Argumentation Schemes for Reasoning about Factors with Dimensions

Katie ATKINSON[1], Trevor BENCH-CAPON[1] Henry PRAKKEN[2], Adam WYNER[3],
[1]*Department of Computer Science, The University of Liverpool, England*
[2]*University of Utrecht, Department of Information and Computing Sciences and University of Groningen, Faculty of Law, The Netherlands*
[3]*Department of Computing Science, The University of Aberdeen, Scotland*

Abstract.
In previous work we presented argumentation schemes to capture the CATO and value based theory construction approaches to reasoning with legal cases with factors. We formalised the schemes with ASPIC+, a formal representation of instantiated argumentation. In ASPIC+ the premises of a scheme may either be a factor provided in a knowledge base or established using a further argumentation scheme. Thus far we have taken the factors associated with cases to be given in the knowledge base. While this is adequate for expressing factor based reasoning, we can further investigate the justifications for the relationship between factors and facts or evidence. In this paper we examine how dimensions as used in the HYPO system can provide grounds on which to argue about which factors should apply to a case. By making this element of the reasoning explicit and subject to argument, we advance our overall account of reasoning with legal cases and make it more robust.

Keywords. case based reasoning, dimensions, argumentation schemes

1. Introduction

Understanding how cases are used in legal reasoning is central to AI and Law. This paper is part of an on-going project to articulate our current understanding of this topic. Our overall aim is to present reconstructions of AI approaches to reasoning with cases as argumentation schemes, formalised in ASPIC+ [9, 8], with the following agenda:

1. to provide a precise and transparent account of what is involved in the reasoning, in particular to articulate the argumentation involved;
2. to specify the knowledge that must be supplied to the system by the analyst;
3. to use a formalism which will enable logical properties such as consistency and closure to be proven of the formalised knowledge;
4. to provide precise specifications which can readily be implemented using standard techniques.

ASPIC+ is suitable for this since it satisfies the third point. In addition, the fourth point is addressed since the formalisation, being founded on a knowledge base and rules, maps immediately into a logic programming language such as Prolog, and thus effectively serves as a program specification.

In [12] and [3] we presented argumentation schemes to capture the reasoning of CATO [1] and value based theory construction [5] respectively. Legal reasoning can be seen as moving from evidence to facts, from facts to factors and from factors to legal consequences. The factor based reasoning of CATO and [5] is essentially but a single step of argument, covering only the last of these stages. In [1] and [5] cases are sets of factors assigned by an analyst outside the system: the schemes presented here bring the assignment of factors within the scope of the system, and so open this aspect to explicit argument. This is needed because some cases (e.g. *Pierson v Post*) turn on which factors are assigned. We base this argumentation on dimensions as in HYPO [2]. Since factors correspond to a range of points on a dimension and favour a particular party, factors can be assigned by using the facts to determine which dimensions are relevant, and the points on these dimensions satisfied by the case. This paper is thus a small but necessary increment to the work in [13, 12, 3], which extends the coverage of formal legal argumentation one step further towards the case facts.

2. Preliminaries

We now summarise the formal frameworks used in this paper. An *abstract argument framework*, as introduced by Dung, [7] is a pair $AF = \langle \mathcal{A}, \mathit{defeat} \rangle$, where $\mathcal{A}$ is a set of arguments and *defeat* a binary relation on $\mathcal{A}$. A subset $\mathcal{B}$ of $\mathcal{A}$ is *conflict-free* if no argument in $\mathcal{B}$ defeats an argument in $\mathcal{B}$ and it is said to be *admissible* if it is both conflict-free and also defends itself against any attack, i.e., if an argument A_1 is in $\mathcal{B}$ and some argument A_2 in A but not in B defeats A_1, then some argument in $\mathcal{B}$ defeats A_2. A *preferred extension* is then a maximal (with respect to set inclusion) admissible set. Dung defines several other types of extensions but they are not used in our model.

The ASPIC+ framework [9, 8] gives structure to the arguments and defeat relation. It defines the notion of an *argumentation system*, which consists of a logical language $\mathcal{L}$ with a binary contrariness relation $^-$ and two sets of inference rules $\mathcal{R}_s$ and $\mathcal{R}_d$ of *strict* and *defeasible inference rules* defined over $\mathcal{L}$, written as $\varphi_1, \ldots, \varphi_n \rightarrow \varphi$ and $\varphi_1, \ldots, \varphi_n \Rightarrow \varphi$. Informally, that an inference rule is strict means that if its antecedents are accepted, then its consequent must be accepted *no matter what*, while that an inference rule is defeasible means that if its antecedents are accepted, then its consequent must be accepted *if there are no good reasons not to accept it*. An argumentation system also contains a function n which for each defeasible rule in $\mathcal{R}_d$ returns a formula in $\mathcal{L}$. Informally, $n(r)$ is a wff in $\mathcal{L}$ which says that the defeasible rule $r \in \mathcal{R}$ is applicable.

In this paper we use an argumentation system in which $\mathcal{L}$ is a many-sorted first-order language with equality further specified in the coming sections, its contrariness relation corresponds to classical negation, the strict rules $\mathcal{R}_s$ are all valid first-order inferences over $\mathcal{L}$ and the defeasible rules $\mathcal{R}_d$ and naming function n are as specified further on.

An ASPIC+ arguments are constructed from a knowledge base $\mathcal{K}$, which contains two disjoint kinds of formulas: the *axioms* $\mathcal{K}_n$ and the *ordinary premises* $\mathcal{K}_p$. The formal definition of an argument is as follows:

Definition 2.1 [Argument] An *argument* A on the basis of a knowledge base $\mathcal{K}$ in an argumentation system $(\mathcal{L}, ^-, \mathcal{R}_s, \mathcal{R}_d)$ is:

1. φ if $\varphi \in \mathcal{K}$ with: $\texttt{Prem}(A) = \{\varphi\}$; $\texttt{Conc}(A) = \varphi$; $\texttt{Sub}(A) = \{\varphi\}$; $\texttt{TopRule}(A)$ = undefined.

2. $A_1, \ldots A_n \rightarrow$/$\Rightarrow \psi$ if $A_1, \ldots, A_n$ are arguments such that there exists a strict or a defeasible rule
$\texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \rightarrow$/$\Rightarrow \psi$ in $\mathcal{R}_s$/$\mathcal{R}_d$.
$\texttt{Prem}(A) = \texttt{Prem}(A_1) \cup \ldots \cup \texttt{Prem}(A_n)$; $\texttt{Conc}(A) = \psi$; $\texttt{Sub}(A) = \texttt{Sub}(A_1) \cup \ldots \cup \texttt{Sub}(A_n) \cup \{A\}$;
$\texttt{TopRule}(A) = \texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \rightarrow$/$\Rightarrow \psi$.

An argument is *strict* if all its inference rules are strict and *defeasible* otherwise, and it is *firm* if all its premises are in $\mathcal{K}_n$ and *plausible* otherwise.

An argumentation system and a knowledge base are combined with an *argument ordering* into an *argumentation theory*. The argument ordering could be defined in any way, for example, in terms of orderings on $\mathcal{R}_d$ and $\mathcal{K}_p$.

Definition 2.2 [Argumentation theories] An *argumentation theory* is a triple $AT = (AS, \mathcal{K}, \preceq)$ where AS is an argumentation system, $\mathcal{K}$ is a knowledge base in AS and $\preceq$ is a partial preorder on the set of all arguments on the basis of $\mathcal{K}$ in AS (below denoted by $\mathcal{A}_{AT}$).

Arguments can be attacked in three ways: attacking a conclusion of a defeasible inference, attacking the defeasible inference itself, or attacking a premise. Inference attacks are defined with the help of the naming function n that assigns to each element of $\mathcal{R}_d$ a well-formed formula in $\mathcal{L}$. For our argumentation system, ASPIC+'s definitions of attack can be simplified as follows:[1]

Definition 2.3 [Attacks] A *attacks* B iff A *undercuts*, *rebuts* or *undermines* B, where:

- A *undercuts* argument B (on B') iff $\texttt{Conc}(A) = -n(r)$ for some $B' \in \texttt{Sub}(B)$ such that B''s top rule r is defeasible.
- A *rebuts* argument B (on B') iff $\texttt{Conc}(A) = -\varphi$ for some $B' \in \texttt{Sub}(B)$ of the form $B''_1, \ldots, B''_n \Rightarrow \varphi$.
- Argument A *undermines* B (on φ) iff $\texttt{Conc}(A) = -\varphi$ for some ordinary premise φ of B.

Attacks combined with the preferences defined by an argument ordering yield three kinds of defeat.

Definition 2.4 [Successful rebuttal, undermining and defeat]

- A *successfully rebuts* B if A rebuts B on B' and $A \not\prec B'$.
- A *successfully undermines* B if A undermines B on φ and $A \not\prec \varphi$.
- A *defeats* B iff A undercuts or successfully rebuts or successfully undermines B.

ASPIC+ thus defines a set of arguments with a binary relation of defeat, that is, it defines abstract argumentation frameworks in the sense of [7]. Formally:

Definition 2.5 [Argumentation framework] An *abstract argumentation framework (AF) corresponding to an argumentation theory AT* is a pair $< \mathcal{A}$, *Def* $>$ such that:

- $\mathcal{A}$ is the set $\mathcal{A}_{AT}$ as defined by Definition 2.1,

[1] In the definitions below, $-\neg\varphi$ denotes φ, while if φ does not start with a negation, $-\varphi$ denotes $\neg\varphi$.

- *Def* is the relation on $\mathcal{A}$ given by Definition 2.4.

Thus any semantics for abstract argumentation can be applied to ASPIC+.

3. Dimensions in Legal Case Based Reasoning

Dimensions, rather than factors, were used in HYPO [2], from which CATO developed. Dimensions have an *extent* and points along the extent. Whereas factors are either simply present or absent, a dimension, if present, may favour either the plaintiff or defendant and *to a particular degree*. Dimensions thus encompass a range of points, with the extreme pro-plaintiff point at one end and the extreme pro-defendant point at the other. Thus, at some unspecified point along the range the dimension will cease to favour the plaintiff and start to favour the defendant.

There are two ways in which dimensions convert to factors. Dimensions may be Boolean, in which case the dimension maps to two disjoint factors: often one of the factors is not in fact used since it does not strengthen the case (e.g. in HYPO it does not help the defendant if he did not bribe an employee). Alternatively, where the dimension is a numeric range or an ordered series of descriptive points, it may give rise to a set of factors, each corresponding to part of the dimension.

Although factors became the *de facto* standard representation of cases, (e.g. [6, 11, 5]), the need for dimensions was argued for in [4], since a case may turn on which factor is applicable: which factor covers the particular point on the dimension occupied by the current case will determine which party is favoured by the fact situation. *Pierson v Post* is an example: the dispute turns on when pursuit can be counted as justifying lawful possession, for which different degrees of progress towards bodily seizure need to be recognised.[2]

As analysed in [6], the only factors present are *NotCaught* and *OpenLand*, both of which are pro-defendant. But the plaintiff argued that Post was sufficiently close to taking bodily possession of the fox that it should be *counted as caught*. Whereas the representation in [6] effectively decides the case by assigning *NotCaught*, to do justice to Post's argument we need to be able to argue that the facts justify assigning *Caught* instead. What this means in ASPIC+ terms is that the factors attributed to the case must cease to be elements in the knowledge base and instead require justification. What form might this justification take?

In *Pierson v Post*, the defendant's argument was in terms of particular commentaries. It is accepted by both sides that how close Post was to bodily securing the fox is relevant - the commentaries are all directed towards this point. Thus the stage of the pursuit is a dimension. The question then becomes *at which point on this dimension does the plaintiff start to be favoured?* In other words, at which point on the dimension does the factor *NotCaught* cease to apply and the factor *Caught* start to apply? In the next sections we will develop a specialist argumentation scheme for attributing factors to cases to enable this sort of argument.

[2] In brief the facts were these. Post was chasing a fox with horse and hounds and had cornered it when Pierson intervened and killed it with a fence pole. Post sued Pierson, but lost because he had not taken possession of the fox.

4. Facts

Our proposal is that cases should be represented as bundles of *facts* rather than *factors*. This means that the KB will need to represent cases using a set of statements of the form $hasFact(Case, Fact)$ Here $Case$ will be a case name, such as $PiersonVPost$, but $Fact$ requires some more consideration, since we do need to be able to compare facts across cases, which means that they will need to be at a sufficient level of abstraction. Moreover, not every fact will play a part in the decision: some facts will be relevant and some irrelevant. Further, we will need a picture of how facts relate to factors, so that we can argue about the attribution of factors to cases.

To illustrate some of these points, let us suppose that an undecided case concerning capturing a wild animal is being argued where the plaintiff claims that the animal was $Caught$ on the basis of hot pursuit. This is what Livingston, speaking for Post, did indeed argue in *Pierson v Post*. However, even though the argument was put forward it was not sustained, and so *Pierson v Post* in fact establishes that the factor *NotCaught* applies to the benefit of the defendant where the fact is only hot pursuit. But to argue as Livingston did, even unsuccessfully, requires that cases are not represented in terms of summarising factors (e.g. *NotCaught*) but as the underlying *facts* which determine the presence of the factor (e.g. *Hot pursuit*). In the next section we will see how case decisions can give rise to rules for assigning factors based on precedents.

For these reasons we will need to restrict the facts that can be used: to do this we will take our inspiration from [2] and [4]. The relevance of a fact will be ensured by relating it to a dimension. A dimension will be a name and a set of linearly ordered points on the dimension. These will be represented in $\mathcal{K}_p$ as *dimension(Dimension, TermSet)*. For example we might have:

$$dimension(Pursuit, \{Possessed, CaptureInevitable, Wounded, HotPusuit, ChaseStarted, Seen, None\}).$$

The linear ordering on the points of a dimension d with respect to a factor f is formally expressed with a $\leq_{d,f}$ predicate symbol in $\mathcal{K}_p$, along with the usual ordering definitions. The factor is needed in $\leq_{d,f}$ because a greater point may move towards or away from a factor, depending on at which end the extreme pro-plaintiff point lies.

Now a possible fact will be a pair of a dimension and an element of the corresponding set of dimension points. We write this as $hasFact(Case, Dimension.Point)$. Technically, $Dimension.Point$ is a function expression with the period as function symbol and with $Dimension$ and $Point$ as its two arguments. We also add the corresponding sorts for *dimensions*, *facts* and dimension *points* to $\mathcal{L}$.

Note that we could define these facts using other, more concrete, facts. For example we could give a list of sufficient conditions for saying that capture was inevitable. Where dimensions are described in terms of a continuous element (e.g. age may be described in terms of number of years) we normalise this into a series of bands (e.g $\{Infant, Child, Adolescent, Adult, Senior\}$). These definitions will be beyond explicit argument: we will not consider them further here, but rather assume that this part of the reasoning is the responsibility of the analyst, so that cases are represented as bundles of dimension-point pairs.

As in [12], there will be a set of factors available, and these will favour either the plaintiff or the defendant. Factors are therefore represented using two predi-

cates $pFactor(factor)$ and $dFactor(factor)$. An axiom expresses that nothing can be both a pfactor and a dFactor. For this example we need consider only two factors: $pFactor(Caught)$ and $Factor(NotCaught)$. Thus $hasFact(Case, Pursuit.Possessed)$ should imply $hasFactor(Case, Caught)$, and so favour the plaintiff, and $hasFact(Case, Pursuit.None)$ should imply $hasFactor(Case, NotCaught)$ and so favour the defendant, but at which point on the dimension the animal begins to be counted as $Caught$ remains open to debate.

In the next section we will provide argumentation schemes for attributing factors to cases on the basis of this representation of the facts of a case.

5. Reasoning from Facts to Factors

We now turn to the formalisation of argument schemes relating factors to facts. In [13], a preliminary analysis of some aspects was presented; here we provide a fuller account using the improved formalisation of [12]. The knowledge sources we will recognise here are precedent cases and legal commentaries (we also regard minority arguments as commentaries). We will also recognise justifications based on linguistic interpretation of the terms involved, particularly to establish factors representing the extreme dimension points. Moreover, sometimes there will be no supporting justification at all: sometimes we will want to test arguments in a court setting in an effort to establish a precedent, but there is no support for the rule beyond our contention that it should be recognised. As discussed in the previous section, we have a binary relation $hasFact(Case, Fact)$, where a $Fact$ is a dimension-point pair.

- $hasFact(case, fact)$; e.g. $hasFact(PiersonVPost, Pursuit.HotPursuit)$

We also introduce a five-place rule relation:

- $rule(rulename, fact, factor, justifier, justificationType)$,
  where *fact* is a dimension-point list pair, a *justifier* is a named commentary or case decision or none, and the *justificationType* is from *Commentary*, *Precedent*, *Definition*, or *Contention*.

Accordingly, we add to $\mathcal{L}$ sorts for *rules, justifiers* and *justificationTypes*.

We offer a sample of five rules, which are discussed further below.

- $rule(Rule1, Pursuit.CaptureInevitable, NotCaught, Justinian, Commentary)$
- $rule(Rule2, Pursuit.Wounded, Caught, Puttendorf, Commentary)$
- $rule(Rule3, Pursuit.None, NotCaught, None, Definition)$
- $rule(Rule4, Pursuit.HotPursuit, NotCaught, Tomkins, Contention)$
- $rule(Rule5, Pursuit.HotPursuit, Caught, Livingston, Contention)$

These legal rules are included in $\mathcal{K}_p$. The last argument of the *rule* predicate stands for the legal justification type of the rule. *Rule3* needs no specific justification: it is true simply in virtue of the standard English meaning of the words. Rules 1 and 2, however, do need a justifier to specify the particular justifiation. Contentions have the name of the person advancing the contention as justifier. Of the above *Rule1* and *Rule2* are justified by commentaries, Justinian and Puttendorf respectively; if we accept Justinian (Puttendorf) as an authoritative commentator we accept *Rule1* (*Rule2*). Rules *Rule4* and *Rule5*

are not independently justified. At the time of *Pierson*, there were no precedent cases to provide justification, and the dispute was whether *Rule4* or *Rule5* should hold. Following the decision of *Pierson v Post*, we may take it that *Rule4* holds, though Livingston argued for *Rule5*: we can then change the justification type of *Rule4* to *Precedent* and the justifier to *PiersonVPost*. *Rule5*, since rejected in the case, could either be discarded or attributed to Livingston by changing the justification type to *Commentary* and the justifier to *Livingston*, depending on how much regard we have for Livingston. The decision to amend or discard *Rule5* will be made on the basis of the interpretation of the analyst.

This permits the following argumentation scheme. We first give a stylised natural-language version and then formalise it in ASPIC+:

CS1: From facts to factors

Premise 1: The Current case has Fact1
Premise 2: There is a Justification of a certain Type to regard Fact2 as an instance of Factor
Premise 3: Fact1 points at least as strongly to Factor as Fact2
Conclusion: The Current case has Factor

CS1$(ruleName, fact, factor, justifier, justificationType, curr)$:

$$\frac{\begin{array}{c} hasFact(curr, fact1) \\ rule(ruleName, fact2, factor, justifier, justificationType) \\ fact1 = dimension.point_1 \\ fact2 = dimension.point_2 \\ dimension.point_1 \leq_{d,f} dimension.point_2 \end{array}}{hasFactor(curr, factor)}$$

Recall that, as discussed above, our facts are dimension points. Note also that CS1 can be instantiated using both *Rule4* and *Rule5*, whereas *Caught* and *NotCaught* are intended to be exclusive. Adding suitable factor incompatibility axioms to $\mathcal{K}_n$, will enable the resulting arguments to rebut one another.

CS1 is presented as a single scheme, and Premise 2 can be instantiated using justifications of any of our four types. This means that CS1 has four variants, one for each of the justification types in Premise 2: CS1a for justifications by commentary, CS1p for justification by precedent, CS1d for justification by definition and CS1c for justification by contention. This is useful when we consider the characteristic ways of attacking arguments made using the scheme, since the ways of attacking arguments made using the scheme depend significantly on the justification type.

Thus a commentary may be attacked as too old (as Justinian was by Livingston in *Pierson v Post*), or perhaps as relating to the wrong kind of law: Justinian was an commentary on Roman Law, and this expertise is arguably not transferrable to Common Law, or modern Civil Law. Precedents may be from a different jurisdiction. Or they may be from a lower court (cf the principle of *lex superior*) and so not be binding in the current context. Again they may be very old (cf *lex posterior*[3]). Age is not, however, necessarily a problem: an argument against a precedent because it is old, may be countered by

[3]Nothing corresponds to *lex specialis*, perhaps because all precedents are about particular cases, and so of equally generality.

examples of its recent use by relevant courts. Indeed the history of use of a precedent should be considered in these arguments: some cases are established as leading cases by frequent citation, others may have been derogated in a previous decision. Definitions may be attacked as too broad or too narrow, or simply incorrect in the context. Different jurisdictions have different conventions, and conventions may change over time. Contentions are not susceptible to special attacks: that the contention had been advanced is a straightforward fact. Rules justified by contention are, in any case, transient, on their way to becoming precedent rules, commentary rules, or being discarded altogether.

As well as these particular attacks on the various justification types, the rule is authored by an analyst, and perhaps endorsed and supported by others. One of the purposes of our new scheme is to expose the work of the analyst to debate. So we can ask whether the analyst is reliable, whether the analyst is biased, and whether the rule has been endorsed by other analysts. This is particularly important for rules relating to commentaries, which always involve interpretation. For example in *Rule2*, some might think that Puttendorf required inevitable capture rather than mere wounding.

We therefore need to store some additional information to enable these attacks. For example we might use a relation *commentary(Name, Floreat, Code)* (e.g. commentary(Justinian, 300, Roman). For precedents we need *precedent(Name, Court, Date)*, and also a set of facts recording citations and derogations, e.g. *cited(Precedent, CitingCase, Date)*. For example, precedent(KeebleVHickergill, QueensBench, 1707) and cited(KeebleVHickergill, PiersonVPost, 1805). The breadth of definition can be determined from the rule; the rule, if we take account of the stronger dimension points, will cover a number of points, and the more points the broader the definition. Thus Rule3 offers a very narrow definition: a different rule, defining *NotCaught* with *Pursuit.CaptureInevitable* would be maximally broad. Each justification type will have a set of undercutters using this information and corresponding to the characteristic attacks. For example the undercutter used by Livingston to attack the use of Justinian is:
U1$(ruleName, fact, factor, justifier, Commentary, curr)$:

$$\frac{commentary(justifier, floreat, code) \wedge (floreat < 1400)}{\neg\textbf{CS1}(ruleName, fact, factor, justifier, Commentary, curr)}$$

which attacks commentaries pre-1400 as too old. Similar undercutting schemes would express the other characteristic ways of attacking the variants of CS1. If desired additional information about the justification types could be included, giving rise to additional characteristic attacks, and additional undercutters corresponding to them.

As well as these ways of attacking arguments using CS1, we can also suggest ways of using the information to support them, and where we have rebutting arguments, choosing between them. One obvious thing to do is express preferences between justification types. Preferring precedents over commentaries, commentaries over mere definition and definitions over contentions would be one natural ordering. Within justification types, precedents may be preferred according to frequency and recency of citation, and some commentaries may be more highly regarded than others. Additionally, if we have information about the authors of the rules, we may trust some analysts more than others, or we may choose the rule endorsed by the greater number of analysts.

Taken together these ways of attacking and supporting the rules used to assign factors means that we can render transparent and open to debate what was effectively a black box in the CATO system.

To ensure that the factors that are derivable in a given case are the only factors in that case we include the defeasible rule $\Rightarrow \neg hasFactor(case, factor)$ in $\mathcal{R}_d$ with a lower priority than any other rule. The unique-names and domain-closure axioms from [12] are retained and are now also formulated for the new sorts. Then any argument for an unnegated $hasFactor$ conclusion will strictly defeat any argument using this defeasible rule.

The representation that we provide enables a level of argumentation that is not available in HYPO and CATO. We have chosen to represent facts using a list of possible points along particular dimensions as in, e.g. [4] and [5], but, if desired, some more complicated ontology representing the domain could be used. As noted above, there are some similarities with the validity rules of [10], but whereas there the validity in question was a legal rule, here it is rules for the qualification of facts as factors, which are used by the analyst and are themselves more like heuristics than legal rules.

Thus far we have taken dimensions to give rise to two factors, one plaintiff and one defendant. Thus any given dimension point will give rise to at most one factor (there may be a gap in the middle of the range, but this is not a problem: the gap can be closed by additional rules, and different ways of closing the gap may be resolved by subsequent decisions). If, however, we have a dimension with three ranges, we will find that CS1 will ascribe two factors to part of the range. Thus suppose we have three factors: a defendant factor *NoEffort*, a plaintiff factor *Effort* and a stronger plaintiff factor *SuccessfulEffort*. We might map these factors to the Pursuit dimension using the following rules:

- $rule(Rule7, Pursuit.CaptureInevitable, SuccessfulEffort, None, Definition)$
- $rule(Rule8, Pursuit.ChaseStarted, Effort, None, Definition)$
- $rule(Rule9, Pursuit.Seen, NoEffort, None, Definition)$

Now if the facts of a case are that capture is inevitable, rules 7 and 8 will both apply, suggesting that both *SuccessfulEffort* and *Effort* are factors in the case. We may be happy with this, or we may wish to avoid having both factors apply, to avoid double counting. Although examples of both can be found in [1], we regard exclusive ranges as being preferable. Thus we wish to ensure that if *SuccessfulEffort* is present, *Effort* is not. To express this relationship, we need an additional predicate $subsumes(factor1, factor2)$ and the axiom

$$\forall case, factor1, factor2.hasFactor(case, factor1) \wedge subsumes(factor1, factor2) \supset \neg hasFactor(case, factor2)$$

We have provided the means to argue about the ascription of factors to cases, based on a set of facts expressed as dimension points and rules taken from a variety of sources. We have given a scheme applicable to any source of justification, and suggested how arguments based on this scheme might be attacked, for several types of justification. We have indicated what additional information information is needed to support such argumentation, and how this can be used to resolve conflicts based on rebuttals. Finally we have discussed how factors may exclude opposing or subsumed factors.

6. Conclusion

In this paper we have provided argumentation schemes to enable explicit reasoning about which factors should be considered to be present in particular cases. In previous work

this task has been left entirely to the unchallengable opinion of the analyst. So as to remain linked to preceding work in AI and Law we have taken as our starting point facts as represented by points on dimensions, as used in [2]. The dimensions determine which facts are relevant, and the points provide a degree of abstraction which allows comparison of cases. The schemes allow us to argue for the assignment of a factor to a case on the basis of such facts in terms of a commentary, a precedent case, a standard interpretation of the words, or simply as a contention. Allowing the debate to include discussion of the assignment of factors to cases is a small but necessary step towards meeting our overall goal of developing a robust and comprehensive account of reasoning about legal cases with argumentation schemes. Much more remains to be done in future work including: argumentation about differing strengths of factors, arguments about how observable facts relate to the points on a dimension used here, and arguments about how facts are assigned on the basis of, perhaps conflicting, evidence.

References

[1] V. Aleven. *Teaching case-based argumentation through a model and examples*. PhD thesis, University of Pittsburgh, Pittsburgh, PA, USA, 1997.

[2] K. Ashley. *Modelling Legal Argument: Reasoning with Cases and Hypotheticals*. Bradford Books/MIT Press, Cambridge, MA, 1990.

[3] T. Bench-Capon, H. Prakken, A. Wyner, and K. Atkinson. Argument schemes for reasoning with legal cases using values. In *The 14th International Conference on Artificial Intelligence and Law*, pages 1–10, 2013.

[4] T. Bench-Capon and E. L. Rissland. Back to the future: dimensions revisited. In B. Verheij and et al., editors, *Proceedings of JURIX 2001*, pages 41–52, Amsterdman, The Netherlands, 2001. IOS Press.

[5] T. Bench-Capon and G. Sartor. A model of legal reasoning with cases incorporating theories and values. *Artificial Intelligence*, 150(1-2):97–143, 2003.

[6] D. H. Berman and C. D. Hafner. Representing teleological structure in case-based legal reasoning: the missing link. In *Proc. of the 4th ICAIL*, pages 50–59. ACM Press, 1993.

[7] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

[8] S. Modgil and H. Prakken. A general account of argumentation with preferences. *Artificial Intelligence*, 195:361–397, 2013.

[9] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2010.

[10] H. Prakken. Reconstructing Popov v. Hayashi in a framework for argumentation with structured arguments and Dungean semantics. *Artificial Intelligence and Law*, 20(1):57–82, 2012.

[11] H. Prakken and G. Sartor. Modelling reasoning with precedents in a formal dialogue game. *Artificial Intelligence and Law*, 6(2-4):231–287, 1998.

[12] H. Prakken, A. Wyner, T. Bench-Capon, and K. Atkinson. A formalization of argumentation schemes for legal case-based reasoning in aspic+. *Journal of Logic and Computation*, 2013.

[13] A. Wyner, T. Bench-Capon, and K. Atkinson. Towards formalising argumentation about legal cases. In *Proc of the 13th ICAIL*, pages 1–10, 2011.