DiaLaw: Levels, Dialog Trees, Convincing Arguments

Arno R. Lodder

Computer/Law Institute, Vrije Universiteit Amsterdam, The Netherlands
lodder@rechten.vu.nl, http://www.rechten.vu.nl/~lodder

Abstract

DiaLaw is a dialogical model of legal justification. An important characteristic of the formal and implemented model DiaLaw is that both logic-based arguments and mere convincing statements can be modelled. The support of both structural (logic-based) arguments and procedural arguments (mere convincing statements) is expressed by different levels in the dialog. In this paper so-called dialog trees are used to show the connection between the assertions (both structural and procedural arguments) put forward on the different levels in the dialog. Furthermore, a definition of the previously informally introduced procedural argument is given, and procedural arguments are compared with similar features in games like MacKenzie's DC and Bench-Capon's TDG. Finally, it is indicated that procedural arguments cannot be defined in the logical layer, but have to be defined in the procedural layer (cf. Prakken's layered models of argumentation).

1 Introduction

DiaLaw is a formal and implemented, dialogical model of legal justification (Lodder & Herczog 1995, Lodder 1999) that allows both logic-based arguments and mere convincing statements. Moreover, a clear analysis of legal argumentation can be obtained because levels show how simple statements support other statements. I use so-called dialog trees to pictorially represent the support of the arguments. In this paper I elaborate on the above characteristics not only because I consider them significant for models of legal argumentation, but also because it seems that I did not address these characteristics clearly enough in the past.[1]

The paper is structured as follows. First, I give a brief characterization of the theory on legal justification underlying DiaLaw. Next, I use a dialog about sickness benefit of Bench-Capon (1998) to demonstrate the working of

1 For instance, Bench-Capon (1998) referred to DiaLaw as a game "based only on logical consequence", Hage (2000) puts DiaLaw under the "systems with a dialogue protocol that is largely logic-based", and although Verheij's (1998) analysis of DiaLaw is accurate, he mentions neither the modeling of mere convincing statements, nor the use of levels.

Arno R. Lodder, 'DiaLaw: Levels, Dialog Trees, 'Convincing Arguments', in: H.Jaap van den Herik et al. (eds), *Legal Knowledge Based Systems, JURIX 1999, The Twelfth Conference*, Nijmegen: GNI, 1999, p. 61-72.

the levels in the dialog and to show a dialog tree. My representation of this dialog is also meant as a rebuttal to Bench-Capon's (1998) observation that models like DiaLaw do "not show the desired structure".[2] Subsequently, a definition of the previously informally introduced procedural argument (Lodder 1997) is given, and procedural arguments are compared with similar features in MacKenzie's DC (1979) and Bench-Capon's TDG (1998). Finally, Prakken's (1997) layers in dialogical models of argumentation are evaluated and it is indicated where the procedural arguments should be located in his model.

2 My view on legal justification

The key idea is that justification of a statement can solely be based on agreement among participants in a dialog.[3] There is no criterion outside the dialog that can determine what is justified (Lodder 1999). So, consensus determines the status of a legal statement as justified or not. Does this mean that many statements never become justified, because of lack of consensus? Yes and no.

Yes, in a particular dispute between two participants a dialog will often remain an unfinished one, meaning that no consensus is reached. As a consequence, the statement has not become justified between the participants. If it is important for (one of) them to settle the dispute, they have to call for an independent third, e.g., an arbiter or a judge. In case the dispute is fought out in court, agreement is located at a meta-level. Namely, the players agree that they will respect the outcome (apart from appeal) of the legal suit. However, the party for whom the decision is contrary to what he desired, generally will not think that the decision is justified. Nevertheless, he will act according to the verdict.

No, it depends on the audience. For instance, consensus can be reached within a particular group that the death penalty should be used, because it is a good measure. To that particular group, the statement is justified. Simultaneously, another group can agree that the death penalty is not a good measure at all. So at the same time two conflicting statements can be justified within two different groups. If these groups meet, surely no consensus will be reached, so no statement (neither that the death penalty is a good measure, nor that it is not) becomes justified. In this situation the Parliament can be considered an independent third that decides whether the death penalty is used.

3 Dialog trees and levels: clear insight into argumentation

At the previous JURIX conference Bench-Capon (1998) presented his Toulmin Dialogue Game (TDG). After the presentation of a sample dialog he made some observations about games like DiaLaw:

> "The intention here is to present a natural dialog (...) Contrast this with a less rich dialogue, based only on logical consequence, such as is found in Lodder (1998). (...) The result is a dialogue which, while it may exhibit impeccable logic, is rather stilted, and

2 Because 'desired' is a subjective notion that is therefore hard to rebut, my rebuttal has to be put into perspective with the wording of the original attack.

3 Note that ideas similar to mine were expressed previously by amongst others Aristotle, Perelman, Habermas, and Alexy.

does not show the desired structure. Similar criticism can be levelled at other logic based games such as that of MacKenzie (1979)."

In my opinion MacKenzie's game is also capable of modelling dialogs in a natural way. Moreover, the game TDG looks rather similar to MacKenzie's DC. Prakken (1999) even heads TDG under "MacKenzie-style dialogue systems with the possibility of counterargument".

In this section DiaLaw is used to model the dialog of Bench-Capon in order to show how a "desired structure" of argumentation is obtained, while the arguments are not at all based on "impeccable logic". At the end a dialog tree is presented to show the connection between the assertions put forward on the different levels.

Because the purpose here is to show the opportunities of DiaLaw, the formalism of Bench-Capon's TDG is not discussed in detail (see Bench-Capon 1998). TDG is built around the Toulmin argument schemata that is changed in some respects. The commonly used chaining of Toulmin schemata is applied (e.g. data of one argument can be the claim of another), and a new element, the presupposition, has been added (see also section 4). Another element, the qualifier, is left out. The rules of the game are in general set out clearly: each move type is described and preconditions, postconditions and completion conditions are defined.[4]

Note that the present analysis of Bench-Capon's dialog is a reconstruction and not a construction. If a statement has been justified, e.g., a judge has grounded his verdict, the justification can be analyzed. This analysis is an example of reconstruction. If a statement is not yet justified, e.g., the case is still in court, the justification has to be created. This creation is an example of construction. Since the dialog between the two players is already finished, the analysis below is a reconstruction. Because a reconstruction is meant to analyze the actual argumentation, on some points the dialog is slightly moderated. All adaptations of the original dialog will be explained.

In the moves of the dialog, a player performs an illocutionary act concerning some sentence. The move takes place on a particular level. Therefore a move is represented by a term P for the player, a term A for the illocutionary act, a term S for the sentence, and a term L for the level. An element B is needed that indicates either for which sentence the move is an argument, or to which sentence the move is a reaction. This is expressed in the following definition (Lodder 1999):

Definition 1 - the dialog move

A dialog move M_i ($i>0$) is an ordered 5-tuple (P, A, S, L, B), where

$P \in$ {Player1, Player2},

$A \in$ {claim, question, accept, withdraw},

S, B are formulas of Language,

$L \in N$ (set of natural numbers, including zero).

4 However, to me it is not always clear how the formalism works. For instance, at move eight player B asks justification for the claim that John is fit for work. The comments on this move are that the rule is accepted. If we look at the formalism, the consequence of the WHY-move is not that a player becomes committed.

As far as the formal representation of the dialog is concerned, only in case a claim is made the move is shown.[5] The dialog tree shows the layered structure that is based on the levels of the dialog. Each time a player questions a sentence, the dialog shifts to a deeper level. For instance, if a statement put forward on level 1 is questioned, the level becomes 2. The formal moves have the following format:

(P, A,
S,
L, B).

The dialog can be understood without reading the formal representations of the moves. The reason to show the formal moves anyway is that the tree is built using them. By applying the definition of dialog trees presented after the dialog, one can build a dialog tree from the formal moves. All moves that were made in Bench-Capon's dialog are numbered A1, B1, A2, B2, ..., An, Bn. If particular moves were added, I have indicated this with an apostrophe. For instance, if a move is added after A1, this new move is referred to as A1'. In italics some comments are made during the dialog.

A1: John should not get sickness benefit (0-level)
(A, claim,
~benefit(john),
0, dialaw)

This move is on the zero, initial level. Since it is the first move it cannot refer to any previous move, reason why, by convention, 'dialaw' is used to express 'B'.

B1: Why? (0-level)
A2: He is a lazy person (1-level)
(A, claim,
lazy(john),
1, ~benefit(john))

That John is a lazy person is adduced at the first level and is meant as support for the 0-level sentence that John should not get benefit.

B2: I accept he's lazy (1-level)
A3: Lazy people should not get sickness benefit (1-level)
(A, claim,
valid(rule(lazy(X),~benefit(X))),
1, ~benefit(john))

In DiaLaw, if a rule is claimed, a player claims that a particular rule is valid. The predicate valid is derived from Reason-Based Logic (Hage & Verheij 1994).

B3: Why? (1-level)
A4: I withdraw the rule just claimed (2-level)

The player A is not able to provide arguments for the validity of the rule, reason why he withdraws it.

A6: If someone is fit for work, they should not get sickness benefit (1-level)

5 Besides that it would take much space to show all moves, for the building of the dialog tree the other acts are not necessary.

(A, claim,
valid(rule(fit(X),~benefit(X))),
1, ~benefit(john))
B6: Why? (1-level)
A8: See the SSA 1984, section 32(1) (2-level)
(A, claim,
reason(SSA84-321, valid(rule(fit(X),~benefit(X))),
2, valid(rule(fit(X), ~benefit(X))))

The player A mentions a section from the Social Security Act in order to support the rule claimed in A6. This so-called reason is an example of a structural argument. It supports the valid rule because of its structure.

B8: I accept that if someone is fit for work, they should not get sickness benefit (2-level)
A5: John is fit for work (1-level)
(A, claim,
fit(john),
1, ~benefit(john))

The player A is trying to show that the condition of the rule claimed in move A6 is satisfied.

B8: Why? (1-level)
A9: I saw him gardening (2-level)
(A, claim,
gardening(john),
2, fit(john))

That John was gardening, is provided as support for the claim that John is fit for work.

B9: I accept that (2-level)

The player B accepts that John was gardening. Because player B does not yet accept that John is fit for work, the player A has to adduce additional arguments to support his claim that John is fit for work. The player A does so in the next move.

A9': That he was gardening is the reason why he is fit for work (2-level)
(A, claim,
reason(gardening(john), fit(john)),
2, fit(john))
B9': It is not (2-level)
(B, claim,
~reason(gardening(john), fit(john)),
2, reason(gardening(john), fit(john))

Player B denies that the fact that John was gardening is a reason why he is fit for work and takes over the initiative in the dialog.

A10: Why? (2-level)
B10: Fit for work means fit for suitable work. (3-level)
(B, claim,
valid(rule(suitable_work(X), fit(X))),
3, ~reason(gardening(john), fit(john))
A10': Why? (3-level)
B10': Commissioner, RS3/58 (4-level)

(B, claim,
reason(RS3/58, valid(rule(suitable_work(X), fit(X)))
3, valid(rule(suitable_work(X), fit(X)))

The player B provides backing for his rule, by mentioning relevant case law. He claims that the mentioned case, RS3/58, is the reason why the rule claimed in B10 is a valid one.

A10'': I accept that (4-level)

According to Bench-Capon player A is forced to accept the rule, because authority is provided. It surprises me that providing authority forces someone to accept rules. What if the backing is false?

B10'': Gardening is not suitable work for John (3-level)
(B, claim,
~suitable_work(gardening(john)),
3, ~reason(gardening(john), fit(john))

Player B tries to show that the condition of the rule claimed in B10 is not satisfied.

A11: Why? (3-level)

B11: John is a University Lecturer (4-level)
(B, claim,
university_lecturer(john),
4, ~suitable_work(gardening(john)))

This is the final argument. The fact that John is a lecturer makes player A believe that John, despite his gardening activities, is not fit for work and as a consequence that he indeed should get sickness benefit.

A12: I withdraw my initial claim (4-level)

I changed the order of the moves. In the original dialog first “John is fit” was claimed, and afterwards the rule “if fit then no benefit”. The reason for this alteration is that the players now agree about the rule before further elaborating on the statement “John is fit”. By changing the sequence, a better analysis of the argumentation is obtained. The refinement of the rule “if fit then no benefit” in the original dialog is left out (fit for work means fit for work someone normally pays for), because it is not used in the remainder of the dialog. After rule B9 some moves were added. This has to do with a different view on burden of proof. The purpose of DiaLaw is modelling legal justification. Therefore, in DiaLaw the proponent has the responsibility to bear his burden of proof (to justify statements he puts forward), and that burden only switches after a denial by the opponent. Finally move B10 was split into two moves in order to show the backing by case law more explicitly. Some other differences are discussed in the next section on procedural arguments.

Dialog trees are built in the course of a dialog. After each move necessary changes are made to the tree. These depend on the status of the commitment. It would lead to far astray to explain the b-part of the definition below, because knowledge of the way commitment is handled is necessary for that purpose. Basically, if both players are committed to a sentence (in which case a box has a solid border), then one player has claimed the sentence and the other player has explicitly accepted that sentence (Lodder 1999).

Definition 2 - The dialog tree

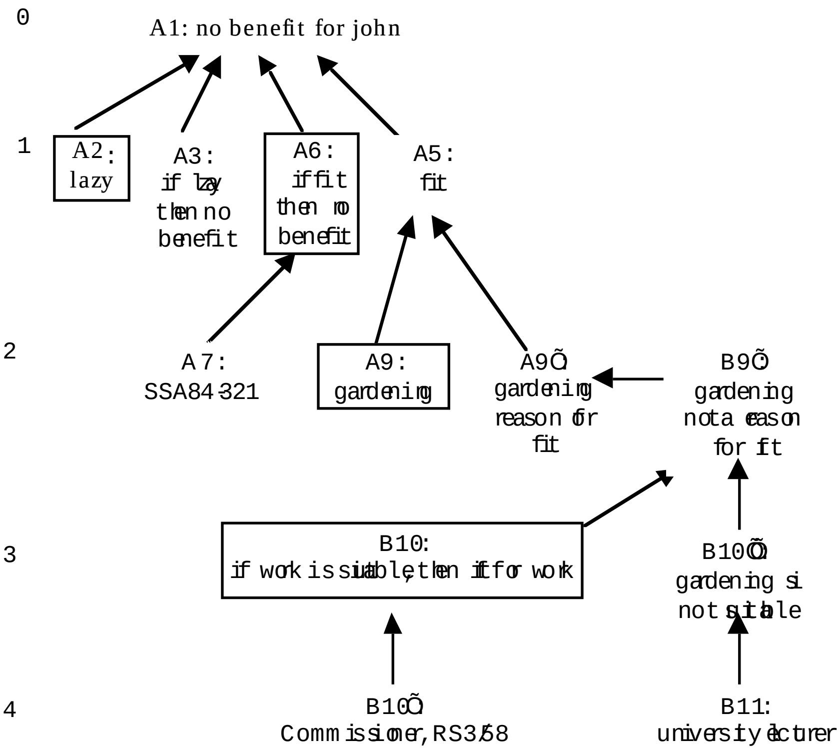
DT_1 consists of a dotted box on the level 0 with the sentence S that was claimed in the first move.

A dialog tree DT_i (i>1) is equal to DT_{i-1}, except in the following cases:

a. if M_i = (–, claim, S, T, B),
then a dotted box with the sentence S is added, on the level T, with an arrow pointing to the box with the sentence B;

b. if M_i = (–, A, S, –, –), where A ∈ {accept, withdraw},
then the dotted boxes with the sentence S change in the following cases:
  1. the border vanishes in case (–, S) ∉ C_i;
  2. the border becomes solid in case (–, S) ∈ $C_i \setminus O_i$.

In the tree the sentences are represented in an informal way. The formal sentences that were used to build the tree can be found in the move that is indicated in the box. The dialog tree after the last move of the dialog is the following.



The tree shows at a single glance the following:

- all the sentences claimed in the dialog;
- at what level the sentences were claimed;
  The utter left side of the picture shows the levels. Each sentence that was claimed in the dialog is shown in the tree on the level it was claimed.
- what sentence an adduced sentence reacts to or means to justify;
  If an arrow points up, then the sentence in the box at the lower end of the arrow is meant to justify the sentence that is at the higher end of the arrow. If an arrow is horizontal, then the sentence in the box at the right end of the arrow is a reaction to the sentence at the left end of the arrow.

- whether both players (solid border), one player (dotted border), or no player are/is committed to a particular sentence (no border).[6]

4 Procedural arguments revisited

In models of legal argumentation the use of non-logical arguments, mere convincing statements is important. In order to distinguish between logical and non-logical arguments, a division into structural and procedural ones was introduced in (Lodder 1997). The reason the former are called arguments is due to their structure (e.g., Modus Ponens: 'A' & 'A therefore B', so 'B'). The above dialog shows that justification of statements can be based on other than structural arguments. For instance, the final plank to the argumentation that John should indeed get benefit was not a structural argument but a simple statement: "John is a University Lecturer". The particular sequence of claims can make a statement an argument for a previous one. In other words, a statement can become an argument in a procedure if the *structure of the procedure* makes it an argument. "John is a University Lecturer" is an argument, because the structure of the procedure makes it an argument, namely an *argument* supporting the sentence "Gardening is not suitable work for John". Whether an argument is a procedural argument, depends on who claimed it and why he claimed it. The qualification of a sentence as a procedural argument is determined by the procedure. The following definition is from Lodder (1999).

Definition 3 - Procedural arguments
If
M_i = (P, claim, S, L, B),
and (P, claim, B, L-1, –) $\in D_{i-1}$[7]
then
to the player P, S is a procedural argument supporting B.

Note that according to this definition a structural argument that is used to support another sentence (so if the sentence S is a structural argument) is also a procedural argument. Only if a sentence is looked at in connection with other sentences of the procedure, the sentence can be a procedural argument. Any sentence that is claimed to *support* another sentence is a procedural argument to the player who claimed it. This means that any sentence claimed by a player is a procedural argument to him, except:

- If the sentence is claimed in reaction to another sentence, so not to support. See above, move B9';
- If the sentence is the first of the dialog, since then there are no other sentences.

How does my approach of procedural arguments relate to MacKenzie's DC (1979) and Bench-Capon's TDG (1998)? On the basis of three characteristics of my procedural arguments I will compare my approach with theirs:

- More than one statement to support a claim;
- Argument are not necessarily based on rules;

6 If a dialog is finished, the tree does not contain dotted boxes. However, after the last but one move the boxes B11, B10", B9', A9', A5 and A1 had dotted borders.

7 D refers to the dialog, which is an ordered set of all the moves that were made.

- Distinction between statements itself and statement as an argument.

In DiaLaw, a statement can be supported by several other statements. It might be necessary to introduce supporting statements one by one: if a player believes his opponent will be satisfied with one supporting statement in which case other –maybe harder to justify- supporting statements do not have to be adduced; in case a player does not know all supporting statements the moment he adduces his first, etc. In DiaLaw, a statement put forward on level 1, can be supported by as many as necessary statements on level 2. In DC only if 'if A then B' has been put forward (which is done automatically if a player adduces A, after the opponent questioned B), it is clear that a statement (e.g. A) is adduced as support for another statement (e.g. B). In DiaLaw this explicit way to support can be used (e.g. the structural argument reason(A, B)), but also implicit support of statements can be expressed via the levels. In TDG data (which can be one supporting statement or a set) that support a claim can be introduced. There is a possibility to adduce an additional supporting statement, if the opponent asks for it with the presupposition move. If the warrant 'If A then B' has been adduced, and the opponent wants a necessary implicit condition to be made explicit, he can ask for it. For instance, if the actual rule runs 'If A & C then B' (or, if C is necessary for justifying B anyway). This is a nice feature that is in line with one of the ideas underlying procedural arguments: not everything has to be made explicit.

In the previous paragraph it became clear that in DC arguments are always based on rules. In DiaLaw rules can be adduced to justify the support relation between two statements, but not every argument have to be based on rules. Also in TDG arguments do not have to be based on rules, but there is one difference. If a support relation is questioned, for instance whether A supports B, the expected reaction in TDG is 'If A then B'. In my opinion, the opponent already knows that A supports B, and would be helped more with additional support then with a rule that makes the relation between A and B explicit.

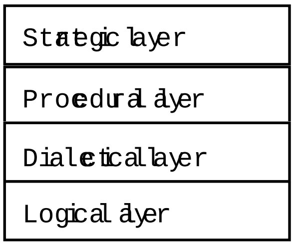
If in DiaLaw a statement is not claimed in reaction to another statement it is always a procedural argument for the proponent of the claim. But, two aspects of statements are distinguished. First, the statement itself, and, second, the statement as an argument in a procedure. Let the following example illustrate this. In move B2 the player B accepts that John is lazy. If in DiaLaw a statement is accepted, the acceptance is only about the content of the statement itself. Although the statement was adduced to support the initial statement here, acceptance does not imply that the statement is accepted as support, so as a procedural argument. At this point of the dialog the player A still has the burden to prove that his initial statement is justified.[8] Therefore player A continues to support his initial statement in move A3 by claiming a particular rule. In TDG, the player B asked explicitly for additional support: "OK, but so what?". The meaning of accept in DiaLaw is in fact exactly the same: "I accept your statement, but I am not yet convinced that the statement you tried to support with it is justified too". The idea is that if player B considers the statement 'John is lazy' enough support for the

8 On burden of proof in dialogical models see Leenes (1999) and Prakken (1999).

initial statement, he would have accepted the initial statement as a reaction to the claim 'John is lazy'. So on this point in the dialog the burden still rests on player A, reason why player B does not have to ask A to bear it. Yet another approach is taken in DC, where the consequence of move A2 would be that player A not only becomes committed to 'John is lazy', but also to 'John is lazy' as a reason for the initial statement. So, in DC player A becomes automatically committed to the underlying reason, in TDG player B can ask for the underlying reason, and in DiaLaw it is the responsibility of player A to provide the underlying reason.

5 Procedural arguments and Prakken's four layer model

During a meeting the GMD in the spring of 1994, Brewka and Gordon presented a three layer model for mediating systems that they later that year presented at the AAAI workshop on Computational Dialectics (Brewka & Gordon 1994). Prakken (1997) has proposed a four layer model.[9] Because the four layer model by Prakken is the most recent, and best worked out model, I will base my discussion on this one.
The first level is the logical level. On this level it is defined amongst others what contradiction is, and when arguments support conclusions. The second level is the dialectical level. On this level notions of argument, counterargument, defeat, etc. are defined. For example, an argument supporting a particular conclusion can be defeated by a counterargument supporting the opposite conclusion. The third level is the procedural level. On this level the rules on how to conduct the dialog are defined and what speech acts can be used. The fourth level is the strategy level. As Prakken claims correctly, knowing the rules of the game is only one aspect of becoming a good player. Good debating without debating skills is not imaginable. As I understand Prakken, each layer builds on the proceeding layer. This can be pictured as follows.

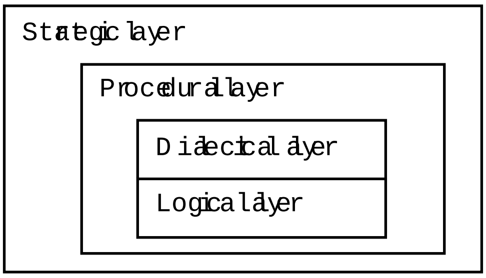


Prakken argues that the dialectical layer provides the link between logical and procedural models. I believe he is right, but only partially. Indeed, procedural models add to the logical, product models.[10] The interaction between arguments is important. However, not only the genesis (first layer) and use of (second layer) logical arguments should be incorporated in a procedural model. Besides that the procedural layer builds on the logical and dialectical layer, namely in case structural arguments are used, in the procedural layer arguments must be allowed that do not build

9 Beside his three layer model presented at the ICAIL 1995, and a early sketch was presented in the summer of 1994 on a Dutch KNAW-symposium.

10 If the product of justification is studied, general structures of support between sets of premises and conclusions are defined. If the process of justification is studied rules are defined that determine for each stage of the process whether a statement is justified.

on the logical layer, namely procedural arguments. The same holds for the strategic layer. In this interpretation, the upper two layers can be depicted as follows.



So, structural arguments are defined on the first level, the interaction between arguments (defeat, rebut, etc.) is defined on the second level. Because the rules of the procedure define the structure that can make ordinary statements into procedural arguments, these arguments should be defined in the third layer. If in a procedural model of argumentation the layers build on each other like Prakken suggests, all arguments have to be logically compelling. However, the power of procedural, dialogical models is that argumentation can be modelled both by using logically compelling arguments and by using convincing arguments which are not necessarily logically compelling. That Prakken only allows logically compelling arguments is, in my opinion, a pity, because as long as these arguments are omitted, not all argumentation can be modelled adequately.

6 Concluding remarks

In this paper I addressed two elements of DiaLaw that I consider important for formal models of legal argumentation. First, levels can be used to show the structure of the argumentation. I used the dialog trees in order to show the argumentation structure of a dialog. Second, the possibility of modelling mere convincing statements or procedural arguments is indispensable in models of legal argumentation. I provided a definition of procedural arguments. Finally, I commented on the four layer model by Prakken, and indicated in which layer the procedural arguments should be located.

DiaLaw is implemented in Prolog 2.0 for Windows 3.1 (p2w3), but, unfortunately, the program p2w3 does not run anymore and the company who made it is bankrupt. Therefore, my colleague Paul Huygen has adapted the code of DiaLaw to run in SWI-prolog. By the time you are reading this, a version of DiaLaw can be downloaded, so that anyone has the opportunity to use DiaLaw (see http://www.rechten.vu.nl/~lodder).

References

Bench-Capon 1998

T.J.M. Bench-Capon, 'Specification and implementation of Toulmin Dialogue Game', in: J.C. Hage et al. (eds.), *Legal Knowledge Based Systems: JURIX 1998: The Eleventh Conference*, Nijmegen: GNI, 1998, pp. 5-19.

Brewka & Gordon 1994

G. Brewka & T. Gordon, 'How to buy a Porsche: an approach to defeasible decision making', in: *Workshop notes AAAI workshop 1994 – Computational Dialectics*, 1994, pp. 28-38.

Hage 2000
J.C. Hage, 'Dialectical Models in Artificial Intelligence and Law', to appear in: *Artificial Intelligence & Law*, 2000.

Hage & Verheij 1994
J.C. Hage & B. Verheij, Reason-based logic: a logic for reasoning with rules and reasons, in: *Law, Computers & Artificial Intelligence*, vol. 3, no. 2/3, 1994, pp. 171-209.

Leenes 1999
R.E. Leenes, 'Burden of Proof in DiaLaw and Dutch civil Procedure', in: *Proceedings of LawTech '99*, Hawaii, USA, 1999.

Lodder & Herczog 1995
A.R. Lodder & A. Herczog, 'DiaLaw – A dialogical framework for modelling legal reasoning, *Proceedings of the fifth International Conference on Artificial Intelligence and Law*, New York: ACM Press, 1995, pp. 146-155.

Lodder 1997
A.R. Lodder, 'Procedural arguments', in: A. Oskamp *et al.* (eds.), *Legal Knowledge Based Systems – JURIX 1997: The Tenth Conference*, Nijmegen: GNI, 1997, pp. 45-58.

Lodder 1999
A.R. Lodder, *DiaLaw – on legal justification and dialogical models of argumentation*, Dordrecht: Kluwer Academic Publishers, 1999.

Mackenzie 1979
Mackenzie, J.D., 'Question-Begging in non-cumulative systems', in: *Journal of Philosophical Logic*, 8, 1979, pp. 117-133.

Prakken 1997
Prakken, H. (1997), *Logical tools for modelling legal argument: a study of defeasible reasoning in law,* Dordrecht: Kluwer Academic Publishers.

Prakken 1999
H. Prakken, 'On Formalising Burden of Proof in Legal Argument', in: H.J. van den Herik et. al., *Legal Knowledge Based Systems – JURIX 1999: The Twelfth Conference*, Nijmegen: GNI, 1999, pp. 89-101.

Verheij 1998
B. Verheij, 'ArguMed – A Template-Based Argument Mediation System for Lawyers', in: J.C. Hage *et al.* (eds.), *Legal Knowledge Based Systems: JURIX 1998, The Eleventh Conference*, Nijmegen: GNI, 1998, pp. 113-130.