Modelling Reasoning about Evidence in Legal Procedure

Henry Prakken
Institute of Information and Computing Sciences, Utrecht University
P.O. Box 80089, 3508 TB Utrecht, The Netherlands
henry@cs.uu.nl, http://www.cs.uu.nl/staff/henry.html

ABSTRACT

This article investigates the modelling of reasoning about evidence in legal procedure. To this end, a dialogue game model of the relevant parts of Dutch civil procedure is developed with three players: two adversaries and a judge. The model aims to be both legally realistic and technically well-founded. Legally, the main achievement is a more realistic account of the judge's role in legal procedures than that provided by current models. Technically, the model aims to preserve the features of an earlier-developed framework for two-player argumentative dialogue systems.

1. INTRODUCTION

The procedural aspects of legal argument have become an important topic of AI & Law research. Several models have been developed in the form of dialogue games, regulating the use of argumentative speech acts and defining the outcome of a dispute [3, 5, 10, 1, 8, 14, 2]; cf. also [16]. Although these contributions have been very valuable, further research is needed.

Firstly, in the current models the judge's role, if modelled at all, is limited to the simple activity of determining the truth of the parties' claims. Yet in actual legal procedures judges have a much more elaborate role. For instance, in Dutch civil procedure judges allocate the burden of proof, determine whether grounds sufficiently support a claim, complete the parties' arguments with legal and common knowledge, decide about admissibility of evidence, and assess the evidence. The main aim of the present paper is to show how procedural models of legal argument can give more realistic accounts of the judge's role in legal disputes.

A second limitation of current models is that their formal foundations are rather ad-hoc. Although most current systems are carefully designed, it is often hard to see the underlying design principles used and design choices made; this in turn makes it hard to compare the various models and to investigate their formal properties. For these reasons I have in [13, 15] begun to develop a general framework for argumentative dialogue systems, and a secondary aim of the present paper is to show how procedural models of legal argument can be based on this framework.

I will attempt to realise these aims by modelling an example procedure, viz. a somewhat simplified version of the Dutch law of evidence in civil procedure. Ultimately, the present research should lead to automated procedural-support tools for officials deciding legal disputes. However, designing such a support tool is not an aim of the present paper.

This paper is organised as follows. Section 2 surveys the legal-procedural phenomena to be modelled. Section 3 then discusses the formal tools used in this paper, after which Section 4 discusses how the main tool, the framework of [13], must be extended to suit the present aims. Then the formal model is presented in Section 5 and illustrated with an example in Section 6, after which Section 7 concludes.

2. LEGAL ANALYSIS

2.1 Dutch civil procedure as far as modelled

I now briefly review Dutch civil procedure as far as relevant for present purposes (profiting from a discussion of this procedure from an AI & Law perspective by [7]).

A procedure is divided into a 'pleadings' phase, where the adversaries plea their case before the judge and provide evidence when necessary, and a 'decision phase', where the judge withdraws to decide the case. The pleadings phase is separated into a written and an (optional) oral part. In the written part the parties exchange at least two documents. The first is plaintiff's *Statement of Claim*, which has to contain plaintiff's claim plus his grounds for the claim. These grounds may be purely factual: plaintiff may leave out the legal 'warrant' connecting grounds and claim, as may both parties in all their other arguments. Also, parties do not need to explicitly state common-sense knowledge, and if they state such knowledge, they don't need to prove it. However, the judge decides what is common-sense knowledge. Defendant replies with her *Defence*, which has to contain all of defendant's challenges against plaintiff's claim and grounds. These challenges may also concern issues of procedure, so that the procedural legality of a move can itself become the subject of dispute. The adversaries may then exchange further documents, on which' content the procedural rules state no conditions. Each party may also ask to provide oral pleading. During the pleadings phase both parties may offer to provide evidence for their claims, plaintiff may change his initial claim, and the judge assigns the burden of proof to a party whenever appropriate, after which that party must

ICAIL-2001 St. Louis, Missouri USA

provide evidence. After the pleadings phase has ended, the judge gives his/her verdict, bound by the following rules.

An important principle of Dutch civil procedure is that the judge is passive. For instance, the judge must accept uncontested claims of the adversaries, and s/he must evaluate the evidence and give the verdict on the basis of the facts adduced by the parties (with the exceptions of generally known facts and legal rules).

As for allocating the burden of proof, the general rule is that the parties bear the burden of proving their claims. However, the judge may decide otherwise on the basis of special statutory provisions or on grounds of reasonableness. Among other things, this means that the burden of proof can be distributed over the parties, and that making a claim does not automatically create a burden to prove it; cf. [7, 14].

When assessing the evidence provided by the parties, the judge is bound by several rules. *Conclusive evidence* must be accepted in the absence of counterevidence, but the other party is allowed to provide such counterevidence. For instance, an 'avidavit', i.e., an official document containing statements by a legal official, is conclusive evidence that its content is true. *Incontrovertible evidence* must always be accepted by the judge and cannot be attacked by counterevidence. For instance, a so-called 'decisive oath' of one of the parties is incontrovertible evidence that its content is true. Finally, the law states rules for admissibility of evidence, such as the rule that a witness testimony may count as evidence only for what the witness him or herself observed (i.e., so-called 'de auditu' testimonies are inadmissible). Within these bounds, the judge is free to assess the evidence.

Besides these *legal* rules on civil procedure, the parties are, as always, also bound by *rational* rules of reasoning and discourse (as studied by, for example, philosophers, logicians and argumentation theorists). For instance, the parties should correctly apply the rules of logic (whether deductive or defeasible), and they should respect certain rational conventions for discourse. In my opinion, a main task of AI & Law models of legal procedure is to explain how these rational rules and conventions can be integrated with specific legal procedures. The explanation offered by the present paper is a clear separation of the actual dispute as it takes place during the pleadings phase from a rational reconstruction of this dispute made by the judge during the decision phase. In this paper only the second phase will be modelled.

2.2 Implications for the model

2.2.1 Actual pleadings vs. rational reconstruction

A key element of the present approach is to model the decision phase of civil procedure as a rational reconstruction of the pleadings phase. In the pleadings phase the judge is a participant in the dispute (for instance, when assigning the burden of proof). However, in the decision phase (where the adversaries are not involved), the judge reconstructs the dispute of the pleadings phase into a dialectical structure from which s/he can 'compute' the decision. S/he will identify the argumentative speech acts made by the parties (including the judge), whether they were procedurally correct, and how they logically and dialectically relate to each other. The judge will also complete the adversaries' arguments when necessary, and will decide whether the advanced arguments are internally strong enough. Finally, the judge will adjudicate between conflicting arguments.

Although the adversaries are not involved in the decision phase, this phase will nevertheless be modelled as a dialogue between them and the judge. This is since the judges' activity of rationally reconstructing the pleadings phase can be imagined as the attempt to extract argumentative speech acts from what has been said (in writing or orally) during pleadings, and building a rational dialogue with them according to the rules of a rational dialogue game.

How does this approach account for the relation between the legal rules of civil procedure and the rational rules of argumentative discourse? Firstly, it respects that the law governing the pleadings phase says nothing on, for instance, respecting commitments, or when a move is dialectically relevant. These are issues to be modelled in the decision phase, as rules for rational discourse. Secondly, the legal rules governing the pleadings phase are not built into the rules for the decision-phase dialogue. For instance, rational admissibility of defendant's attacks on plaintiff's main claim is independent of whether defendant had stated all her attacks in her *Defence*. As long as allowed by the rules of rational discourse, the judge can input such an attack at any point in his/her rational reconstruction. If defendant had not made the attack in her *Defence*, the judge can subsequently decide the attack to be 'illegal'.

2.2.2 Aspects to be modelled

I now list in more detail the activities to be modelled. For an example of a dialogue to be modelled, the reader is invited to look ahead to Section 6.

The parties' acts:

- Making, challenging, retracting and conceding a claim.
- Stating, conceding and retracting arguments.
- Claiming that a move is procedurally illegal, and arguing about such claims.

The judges' acts:

- Deciding about legality of a move.
- Deciding about the burden of proof.
- Deciding whether a party has met a proof burden. This involves deciding about the following issues.
  - Whether to complete an argument with legal or commonsense knowledge;
  - Whether an argument is able to support its conclusion even in the absence of counterarguments (internal validity);
  - Whether counterarguments are allowed;
  - Whether the argument survives competition with its counterarguments (dialectical validity).

3. AN OVERVIEW OF THE FORMAL TOOLS

3.1 Formal dialectics

Procedural AI & Law models have largely been based on a branch of argumentation theory and philosophical logic called 'formal dialectics' [11, 19]. Here argumentative discourse is modelled as dialogue games, regulating the use of argumentative speech acts and determining the result of a dialogue on the basis of what the players have committed themselves to and what logically follows from this.

Walton & Krabbe [19] identify four kinds of rules in dialogue systems: *locution rules* (what moves are possible); *structural rules* (when moves are admissible); *commitment rules* (the effects of moves on the players' commitments); and *termination rules* (when dialogues terminates and with what outcome). One important contribution of AI & Law to formal dialectics is the possibility of counterargument. A formal underpinning for this contribution is provided by a recent development in AI.

3.2 Games for defeasible argumentation

AI research on defeasible reasoning has resulted in *dialectical argument games*, e.g. [9, 18, 17, 6]. Such games model defeasible reasoning as a dispute between a proponent and opponent of an argument. Each player must attack the other player's arguments with a counterargument of sufficient strength. Usually, a player wins if the other player has run out of moves. The initial argument is justified if the proponent has a winning strategy. A natural idea in argument games is dialectical asymmetry. Since proponent wants to prove his initial argument justified, his attacks must be stronger than their targets (strict defeat) while, since opponent only wants to prevent proponent from meeting his aim, her attacks may be just as strong as their targets (defeat).

Clearly, these argument games fit well with formal dialectics. However, for present purposes these games, being logical proof theories, have one important limitation: they assume a given and fixed information base from which the arguments can be constructed. By contrast, in legal disputes the information is supplied dynamically during a dispute. Accordingly, in [15] I showed how argument games can be 'dynamified' in this sense (cf. also [6]).

Another limitation of the argument games is that they do not fully capture shifts of the burden of proof from one party to the other. Suppose that plaintiff fulfills his burden of proving that agreement on a sales contract was reached, and that defendant then argues for an exception since she was insane at the time of the negotiations. Legally, defendant has the burden of proving this exception, which logically means that the dialectical asymmetry is reversed: for plaintiff it suffices to cast doubt on defendant's evidence, i.e., it suffices to simply defeat her argument. In [14] I showed how argument games can be modified to cope with this phenomenon. Essentially, during an argument game the dialectical asymmetry between the parties can be reversed, depending on whether their argument is moved to meet a proof burden, or to prevent the meeting of a proof burden. For present purposes this means that each time a counterargument is moved, the judge must first decide about the burden of proof, before the required defeating force (defeat or strict defeat) of the attacker can be determined.

The contributions of [15, 14] still abstract from the speech act aspects of argumentation. These aspects are addressed in [13], in which a framework is presented for embedding argument games in dialogue systems. Since below I will build on that framework, I now discuss it in some detail.

3.3 Combining argument games with dialogue systems

The framework of [13] is intended for two-player argumentative dialogue systems. It adds to the elements of [19] a fifth and sixth element, viz. *proof burden rules* and an *underlying dialectical argument game*. The framework is designed to allow for maximal freedom for variations on all six points, within the bounds of some general principles.

The framework allows for various sets of **speech acts**. Each act A has a set $c(A)$ of *act components*, which are the possible targets of replies to the act. Most acts can be replied-to only in their entirety, so that they have just one component, viz. the act itself. For example, a challenge of a claim always attacks the entire claim. However, an argue move can be replied to at different places: its premises can be challenged, its inference rule can be undercut, and its conclusion can be rebutted. Below I shall be ambiguous between 'act' and 'act component' if there is no danger for confusion.

Every kind of speech act (except the move that initiates a dialogue) is defined as either an *attacking reply* or a *surrendering reply* to another speech act (component). For instance, in many systems *claim* φ can be attacked by *why* φ and surrendered by *concede* φ. In line with the intended meaning of surrendering, a surrendering act has no reply.

The propositions used in the content of speech acts come from the logical language of an underlying **dynamic argument game**, while the arguments moved in the dialogue are valid arguments in this game.

Moves have a player, a speech act, and (except the initial move) a target. The latter means that every non-initial dialogue move explicitly reacts to one preceding move in the dialogue. This condition induces a convenient tree structure on each dialogue. For any move M, $Player(M)$ is the player of M, $Act(M)$ denotes its speech act, and $Target(M)$ denotes the act component of the move to which M replies.

The dialogical **effects** of speech acts are captured in terms of the players' **commitments**, which are a set of propositions that s/he has somehow become committed to during a dialogue and should respect during the rest of the dialogue. For instance, in most dialogue systems a *claim* φ move commits the mover to φ, so that, for instance, a later *why* φ move of the same player is not allowed.

A crucial notion of the framework is **admissibility** of moves.[1] At each point of a dialogue, a system should define which moves can be made at that point. The framework leaves the content of admissibility rules free, except for some necessary conditions. Most importantly,

- Each moved speech act must be defined as a reply to the move's target.
- Each argue move is admissible in the underlying argument game.
- Each move must be 'structurally' relevant, in a sense to be defined below.

As for the 'underlying argument game', the idea, taken from [13], is that during the dispute the players implicitly build an argument tree, containing the arguments exchanged. This tree is used to determine admissibility of argument moves, viz. with the underlying argument game; cf. [15]. Ideally, the tree also determines the outcome, since (ideally) if the proponent wins a dispute, the tree is a dialectical proof of the main claim on the basis of what has

[1] In [13] this was called 'move legality'; the name change is to prevent confusion with a move's legal correctness.

been said during the dispute. (Actually, this argument tree 'associated' with a dispute is not just one tree but a forest of trees, since the proponent may give several alternative arguments for his main claim.) To make this idea work, a function must exist that extracts the associated argument trees from a dialogue. [13] defines such a function.

The notion of structural relevance is essential to the framework, and therefore I will discuss it in some detail. Relevance is defined in terms of another essential notion, viz. that of the *dialogical status* of a move. This status (either 'in' or 'out') captures whether a move is currently adequately defended by its mover or not.

I first discuss a move's dialogical status. Since all attacks are directed to an act component, the status of a move is defined in terms of that of its act components. There are two ways in which an act component can be in: the other player can have 'conceded' it, or all attacks of the other player have been successfully replied to (where success is determined recursively). An act component is *conceded* if it has a surrendering reply.

As for dialogical status, leaves of a dialogue tree are in iff they are attackers, and then the status of moves is recursively defined upwards:

Definition 3.1. *(Dialogical status of moves and components) Moves and move components are either* in *or* out.

1. *A move is in iff all its act components are in.*
2. *A move component C is* in *iff*
   (a) *C is conceded; or else*
   (b) *all attacking moves that reply to C are out.*

Now relevance of a move is defined as follows.

Definition 3.2. *(Relevance.) A target is* relevant *iff any attacking reply to it changes the status of the main claim. A move is* relevant *iff it replies to a relevant target.*

Note that a reply to a conceded move is never relevant.

To illustrate these definitions, consider figure 1 (with only one-component and attacking moves). The dispute tree on the left is the situation after P_4. The tree in the middle shows the dialogical status of the moves when O has continued after P_4 with O_4, replying to P_3: this move does not affect the status of P_1, so O_4 is irrelevant. Finally, the tree on the right shows the situation where O has instead continued after P_4 with O'_4, replying to P_4: then the status of P_1 has changed, so O'_4 is relevant.

The notion of relevance serves several purposes. First, it provides a lower bound on the freedom to move alternatives to one's earlier moves. Secondly, it regulates whether a postponed reply still makes sense. And, finally, it enables an elegant notion of **turntaking**: since each move must be relevant, a turn shifts to the other player as soon as a player has succeeded in changing the status of the main claim his way.[2] So proponent is to move as long as the main claim is out, and opponent is to move as long as it is in. In other words, each player first moves zero or more surrenders, and then moves zero or one attacker: if no attackers are moved, this is because the player has no admissible moves, so that the dialogue has terminated.

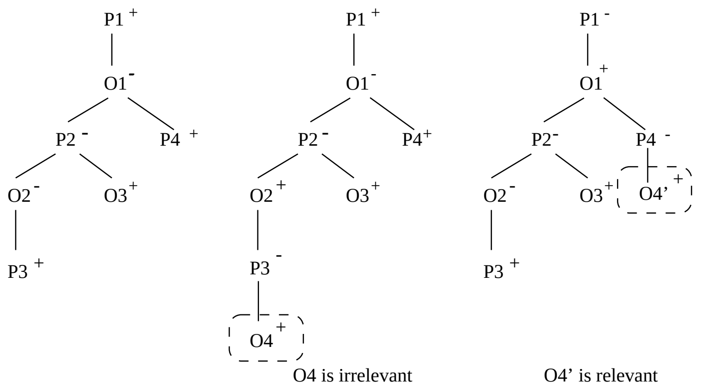

Figure 1: Dialogical status of moves.

In fact, **Termination** is defined implicitly, as the situation where a player has to move but has no admissible moves. Finally, the framework assumes a notion of **winning** a dispute; it is constrained by the condition that at termination the main claim is labelled the winner's way.

4. CONCEPTUAL SPECIFICATION

I now turn to the modelling of our example procedure. Our challenge is twofold. A legal challenge is to model the procedure accurately, while a technical challenge is to preserve the structure of the above general framework. In meeting the latter challenge the key issue is how to incorporate a third party (the judge) while preserving the useful roles of the notions of dialogical status and relevance.

Ideally, I would now first extend the general framework with a third party and then instantiate it for our procedure. However, this goes beyond the limits of the present paper, so I will confine myself to defining the particular dialogue game, leaving the generalisation to future research.

I first sketch how a dispute according to our dialogue game generally evolves (recall that it is a rational reconstruction of the actual events during pleading). Each dispute starts with a main claim of the plaintiff. When challenged by the defendant, the adversaries exchange arguments and counterarguments, and can challenge their premises, as well as legality of each other's moves. Sometimes, disagreements are resolved by conceding or retracting claims and/or arguments. During the dispute the adversaries implicitly build an argument tree, containing the arguments exchanged and determining admissibility of argument moves.

As for the judge, after each move of the adversaries, s/he decides on legality of the move, if necessary assigns the burden of proof, and, when an argument is moved, decides on its internal and dialectical strength. When the adversaries have terminated their dispute, the judge completes it with new arguments when possible. When these moves have also run out, the judge determines the winner of the dispute by looking at the dialogical status of plaintiff's main claim.

Let us now look at the elements in more detail.

4.1 The players

As for the players, proponent and opponent will be renamed as *plaintiff* (π) and *defendant* (δ). In addition, there

[2] This turntaking rule is taken from [9], while the notion of dialogical status generalises an idea of [4]. The innovation of [13] is to extend these ideas to other speech acts than moving arguments.

now is a third party, the *judge* (j). Plaintiff and defendant will be called the *adversaries*. If a is an adversary, then $\overline{a}$ denotes the other adversary. So $\overline{\pi} = \delta$ and $\overline{\delta} = \pi$.

4.2 The underlying argument game

As for the underlying argument game, I make as few assumptions as possible.

Firstly, the game must allow for switches of the dialectical asymmetry caused by shifts of the burden of proof; cf. [14]. This means that the players (including the judge) can move arguments in one of two 'dialectical roles', proponent or opponent, and that the players can have different roles at different stages of a dispute.

Also, the game must allow for reasoning about the strength of arguments. Recall from [17] that such reasoning takes the following form. Opponent's arguments need not state any priorities to defeat their target. Proponent either attacks opponent's argument with an argument that includes the priorities making it strictly defeat opponent's argument, or moves a 'neutralising' priority argument, i.e., a priority argument according to which opponent's argument does not defeat (so is strictly defeated by) its target. Priority arguments can themselves also be attacked.

As for the arguments admitted by the argument game, I assume that they can be divided into *strict* (= deductive) and *defeasible* arguments, and that only the latter category is subject to defeat. Further, the class of defeasible arguments may contain, for instance, analogical, inductive or abductive arguments. I also assume that the game allows for either *undercutting* arguments [12] or *assumption attacks* [17]. Note that since the adversaries may state incomplete arguments by leaving out common-sense and legal information, their arguments need not be admissible in the underlying argument game; it is the judge's task to decide whether they can be completed to admissible arguments; if so, it is the completed versions of the arguments that are entered into the argument game tree associated with the dispute.

Finally, as for the logical language, this can but need not be a logical language in the usual sense. In fact, in practical applications it will often be better to allow for structured and linked pieces of natural language, such as Toulmin's argument schemes (cf. [1]), or the pictorial languages of [4]'s ZENO system or [10]'s ROOM 5 system.

4.3 The speech acts

The dialogue game will contain the usual speech acts for making, challenging, retracting and conceding claims, and for moving arguments. The game will also allow for conceding and retracting arguments and for questioning the legality of moves. The move that concedes an argument is taken from [3]. Its effect is to give up the possibility of counterargument. The *illegal* move is adapted from [8] and [2].

Next a third category of speech acts must be introduced besides attacks and surrenders, viz. *determiners*. Such acts, to be played only by the judge, have no replies but are yet not surrenders; instead they decide a certain issue. The effect of determiners on the dialogical status of their target must be specified for each case. Our model will contain two determiners: allocations of the burden of proof, and internal validations of arguments.

4.4 Commitment rules

In the literature on formal dialectics the issues of consistency and logical closure of the commitments are much-discussed. However, in legal applications these issues seem less important than the 'legal' requirements discussed in Section 2; therefore, I will choose simple treatments of the former aspects, in order to focus on the latter issues. Firstly, although the commitments are not defined to be logically closed, several admissibility rules will look at what classically follows from them. Secondly, (and an illustration of the first point) making one's commitments classically inconsistent will not be admissible.

The judge can also incur commitments, for instance, when completing or stating an argument. Clearly, the judge must respect his/her commitments in the same way as the other players, on penalty of being dialogically incoherent. Moreover, the adversaries must respect those commitments of the judge that are common sense or legal knowledge.

As for the effects of speech acts on the speaker's commitments, the dialogue game will contain the usual rules that making or conceding a claim commits to the claim and that moving arguments commits to the premises. Furthermore, in line with [19], moving and conceding an argument also commit to the material implication *premises* $\rightarrow$ *conclusion*. Thus moving an argument implicitly creates a commitment to the conclusion. However, when conceding an argument this is different, since that only means that the connection between premises and conclusion is conceded, so it leaves room for challenging the premises. As for retractions of propositions, it will simply be assumed that every retraction ends commitment. Problems with implied commitments will be avoided by making any 'unsuccessful' retraction inadmissible (i.e., when the retracted commitment is still implied by other commitments). Finally, when retracting an argument, the material implication *premises* $\rightarrow$ *conclusion* is removed from the commitments.

4.5 The proof burden stores

Since the judge can assign the burden of proof to the players, the players not only have a commitment store but also a *proof burden store*. Roughly speaking, commitments are about being coherent while proof burdens are about being right. In the present model, the judge can also assign the burden of proof to him/herself; this is a technical way to capture the judge's decision that a proposition needs no proof.

Having the burden of proof is independent of being committed. Assume, for instance, that plaintiff claims φ, defendant then challenges φ after which the judge assigns the burden of proving $\neg\varphi$ to defendant. Then plaintiff is committed to φ without having the burden of proof of φ, while defendant has the burden of proving $\neg\varphi$ without (yet) being committed to $\neg\varphi$.

4.6 Turntaking

Recall that the aim is to let turntaking be regulated by relevance. Now the idea is that after each move by an adversary the turn shifts to the judge, who then decides on several issues, viz. legality of the move, the burden of proof (if necessary), and internal and dialectical strength of a moved argument. After the judge's turn has been completed, the turn shifts to the adversary who then has the status of the main claim against him. The judge's turn can be completed in two ways. The first is as soon as s/he has changed the dialogical status of the main claim, and the second is when the judge has made all relevant decisions without having effected such a change. In fact, the latter is the intended result of the last-moving adversary, since otherwise something has turned out to be wrong with his or her move.

Actually, the above holds only as long as there are new adversaries' moves to which the judge's procedure can be applied. When no such moves are left, the judge can move relevant additional arguments, as further explained below in 4.9.

4.7 A procedure for judge's turns

The judge's turns can simply be defined as a fixed procedure (except in the completion phase, as explained in 4.9). Consider a move M moved by a:

1. M is first checked on *legality*. If illegal, the turn switches back to a. If legal, then if M is not a *why, argue* or *illegal* move, the turn switches to $\overline{a}$, else:
2. If M is a *why* move, the judge decides about the *burden of proof*, after which the turn switches to the adversary who gets it.
3. Else the judge assesses *internal validity* of the argument moved in M. If invalid, the turn switches back to a. Else:
4. The judge decides whether the evidence is *incontrovertible*. If so, the turn switches to $\overline{a}$, else:
5. If the moved argument is not a counterargument, the turn switches to $\overline{a}$, else:
6. The judge assigns the *burden of proof*; and:
7. The judge assesses the argument's *dialectical validity*. If valid, the turn switches to $\overline{a}$, otherwise to a.

As noted in Section 2, several decisions in the procedure are governed by procedural law. However, this law is not captured by our dialogue game; it could be modelled, for instance, in a knowledge-based system supporting the judge when taking the various procedural decisions.

Let us now examine the details of this procedure. The task is to map each decision in the tree onto zero or more acts in the dialogue game.

4.7.1 Deciding on move legality

Deciding that a move is procedurally legal is done by moving no speech act, while deciding that a move is illegal is done with an attacking speech act $illegal(M, A)$, where M is the move judged illegal and A an argument why the move is illegal. This speech act is not a determiner but an attacker, so it can also be used by the adversaries. It can be replied to with the same replies as to *argue* A, so the legality of a move can also be disputed by the adversaries. If so, then the judge's decision on move legality takes the form of his/her decision of this dispute. The effects of an $illegal(M, A)$ move on the mover's *commitments* are the same as those of *argue* A. The effect on the *argument tree* associated with the dispute is that A is the root of a new tree. The effect on M's *dialogical status* is the same as with any attacker, viz. that M is made out. Finally, there are no special *admissibility* conditions for the *illegal* move. Note that this speech act does not question admissibility of a move in the decision phase, but legality in the pleadings phase.

4.7.2 Assigning the burden of proof

Assigning the burden of proof (after a *why* move or a counterargument) is done with a determining speech act $burden(\varphi, p)$, where p is the player to which the burden of proving φ is assigned. If p is the judge, this is taken to mean that φ needs no proof. The only case where the $burden(\varphi, p)$ speech act has effects on the speaker's *commitments* is when $p = j$; in that case, φ is added to judge's commitments.

A burden assignment sometimes has an effect on the associated *argument tree*. If the assignment followed a counterargument A, it has become possible to determine the required defeating force of A. Accordingly, A's counterpart in the associated argument tree will, if it has proponent role, be modified by adding priorities that make A strictly defeat its target. Note that this is not yet a decision to regard A dialectically valid, since the judge can still move an undercutting or priority counterargument.

The effect on the *dialogical status* of the target is as follows. Consider first a burden assignment $burden(\varphi, a)$ to one of the adversaries: if the assignment's target was a *why* φ move of $\overline{a}$, then that move stays in, while if the target was a *why* $\neg\varphi$ move of $\overline{a}$, it is made out. Further, if the target was an *argue* move, that move stays in. Consider next an assignment $burden(\varphi, j)$ to the judge. In fact, the judge has thus determined φ to be true, so any further challenge of φ should be made impossible. This can be done as follows. (Note that this method requires a refinement of the notion of a conceded move.) If the assignment's target was a *why* φ move M, then M is made out while M's target is made conceded (note that M's target either is a *claim* φ move or a premise component φ of an *argue* move). Conversely, if the assignment's target was a *why* $\neg\varphi$ move M, then M is made conceded, which effects that M's target (again either a *claim* $\neg\varphi$ move or a premise component $\neg\varphi$ of an *argue* move) is made out. Next, if the target was an $argue(S,\ so\ \varphi)$ act of move M, then M is made conceded, so M's target (an *argue* act) is made out, while, reversely, if the target was an $argue(S,\ so\ \neg\varphi)$ act of move M, then M is made out and M's target (an *argue* act) is made conceded.

Finally, proof burden allocations must satisfy some special *admissibility* conditions (cf. [14]). Most importantly, if one player must prove φ, no other player should have the burden of proving either $\neg\varphi$.

4.7.3 Deciding on internal validity of an argument

At first sight, it would seem that internally validating an argument can be expressed by doing nothing. However, we have to account for the possibility that a judge validates an argument by completing it with commonsense or legal knowledge. Accordingly, deciding that an argument is internally valid is expressed with a speech act $valid(A, A')$, which means that A is validated as A'. Here A' is an argument that extends A with zero or more premises. This move is only *admissible* if the completed argument A' is constructible in the underlying argument game. A $valid(A, A')$ move commits the judge to the premises added by A'. Its effect on the associated *argument tree* is that A is added to it as A'. Finally, its effect on the *dialogical status* of the target move is obvious: this move stays in.

Perhaps surprisingly, deciding that a move is internally invalid also starts with moving a *valid* move. However, the only purpose of this is to add the argument to the associated argument tree, after which it can be invalidated with an undercutting argument. So, in sum, judging an argument internally valid is expressed by first making a *valid move* and then proceeding to the issue of controvertibility, while judging an argument internally invalid is expressed by first making a *valid move* and then undercutting it, after which the dialogical status of the target becomes out.

4.7.4 Deciding whether evidence is incontrovertible

Incontrovertibility of evidence can be expressed as a special way of deciding the argument to be internally valid, viz. by extending it to a deductively valid argument with the material implication 'If premises then conclusion'. Any counterattack is then inadmissible by the rules of the underlying argument game.

4.7.5 Deciding on dialectical validity of an argument

As for modelling decisions on the dialectical validity of an argument, recall first that the adversaries are allowed to move any counterargument, as long as it attacks its target, and that the priorities required by the underlying argument game are, if not already stated by the adversaries, added as a result of assignments of the burden of proof. So when the judge decides on dialectical validity of an argument, the associated argument tree is already fully well-formed. Note also that the issue of dialectical validity only arises after a counterargument is moved.

Now the judge can dialectically validate an argument by doing nothing. If the argument thus validated was a counterargument, the judge thereby indirectly invalidates its target. The judge can directly dialectically invalidate an argument by moving an undercutting argument or a priority counterargument. In the latter case, if the argument was moved in proponent role, it can be invalidated with an argument attacking its priority part, while if it was moved in opponent role, it can be invalidated by moving a 'neutralising' priority argument.

4.8 Admissibility of moves

As for admissibility of moves, many special conditions were already discussed above; some further conditions will be discussed in the following section. In addition, our game inherits, of course, the necessary conditions from the general framework described above. Since Dutch civil procedure allows alternative replies to the same move, there seems no reason to sharpen the lower bound on alternatives provided by the relevance criterion of the general framework. So any alternative to an earlier move is admissible if it is relevant and satisfies all other admissibility criteria.

4.9 The completion phase

When all pleadings-phase moves of the players have been dealt with, the dispute is not yet terminated, since the information base created during the dispute might enable additional argue moves. For instance, a factual claim made by one of the adversaries might, when combined with some legal rule, enable a relevant counterargument that was not actually made during pleading. Accordingly, the present dialogue game allows the judge to complete the dispute with any such move. The decision-phase dispute terminates when these additional argument moves have run out; the dialogical status of the main claim then determines the outcome. Note that in an automated support tool the generation of arguments in this phase could be automated.

4.10 'Passivity' of the judge

Finally, how can the judge's passivity with respect to the facts be modelled? Firstly, any commitment incurred by the judge will be required to be either a generally known factual or legal truth, or something said by one of the adversaries. However, this does not yet capture the 'silence implies consent' principle that the judge must accept noncontested claims. Here a design problem arises, since whether a claim was contested in a dispute can only be determined after the dispute has terminated. So it makes no sense to forbid the judge to incur commitments that violate *as yet* noncontested claims of one of the adversaries.

For this reason, the dialogue game will not exclude that during the completion phase the judge discovers a counterargument against one of his own commitments based on those of an adversary. Now because of the just-stated rule, this contradicted commitment will be one of three types. Firstly, it can be a claim earlier made by one of the adversaries, in which case the counterargument reveals that this claim was not uncontested. Then the judge's argument can prevail only if the judge can state a priority argument based on what has been said in the dispute. Secondly, the contradicted commitment can be a legal rule, in which case the judge is free to determine which legal rule is correct or takes priority. Finally, the contradicted commitment can be a generally known fact or empirical generalisation. In this case, it can happen that the judge's judgement that a fact is generally known conflicts with a noncontested claim of one of the parties. In such a situation Dutch civil law provides no clear answer so that our dialogue game can leave the judge free to prefer one argument or the other. So in all three cases the judge's passivity is preserved.

However, it can also happen that the judge uses a commitment of an adversary that the adversary later retracts. To deal with this, the judge must be required to start the completion phase with retracting all such arguments.

5. THE FORMAL DIALOGUE GAME

I can now present the formal model. Because of space limitations only the most important definitions will be given in detail, and mostly semiformally; for the others the reader is referred to their above specification.

5.1 The speech acts

Table 1 specifies which speech acts can be made, of which type they are, and to which speech acts they are a possible reply. The only multi-component moves are the *argue* and *illegal* moves: each premise of a moved argument is a component, as well as the argument itself. As for notation, φ and φ_i are well-formed formulas, Φ is a set of well-formed formulas, A, B, C, D are arguments, p is a player and M is a move. In the rows concerning the *argue* act, C is an argument attacking A, and D is an argument attacked or neutralised by A. In the *burden* acts, $\varphi/\neg\varphi$ denotes that the burden is allocated of either φ or $\neg\varphi$.

5.2 Admissibility of moves

A move M is admissible in a dispute D iff it satisfies all conditions listed below. The first set of conditions is inherited from the framework discussed in Section 3.

1. M is D's first move iff M is a claim;

Table 1: Speech acts

Acts	Attacks	Surrenders	Determiners
$\mathit{claim}\ \varphi$	$\mathit{why}\ \varphi$ $\mathit{illegal}(M, B)$ $\mathit{argue}\ \Phi,\ \mathit{so}\ \neg\varphi$	$\mathit{concede}\ \varphi$	
$\mathit{why}\ \varphi$	$\mathit{argue}\ \Phi,\ \mathit{so}\ \varphi$ $\mathit{illegal}(M, B)$	$\mathit{retract}\ \varphi$	$\mathit{burden}(\varphi/\neg\varphi, p)$
$\mathit{concede}\ \varphi$	$\mathit{illegal}(M, B)$		
$\mathit{retract}\ \varphi$	$\mathit{illegal}(M, B)$		
$\mathit{argue}\ A$ $(A = \Phi,\ \mathit{so}\ \varphi)$	$\mathit{argue}\ C$ $\mathit{illegal}(M, B)$ $\mathit{why}\ \varphi_i\ (\varphi_i \in \Phi)$	$\mathit{concede}\ A$ $\mathit{retract}\ D$ $\mathit{concede}\ \varphi_i\ (\varphi_i \in \Phi)$	$\mathit{valid}(A, A')$ $\mathit{burden}(\varphi/\neg\varphi, p)$
$\mathit{concede}\ A$	$\mathit{illegal}(M, B)$		
$\mathit{retract}\ A$	$\mathit{illegal}(M, B)$		
$\mathit{burden}(\varphi, p)$			
$\mathit{valid}(A, A')$			
$\mathit{illegal}(M, A)$	as to *argue* A	as to *argue* A	as to *argue* A

2. M's target occurs in D;

3. M is different from earlier replies to its target;

4. M's act is a possible reply to its target (see table 1);

5. If M is an $\mathit{argue}(A)$ move, an $\mathit{illegal}(M', A)$ move, or a $\mathit{valid}(A', A)$ move, then moving A is admissible in the associated argument tree (cf. [15]);

6. M is relevant in D.

The next set of conditions is taken from a game of [13] ('speaker' denotes the player making the move). Conditions 7 and 8 are obvious 'possibility' conditions for concessions and retractions, condition 9 is [11]'s way to avoid circular arguments, while conditions 10 and 11 regulate coherence of one's moves with one' his commitments.

7. If M concedes φ, then speaker's commitments did not already classically imply φ;

8. If M retracts φ, then

   (a) speaker was committed to φ before retracting it; and

   (b) after retraction speaker's commitments do not classically imply φ.

   (Condition 8b allows successful retractions only. If, for instance, one's commitments contain p, q and $p \rightarrow q$, then retracting q is inadmissible. To make it admissible, first either p or $p \rightarrow q$ must be retracted. Thus the players are forced to explicitly indicate how an implied commitment is retracted.)

9. If $\mathit{Act}(M_i) = \mathit{argue}(\Phi,\ \mathit{so}\ \varphi)$, then φ is not under challenge by $\overline{p}$ in the dispute branch $M_1, \ldots M_i$ of D.

   (A proposition φ is under challenge if a move $\mathit{why}\ \varphi$ was made and the mover has not yet 'withdrawn' this challenge by backtracking with $\mathit{concede}\ \varphi$.)

10. If M's act is $\mathit{why}\ \varphi$, then speaker's commitments do not classically imply φ.

11. No commitment to φ is incurred by a move M of player p if either:

    (a) p's commitments classically imply $\neg\varphi$; or

    (b) φ is under challenge by p.

The remaining admissibility conditions are motivated by the present aims.

12. If speaker is the judge and the completion phase has not yet started, his moves follow the procedure specified above in Section 4.7.

13. If M's act is a determiner, then M's mover is the judge;

14. If M allocates a burden of proof, then:

    (a) if φ is already in the proof burdens of one player, no other player is assigned the burden of proving either φ or $\neg\varphi$;

    (b) if M replies to a $\mathit{why}\ \varphi$ move played by a, then $\mathit{Act}(M)$ is either $\mathit{burden}(\varphi, a)$, or $\mathit{burden}(\neg\varphi, \overline{a})$ or $\mathit{burden}(\varphi/\neg\varphi, j)$.

    (c) if M replies to an *argue* move $M' = (a,\ \mathit{argue}\ (S\ \mathit{so}\ \varphi),\ \mathit{Target})$, then

       i. $(S\ \mathit{so}\ \varphi)$ attacks an argument moved in *Target*; and

       ii. $\mathit{Act}(M)$ is either $\mathit{burden}(\varphi, a)$, or $\mathit{burden}(\neg\varphi, \overline{a})$ or $\mathit{burden}(\varphi/\neg\varphi, j)$.

15. If M is an *argue* move and its target is the main claim, then speaker has the burden of proving this claim false.

    (This condition implies that speaker has earlier attacked the main claim with a *why* move.)

16. If M is an *argue* move moved by the judge, then all of A's premises are either generally known facts, legal truths, or commitments of an adversary.

17. If φ is classically implied by the judge's commitments that are factual or legal truths, then:

    (a) No commitment to $\neg\varphi$ is incurred; and

    (b) no $\mathit{why}\ \varphi$ move is made.

(This condition ensures that no move is made that conflicts with a judge's earlier decision that something is a factual or legal truth.)

18. In the completion phase, the judge first retracts all his arguments that use an adversary's commitment that s/he later retracted, and then moves any *argue* A move such that

    (a) A is admissible in the associated argument tree.

    (b) A uses no adversary's commitment that s/he later retracted.

5.3 Termination and winning

When no moves of the adversaries are left to consider, the judge completes the dispute by in the way described in admissibility condition 18. The dispute terminates automatically when the judge has no completion moves left. As for winning, plaintiff wins if after termination the main claim is 'in', while defendant wins otherwise.

6. AN EXAMPLE

I now illustrate the dialogue game model with an example dispute. To save space, the presentation will be semiformal. Effects on commitments, proof burdens and associated argument tree are only shown when a change is effected.

The case is about whether a sales contract was created. Plaintiff argues it was and, when challenged by defendant, provides two witness testimonies as evidence. Defendant attacks this evidence, saying that there is no written agreement and that in this business oral agreements never occur. The judge decides that plaintiff's evidence is more convincing, after which defendant tries an alternative attack, claiming an exception to the rule that an offer and acceptance create a contract, on the basis of her insanity during the negotiations. When challenged by plaintiff, she produces a court's document declaring her insane. Plaintiff argues that it is inadmissible, on the basis of a procedural rule that the judge may ignore documents that were provided after the written pleadings phase. However, the judge declares it admissible since plaintiff's procedural interests were not damaged. Plaintiff then instead challenges authenticity of the avidavit. The judge has to assign the burden of proving the avidavit false to plaintiff, since the appearance of an avidavit counts as conclusive evidence that it is an avidavit. However, plaintiff fails to provide counterevidence, so the dispute ends with a win for defendant.

$\boldsymbol{\pi_1}$: *claim* (1) we have a valid contract (Target: none)
- (1) is added to plaintiff's commitments.
$\boldsymbol{\delta_2}$: *why* do we have a valid contract? (Target: π_1)
$\mathbf{j_3}$: $burden(1, \pi)$ (Target: δ_2)
(The judge assigns the burden of proving that there is a valid contract to plaintiff.)
$\boldsymbol{\pi_4}$: *argue* A_1 (2) Witnesses Smith and Baker heard you accept my offer, *so* we have a valid contract. (Target: δ_2)
- (2), and $(2 \rightarrow 1)$ are added to plaintiff's commitments.
$\mathbf{j_5}$: $valid(A_1, A_1')$ (Target: π_4)
(The judge completes plaintiff's argument with the rules r_1: $2 \Rightarrow$ *offer* & *acceptance* and r_2: *offer* & *acceptance* $\Rightarrow$ (1) *valid contract*. Here, $\Rightarrow$ is a defeasible implication.)
- The completed argument $A_1' = (\{2, r_1, r_2\}$ *so* $1)$ is the root of the first associated argument tree.
- r_1 and r_2 are added to judge's commitments.
$\boldsymbol{\delta_6}$: *argue* A_2: (3) There is no written agreement, (4) oral agreements never occur in this business, *so* (5) there was no offer and acceptance. (Target: π_4)
(defendant attacks plaintiff's argument by rebutting one of its subconclusions.)
- (3), (4) and $(3 \mathbin{\&} 4) \rightarrow 5$ are added to defendant's commitments.
$\mathbf{j_7}$: $valid(A_2, A_2')$ (Target: δ_6)
(The judge completes A_2 with r_3: $(3 \mathbin{\&} 4) \Rightarrow 5.$)
- The completed argument $A_2' = (\{4, r_3\}$ *so* $5)$ is added to the first argument tree as a reply to A_1'.
- r_3 is added to judge's commitments.
$\mathbf{j_8}$: *argue* A_3: $r_4 :\Rightarrow A_1' \succ A_2'$, *so* (6) $A_1' \succ A_2'$ (Target: δ_6)
(The judge declares defendant's attack dialectically invalid by stating a neutralising priority argument. Since the antecedent of r_4 is empty, the judge simply says 'plaintiff's evidence outweighs defendant's counterevidence')
- r_4 and $(r_4 \rightarrow 6)$ are added to judge's commitments.
- A_3 is added to the first argument tree as a reply to A_2'.
$\boldsymbol{\delta_9}$: *argue* A_4: (7) I was insane during the negotiations, *so* (8) my acceptance was void. (Target: π_4)
(Defendant backtracks to δ_6 with an alternative counterargument that her insanity gives rise to an exception to r_2.)
- (7 and $(7 \rightarrow 8)$ are added to defendant's commitments.
$\mathbf{j_{10}}$: $valid(A_4, A_4')$ (Target: δ_9)
(The judge completes A_4 with r_5: $7 \Rightarrow 8$.)
- (8) and r_5 are added to judge's commitments.
$\mathbf{j_{11}}$: $burden(8, \delta)$ (Target: δ_9)
(The judge has to assign the burden of proving the exception to defendant, so the dialectical asymmetry switches.)
- The completed argument $A_4' = (\{7, r_5, A_4' \succ A_1'\}$ *so* $8)$ is added to the first argument tree as a second reply to A_1'.
$\boldsymbol{\pi_{12}}$: *why* insane? (Target: 7 in δ_9)
$\mathbf{j_{13}}$: $burden(7, \delta)$ (Target: π_{12})
$\boldsymbol{\delta_{14}}$: *argue* A_5: (9) This court's document declares me insane, *so* (7) I was insane. (Target: π_{12})
- (9) and $(9 \rightarrow 7)$ are added to defendant's commitments.
$\mathbf{j_{15}}$: $valid(A_5, A_5')$ (Target: δ_{14})
(The judge has to complete A_5 with r_6: $9 \Rightarrow 7$, since legally, an avidavit is conclusive evidence for its content.)
- The argument A_4' as extended with A_5' to $A_4' \otimes A_5' = (\{9, (9 \Rightarrow 7), 7, r_5, r_6\}$ *so* $8)$ replaces A_4' in the third argument tree as a reply to A_1'.
$\boldsymbol{\pi_{16}}$: *illegal* (δ_{14}, A_6: (10) defendant provided the document after the written pleadings phase, *so*, (11) document is inadmissible evidence). (Target: δ_{14})
(Plaintiff argues that the avidavit is inadmissible evidence.)
- (10) and $(10 \rightarrow 11)$ are added to plaintiff's commitments.
$\mathbf{j_{17}}$: $valid(A_6, A_6')$ (Target: π_{16})
(The judge completes plaintiff's argument with the legal rule r_7: *document provided after written phase* $\Rightarrow$ *document is inadmissible evidence*)
- The completed argument $A_6' = (\{10, r_7, \}$ *so* $11)$ is the root of a second associated argument tree.
- r_7 is added to judge's commitments.
$\mathbf{j_{18}}$: *argue*$(A_7$: (12) plaintiff's interests are not damaged, *so*, (13) r_7 does not apply.) (Target: π_{16})
(The judge rejects plaintiff's illegality claim, undercutting his argument with an argument that the procedural violation did not damage plaintiff's interests.)
- The argument $A_7 = (\{12, r_8\colon\ 12 \Rightarrow 13, \}$ *so* $13)$ is added to the second associated argument tree as a reply to A_6'.

- (12) and r_8 are added to judge's commitments.
$\pi_{\mathbf{19}}$: *why* is this a court's document? (Target: 9 in δ_{14})
(Plaintiff backtracks to π_{16}, challenging a premise of defendant's counterargument to his main argument.)
$\mathbf{j_{20}}$: $burden(\neg 9, \pi)$ (Target: π_{19})

At this point, no moves of the adversaries are left to consider, while the constructed basis for dispute does not enable new arguments, so the dispute has terminated. Since π_1 is labelled *out*, defendant is the winner.

7. DISCUSSION

7.1 Limitations

Our dialogue game has some features which motivate further research. Firstly, the requirement that each move replies to a preceding move excludes some useful moves, such as lines of questioning in cross-examination of witnesses, with the goal of revealing an inconsistency in the witness testimony. Typically, such lines of questioning do not want to reveal what they are aiming at.

Secondly, at several points, the present ways to model legal-procedural acts have no clear one-to-one correspondence with the language of legal decisions. For instance, judges often merge their decisions on internal and dialectical strength of an argument: usually they regard the presence of a defeating counterargument as evidence that the argument is not internally valid. Practical testing should determine whether this indicates a flaw of our model, or whether instead a clear separation of internal and dialectical validity would improve the quality of legal decisions.

7.2 Related research

Because of space limitations, the comparison with related research has to be brief. The investigations of this paper build on the earlier AI & Law models of legal procedural mentioned in the introduction. As I said there, my main contributions are that the present dialogue system is related to a general framework, and that it gives a more realistic account of the judge's role in legal procedure.

7.3 Conclusion

This paper has presented a formal model of legal procedure which improves previous models both legally and technically. Although a particular procedure was modelled, the techniques used seem to be generally applicable. The model also illustrates the usefulness of previous AI & Law research on the logic of legal argument. The model may have some limitations, but my hope is that it will provide a fruitful basis for further formal work as well as for implemented procedural support tools for legal officials.

8. REFERENCES

[1] T. Bench-Capon. Specification and implementation of Toulmin dialogue game. In *Legal Knowledge-Based Systems. JURIX: The Eleventh Conference*, pages 5–19, Nijmegen, 1998. Gerard Noodt Instituut.

[2] G. Brewka. Dynamic argument systems: a formal model of argumentation processes based on situation calculus. *Journal of Logic and Computation*, 2001. To appear.

[3] T. Gordon. The Pleadings Game: an exercise in computational dialectics. *Artificial Intelligence and Law*, 2:239–292, 1994.

[4] T. Gordon and N. Karaçapilidis. The Zeno argumentation framework. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Law*, pages 10–18, New York, 1997. ACM Press.

[5] J. Hage, R. Leenes, and A. Lodder. Hard cases: a procedural approach. *Artificial Intelligence and Law*, 2:113–166, 1994.

[6] H. Jakobovits. *On the Theory of Argumentation Frameworks.* Doctoral dissertation Free University Brussels, 2000.

[7] R. Leenes. *Hercules of Karneades: Hard Cases in Recht en Rechtsinformatica.* (Hercules or Karneades: hard cases in law and legal informatics). Twente University Press, Enschede, 1998. (In Dutch).

[8] A. Lodder. *DiaLaw. On Legal Justification and Dialogical Models of Argumentation.* Law and Philosophy Library. Kluwer Academic Publishers, Dordrecht/Boston/London, 1999.

[9] R. Loui. Process and policy: resource-bounded non-demonstrative reasoning. *Computational Intelligence*, 14:1–38, 1998.

[10] R. Loui, J. Norman, J. Alpeter, D. Pinkard, D. Craven, J. Linsday, and M. Foltz. Progress on Room 5: A testbed for public interactive semi-formal legal argumentation. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Law*, pages 207–214, New York, 1997. ACM Press.

[11] J. MacKenzie. Question-begging in non-cumulative systems. *Journal of Philosophical Logic*, 8:117–133, 1979.

[12] J. Pollock. *Cognitive Carpentry. A Blueprint for How to Build a Person.* MIT Press, Cambridge, MA, 1995.

[13] H. Prakken. On dialogue systems with speech acts, arguments, and counterarguments. In *Proceedings of the 7th European Workshop on Logic for Artificial Intelligence (JELIA'2000)*, number 1919 in Springer Lecture Notes in AI, pages 224–238, Berlin, 2000. Springer Verlag.

[14] H. Prakken. Modelling defeasibility in law: logic or procedure? *Fundamenta Informaticae*, 2001. To appear.

[15] H. Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese*, 127:187–219, 2001.

[16] H. Prakken and T.F. Gordon. Rules of order for electronic group decision making – a formalization methodology. In J.A. Padget (ed.) *Collaboration between Human and Artificial Societies. Coordination and Agent-Based Distributed Computing*, Springer Lecture Notes in AI 1624, 246–263, Berlin, 1998.

[17] H. Prakken and G. Sartor. A dialectical model of assessing conflicting arguments in legal reasoning. *Artificial Intelligence and Law*, 4:331–368, 1996.

[18] G. Vreeswijk. The computational value of debate in defeasible reasoning. *Argumentation*, 9:305–341, 1995.

[19] D. Walton and E. Krabbe. *Commitment in Dialogue. Basic Concepts of Interpersonal Reasoning.* State University of New York Press, Albany, NY, 1995.