogue game, Bench-Capon's TDG, in which the moves of the game relate to providing various elements of the scheme.

While Toulmin attempts to supply a general scheme for arguments, others have attempted to classify arguments in terms of various specific schemes (e.g. Walton 1996). One of the schemes discussed by Walton (pp. 61-63) is the scheme of arguments from the position to know:

> Person *W* says that *p*
> Person *W* is in the position to know about *p*
> *Therefore*, *p*

Walton also discusses two special versions of this scheme for witness and expert testimonies. Clearly, these schemes are very relevant for evidential legal reasoning. Another scheme discussed by Walton (pp. 75-77) is the scheme from good (or bad) consequences:

> If *A* is brought about, then good (bad) consequences will (may plausibly) occur.

Therefore, *A* should (not) be brought about.

One instantiation is adapted from a recent discussion in Dutch privacy law whether email addresses are personal data.

> If the term "personal data" of the Dutch Data Protection Act is interpreted to include email addresses, then legal measures against spam become possible, which is good.
> *Therefore*, the term "personal data" of the Dutch Data Protection Act" should be interpreted to include email addresses.

Argument schemes are not classified according to their logical form but according to their content. Many argument schemes in fact express epistemological principles (such as the scheme from the position to know) or principles of practical reasoning (such as the scheme from consequences). Accordingly, different domains may have different sets of such principles. Each argument scheme comes with a customised set of critical questions that have to be answered when assessing whether their application in a specific case is warranted. Thus with argument schemes it becomes clear that the different premises are each associated with their own particular types of attack, in contrast to the purely logical systems in which attacks are uniform. Some of these questions pertain to acceptability of the premises, such as 'is *W* in the position to know about *p*?' or 'is the possibility to use legal means against spam really good?". Other critical questions point at exceptional circumstances in which the scheme may not apply, such as 'is *W* sincere?' or "are there better ways to bring about these good consequences?". Clearly, the possibility to ask such critical questions makes argument schemes defeasible, since negative answers to such critical questions are in fact counterarguments, such as "Person *W* is not sincere since he is a relative of the suspect and relatives of suspects tend to protect the suspect". Another reason why argument schemes are defeasible is that they may be contradicted by conflicting applications of the same or another scheme. For instance, a positive instance of the scheme from consequences can be attacked by a negative instance of the same scheme, such as by "interpreting email addresses as personal data also has bad consequences, since the legal system will be flooded with litigation, so the term "personal data" should not be interpreted to include email addresses". Or one person in a position to know (say an eyewitness) may have said that the suspect was at the crime scene while another eyewitness may have said that the suspect was not at the crime scene.

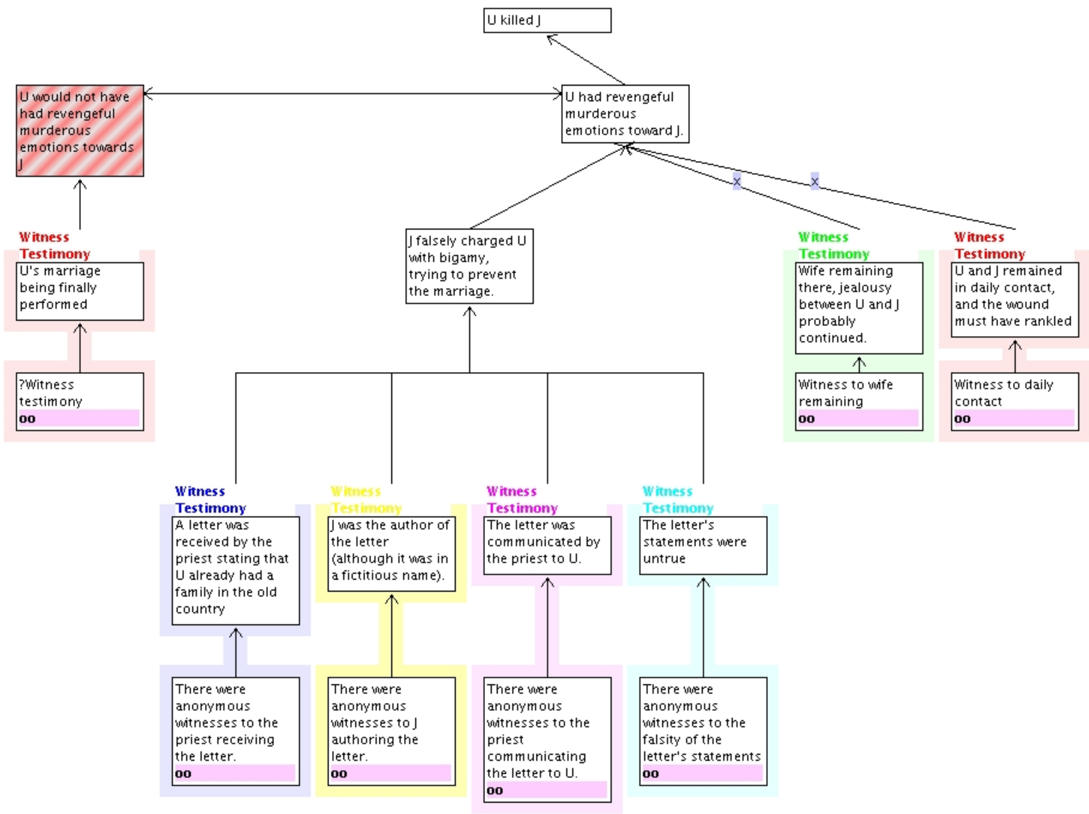
Until recently, except for the use of Toulmin, argument schemes did not receive much explicit attention within AI & Law, although implicit appeal can be seen as made to them in many of the systems discussed above. For example, HYPO identifies the two ways in which the citation of a precedent may be attacked, and reason-based logic identifies ways to reason about the application of legal rules. Two recent attempts to make explicit use of argumentation schemes are Greenwood et al. (2003), employing an extended version of the scheme from consequences and Bex et al. (2003), modelling several schemes for reasoning about evidence.

8 Systems To Structure Argument

Arguments can often be rather complex, so that understanding the web of relationships becomes difficult. There is clear potential for computers to provide a means of addressing this problem. The idea of providing a visual means of structuring legal arguments is not new to the legal field: as early as the 1930s John Henry Wigmore (1931) produced a graphical notation for depicting legal arguments and their relations of support and attack, so as to make sense of a mass of evidence. In this way the relationships between the evidence and the point to be proven, and the ways in which the chain of reasoning could be attacked could be clearly seen.

In Wigmore's days the only way to draw such graphs was with pencil and paper, which perhaps explains why his method was forgotten until David Schum and Peter Tillers (1991) saw the potential of the computer for supporting the drawing and manipulation of such graphs. They proposed a software system MarshalPlan for visualising preliminary fact investigation based on Wigmore's diagrams. Two other systems within AI & Law that provide support for the graphical structuring of argumentation are Bart Verheij's (1999) ArguMed system and Loui et al.'s (1997) Room 5 system, which replaces ArguMed's (and MarshalPlan's) graph structures with encapsulated text boxes, to avoid "pointer spagetthi". Finally, Chris Reed's Araucaria system (Reed & Rowe, 2001) combines an ArguMed-like graphical notation with means to label the arguments as instances of predefined argumentation schemes, which are stored in a database together with their critical questions.

By way of example, we present a screen shot from Araucaria as applied to reasoning about evidence in a murder case also visualised by Wigmore (1931) (taken from Bex et al., 2003, as is the following explanation). In this case, a farm labourer Umilian (U) was accused of killing his colleague Jedrusik (J). The alleged motive was that J had tried to prevent U's marriage with the farm maid by sending a letter to the priest that U already had a wife. When the priest found that the accusations were false, he proceeded to marry U to the farm maid, but U remained angry at J and made various threats of vengeance against him. The purpose of this chart is to visualise how, according to the analyst, the available evidence (several witness testimonies) is relevant for the alleged motive that "U had revengeful murderous emotions towards J". In the chart, vertical and diagonal links represent support relations between propositions. For instance, the proposition "J falsely charged U with bigamy, trying to prevent the marriage" is supported by a conjunction of four propositions, each of which is in turn supported by a witness testimony. Horizontal links capture attack relations between propositions. For instance, the nodes "U had revengeful murderous emotions towards J" attacks and is attacked by the node "U would not have had revengeful murderous emotions towards J". The various colourings around inference steps indicate the types of argument schemes used in these steps. In this graph all inference steps are either untyped or of the witness testimony type.



Argument structuring systems have uses in areas where the clear presentation of the argument is of prime importance. They could be used in preliminary fact investigation (see MarshalPlan), in teaching (many argument structuring systems outside the legal domain have been developed especially for teaching), for case management or for mediation in Online Dispute Resolution (Lodder, 2001). In all these cases, the usefulness of such systems might be increased by integrating them with documentary sources. For instance, when supporting preliminary fact investigation, the structured evidential arguments could be linked to police documents containing the available evidence. Or when used for case management, the structured arguments could be linked to the case files. Or when a structuring system is used for teaching the analysis of a case decision, the structured arguments could be linked to the corresponding fragment in the case decisions in the casebook used by the students. Work on argumentation schemes can further augment the usefulness of such systems. When constructing arguments, argument schemes provide a repertoire of forms of argument to be considered, and a template prompting for the pieces that are needed; when attacking arguments they provide a set of critical questions that can identify potential weaknesses in the opponents case. Araucaria provides an example of a research system pointing in this direction.

9 Prospects for Practical Realisation

Currently, all the systems using techniques drawn from work on AI and Law that are in practical use, such as the systems developed by Softlaw in Australia and MRE in the Netherlands, make use only of rather straightforward deductive methods. These methods are entirely appropriate for the routine tasks these systems are designed to support. The techniques described in this chapter have the aim to extend the capacity for support beyond these routine tasks. They are still at the research stage, but must play an important role if the scope of computer support is to be extended. In this section we will discuss some of the more sophisticated tasks which could be supported by argumentation techniques.

Kevin Ashley's book on HYPO (Ashley, 1990) opens with a description of an advocate charged with preparing a case at short notice. His vision suggests that a system which is able to accept the facts of the case and then generate arguments for the two sides to the case and counterarguments to them, together with the precedents on which they are based, would provide the answer to such an advocate's needs. We have discussed several systems which could provide such support, but all of them are critically dependent on the possibility of acquiring a large amount of knowledge and representing it in a form which can be manipulated by the system. The same holds for decision support systems. This is an instance of the well known "knowledge acquisition bottleneck", which has proved a major barrier to the practical exploitation of intelligent techniques in many domains. At one time it was expected that this barrier would be lower in the legal domain because of the availability of documented sources, but this has proven to be so only for routine, regulation-dependent tasks.

There are two ways to cope with the bottleneck problem: to solve it or to avoid it. The problem could be solved by automating the process of knowledge acquisition. This would, however, require major advances in machine learning and natural language understanding. Moreover, if we wish to acquire knowledge from sources which need considerable interpretation – such as the case decisions which play a significant role in argumentation – the problems may well appear insurmountable. To avoid the bottleneck we must find an area or task in which the amount of knowledge to be acquired can be kept within reasonable bounds. Possibly it is for this reason that since HYPO work has tended to address more constrained, less ambitious tasks where a limited amount of knowledge can still form the basis of an effective system. One approach is to focus on more tractable aspects of the task, so that it might take the form of providing tools to support information retrieval and structuring of arguments. Another is to constrain the application, for example building a teaching system, where the completeness of the knowledge ceases to be an issue. We will discuss these two possibilities below.

Argument structuring systems (discussed above in Section 8) are an example of the attempt to focus on more tractable aspects of a task. They do not require a knowledge base since the arguments are provided by the user. A commercial argument structuring system currently being developed is Legal Apprentice, jointly developed by Vern Walker and Legal Apprentice, Inc[6].
Teaching legal argumentation provides an example of an area where the practical utility of the system is not compromised by having only a limited knowledge base. When Ashley moved on from HYPO, he began work on the CATO system (developed with Vincent Aleven) which uses many of the ideas, and the domain, of HYPO, but which is targeted at teaching law students how to argue with precedents.

[6] Demos can be found at http://people.hofstra.edu/faculty/vern_r_walker/LegalReasoning.html.

Now the exercises presented to the students could be designed with the cases represented in the system in mind, and focussed on deploying cases already available. Even so the case base used in CATO is a considerable extension of that used in HYPO: although the knowledge base need not be complete with respect to the domain, it must still be substantial. This system has been used in practice with actual law students, and was subjected to a detailed empirical evaluation with respect to its effectiveness, with encouraging results, providing evidence that a complete knowledge base is not essential for this task.

We have given some examples above of argumentation techniques which are, or are on the point of, being used in practical systems. Success, however, requires more than that such systems are possible: they must also be acceptable to the user. It is worth noting that the successful introduction of expert systems techniques in systems such as those developed by Softlaw, was not as a stand-alone system, but as integrated into a system that was able to address all the aspects of the user's task, incorporating such mundane things as word processing and e-mail as well as the deductive application of regulations. Providers of argumentation systems should similarly consider how their tools can be integrated into the working environment of their intended users. (For example, Room 5 was integrated with features to search legal precedent databases).

Another barrier to acceptance of these tools may be that they are often based on normative views of what legal reasoning should be. As such they will prove acceptable only in so far as users are able and willing to relate these normative models to their tasks as they see them, or can be persuaded that the normative model is superior to their current practice. As an example, consider a system for structuring evidential arguments, such as MarshalPlan. It has been argued that if judges would systematically make their generalisations that connect the evidence to their conclusion explicit, this would improve the quality of their decisions, because it would enable critical testing of these generalisations (Wagenaar et al., 1993). Although in civil law systems judges are required to justify their decisions on matters of fact, these requirements are rather weak, and judges almost never make the generalisations that may underlie their decisions explicit. A system that required them to do so would therefore be accepted only if the judges can be convinced or forced to change their practice.

10 Concluding Remarks

In this chapter we have tried to show that Artificial Intelligence has more to offer the lawyer than mechanical deduction. It is universally recognised that legal reasoning requires something more sophisticated than this, and we have described a variety of approaches that attempt to provide this additional sophistication. Despite their variety, we feel that they all have in common the recognition of the need to address the dialectical and contextual elements of legal reasoning. By addressing argumentation we recognise the need to replace things that are lost when we abstract from an argument to a deductive proof, and are forced to take seriously the procedural and contextual elements that come with dialectics.

Addressing these issues is currently an area of active research. We have considered the prospects for practical implementation, and identified some of the obstacles that need to be overcome, most notably the knowledge acquisition bottleneck. Nevertheless we believe that the techniques are of more than purely theoretical interest and, provided support tools are developed with a clear understanding of their limitations, areas where they can provide highly effective support can be identified. Currently we see systems to support the structuring of arguments, on-line dispute resolution and teaching of argumentation to be the most promising for early exploitation.

References

Aleven, V. (1997), *Teaching Case Based Argumentation Through an Example and Models*. PhD Thesis, The University of Pittsburgh.

Ashley, K.D., (1990), *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*. Cambridge, MA: MIT Press.

Bench-Capon, T.J.M. Geldard., T. & Leng, P.H. (2000), A method for the computational modelling of dialectical argument with dialogue games. *Artificial Intelligence and Law* 8: 233-254.

Bench-Capon, T.J.M. & Sartor, G. (2001), A quantitative approach to theory coherence. In H.B. Verheij et al. (eds.), *Legal Knowledge and Information Systems. Jurix 2001: The Fourteenth Annual Conference*, 53-62. Amsterdam: IOS Press.

Bench-Capon. T.J.M., (2002). Representation of case law as an argumentation framework. In T.J.M. Bench-Capon, A. Daskalopulu & R. Winkels (eds), *Legal Knowledge and Information Systems. JURIX-2002: The Fifteenth Annual Conference*. 103-112

Bench-Capon, T.J.M., (2003). Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation* 13(3): 429-448.

Bench-Capon, T.J.M. & Sartor, G. (2003), A model of legal reasoning with cases incorporating theories and values. *Artificial Intelligence* 150: 97-143.

Bench-Capon, T.J.M. & Staniford, G. (1995), PLAID - Proactive legal assistance. *Proceedings of the Fifth International Conference on AI and Law*. ACM Press: New York, 81-88.

Berman, D.M., and Hafner, C.L., (1993). Representing teleological structure in case-based legal reasoning: the missing link, *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*. ACM Press: New York, 50-59.

Bex, F.J., Prakken, H., Reed, C. & Walton, D.N. (2003), Towards a formal account of reasoning about evidence: argumentation schemes and generalisations. *Artificial Intelligence and Law* 11: 125-165.

Branting, L.K. (2000) *Reasoning with Rules and Precedents: A Computational Model of Legal Analysis*. Kluwer Academic Publishers, Dordrecht.

Branting, L.K. (2003), An agenda for empirical research in AI and Law. *Workshop Notes of the ICAIL-2003 Workshop on Evaluation of Legal Reasoning and Problem-Solving Systems*, Edinburgh, UK, 28-35.

Dung, P.M., (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial Intelligence*, 77: 321-57.

Gardner, A. von der Lieth, (1987), *An Artificial Intelligence Approach to Legal Reasoning*. Cambridge, MA: MIT Press.

Gordon, T.F. (1991), An abductive theory of legal issues. *International Journal of Man-Machine Studies* 35: 95-118.

Gordon, T.F. (1995), *The Pleadings Game. An Artificial Intelligence Model of Procedural Justice*. Dordrecht/Boston/London: Kluwer Academic Publishers.

Greenwood, K. Bench-Capon, T.J.M. & McBurney, P. (2003), Towards a computational account of persuasion in law. *Proceedings of the Ninth International Conference of Artificial Intelligence and Law*. New York: ACM Press, 22-31.

Hage, J.C. (1996), A theory of legal reasoning and a logic to match. *Artificial Intelligence and Law* 4: 199-273.

Hage, J.C., (2001), Formalising Legal Coherence. In *Proceedings of the Eighth International Conference on AI and Law*, ACM Press: New York. pp 22-31.

Kowalski, R.A & Toni, F. (1996), Abstract argumentation. *Artificial Intelligence and Law* 4: 275-296.

Lodder, A.R. (1999), *DiaLaw – On Legal Justification and Dialogical Models of Argumentation*. Dordrecht/Boston/London: Kluwer Academic Publishers.

Lodder, A.R. (2001), A simple model to structure the information of parties in online ADR. *Proceedings of the Eighth International Conference of Artificial Intelligence and Law*. New York: ACM Press, 2001, 233-234.

Loui, R.P. & Norman, J. (1995), Rationales and argument moves. *Artificial Intelligence and Law* 3: 159-189.

Loui, R.P., Norman, J., Alpeter, J., Pinkard, D., Craven, D., Linsday, J., & Foltz, M. (1997), Progress on Room 5: a testbed for public interactive semi-formal legal argumentation. *Proceedings of the Sixth International Conference of Artificial Intelligence and Law*. New York: ACM Press, 1997, 207-214.

McCarty, L.T. (1977), Reflections on TAXMAN: An experiment in artificial intelligence and legal reasoning. *Harvard Law Review* 90: 89-116.

McCarty, L.T. & Sridharan, N.S. (1981), The representation of an evolving system of legal concepts II. Prototypes and deformations. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, 246-253.

McCarty, L.T. (1995). An Implementation of Eisner v Macomber. In *Proceedings of the Fifth International Conference on AI and Law*, 276-286. ACM Press: New York.

MacCormick, D.N. &. Summers, R. S. (eds.) 1997. *Interpreting Precedents: A Comparative Study* Dartmouth Publishing: Aldershot, U.K

Mommers, L., (2002). *Applied Legal Epistemology*. PhD Thesis, University of Leiden.

Peczenik, A., (1996). Jumps in Logic and the Law. *Artificial Intelligence and Law*, Vol 4:3-4, pp 297-329.

Prakken, H. (1993), A logical framework for modelling legal argument. *Proceedings of the Fourth International Conference of Artificial Intelligence and Law*. New York: ACM Press, 1993, 1-10.

Prakken, H. (2001), Modelling reasoning about evidence in legal procedure. *Proceedings of the Eighth International Conference of Artificial Intelligence and Law*. New York: ACM Press, 2001, 119-128.

Prakken, H. (2002), An exercise in formalising teleological case-based reasoning. *Artificial Intelligence and Law* 10: 113-133.

Prakken, H. & Sartor, G. (1996), A dialectical model of assessing conflicting arguments in legal reasoning. *Artificial Intelligence and Law* 4: 331-368.

Prakken, H. & Sartor, G. (2002), The role of logic in computational models of legal argument: a critical survey. In A. Kakas and F. Sadri (eds.), *Computational Logic: Logic Programming and Beyond. Essays In Honour of Robert A. Kowalski*, Part II. Springer Lecture Notes in Computer Science 2048, Berlin, 342-380.

Reed, C.A. & Rowe, G.W.A (2001), Araucaria: Software for Puzzles in Argument Diagramming and XML. Department of Applied Computing, University of Dundee Technical Report. [The Araucaria software can be downloaded from www.computing.dundee.ac.uk/staff/creed/araucaria/]

Schum, D.A. & Tillers, P. (1991), Marshalling evidence for adversary litigation. *Cardozo Law Review* 13: 657-704.

Skalak, D.B. & Rissland, E.L. (1992), Arguments and cases: an inevitable intertwining. *Artificial Intelligence and Law* 1: 3-44.

Toulmin, S.E. (1958), *The Uses of Arguments*. Cambridge University Press.

Verheij, H.B. (1996), An integrated view on rules and principles. In R.W. van Kralingen et al., *Legal Knowledge Based Systems. Foundations of Legal Knowledge Systems (Proceedings of JURIX-1996)*, 25-38. Tilburg: Tilburg University Press.

Verheij, H.B. (1999), Automated argument assistance for lawyers. *Proceedings of the Seventh International Conference of Artificial Intelligence and Law*. New York: ACM Press, 1999, 43-52.

Vey Mestdag, C.N.J. de (1998), Legal expert systems. Experts or expedients? The representation of legal knowledge in an expert system for environmental permit law. In C. Ciampi & E. Marinai (eds.), *The Law in the Information Society, Conference Proceedings on CD-Rom*, Firenze, 2-5 December 1998 (8 pages).

Wagenaar, W.A., Koppen, P.J. van & Crombag, H. (1993), *Anchored Narratives. The Psychology of Criminal Evidence*. NewYork: St. Martin's Press

Walton, D.N. (1996), *Argumentation Schemes for Presumptive Reasoning*. Mahwah, N. J.: Erlbaum.

Walton, D.N. & Krabbe, E.C.W. (1995), *Commitment in Dialogue. Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany, NY.

Wigmore, J.H. (1931), *The Principles of Judicial Proof*, 2nd ed., Boston: Little, Brown and Company.

Zeleznikow, J., & Stranieri, A. (1995). The Split-Up system. *Proceedings of the Fifth International Conference on Artificial Intelligence and Law*. ACM Press: New York, 185-195.