Abstract

The Zeno Argumentation Framework is a formal model of argumentation based on the informal models of Toulmin and Rittel. Its main feature is a labelling function using arguments to compute heuristic information about the relative quality of the alternative positions proposed as solutions for some practical issue. The Zeno Argumentation Framework was designed to be used in mediation systems, an advanced kind of electronic discussion forum with special support for argumentation, negotiation and other structured forms of group decision-making. The ability to compute the status of positions in Zeno is the most significant advance over the informal Rittel’s Issue-Based Information System (IBIS) model. It transforms IBIS from a lifeless method to organize and index information into a playing field for stimulating debate. The interested parties can see immediately whether their positions are currently “winning” or “losing”, given the arguments which have been made so far, motivating them to marshall still better arguments in favor of their positions.

1 Problem Statement

The cities of Bonn and Sankt Augustin are planning a residential area and high-technology “park” in an area between the two cities. This area was zoned for agricultural purposes only, so a formal proposal was made to change the zoning ordinance. According to local law, changes of this kind are subject to a formal review procedure. This procedure requires that the plan be made available to the general public for their comments, criticisms and suggestions.

In the European GeoMed project (GeoMed stands for “Geographical Mediation”), our main task at GMD is to design and implement a mediation system for the World-Wide Web, which shall enable public review procedures in a way which is more efficient, transparent, fair and, it is hoped, democratic than current practice. The idea is to use the Web to enable interested citizens and representatives of public interest groups to more easily access and view the development plans and to take part in an electronic discussion forum to express and share their views and opinions.

A mediation system, as we conceive it, is a kind of computer-based discussion forum with particular support for argumentation. In addition to the generic functions for viewing, browsing and responding to messages, a mediation system uses a formal model of argumentation to facilitate retrieval, to show and manage dependencies between arguments, to provide heuristic information focusing the discussion on solutions which appear most promising, and to assist human mediators in facilitating the collaborative search for solutions which are acceptable to the parties concerned.
The persons taking part in the discussion will typically come from a variety of professional backgrounds, with different sources of information and different preferred methods for processing this information. A city planner might model the information at his or her disposal using a geographical information system. A politician might rely on high-level briefs from his or her support staff. A social scientist might have constructed a statistical model. Last but not least, as our goal is to open up public policy and planning procedures to a wide public, most participants will rely on their common sense, without using some more professional methodology.

In summary, the intended application scenarios are characterized by the following features, which will lead to requirements for the mediation system:

- The participants have access to different sources of information, knowledge and models. They do not share a set of premises. The amount and quality of their background knowledge will vary widely.
- The participants use different methods to organize, structure, process and evaluate information and knowledge. Most participants will not be proficient in formal methods from mathematics, logic, or computer science.
- The participants have competing viewpoints and interests. In many cases they will be adversaries, with only a limited willingness to cooperate. The social atmosphere is one of caution, suspicion and mistrust.
- The issues to be decided are practical, rather than theoretical. The central problem is to make a choice among alternatives; the issue is which position or claim to believe, or which course of action to take.
- Finally, time is of the essence. A decision must be reached within a fixed period of time. Other resources required to find an acceptable solution, such as money and information, are also limited.

The rest of this paper is organized as follows. The following section presents a software engineering analysis of the mediation system and includes an example discussion showing some of the kinds of argument we would like to support. The purpose of this first section is to make clear the requirements for our formal model of argumentation, presented next. We finish up with a section on related work and some conclusions.

## 2 Software Engineering Analysis

This section presents a brief software engineering analysis of the problem, using the popular Object Modelling Technique (OMT) [Rumbaugh and et. al., 1991]. OMT provides three basic diagramming techniques for designing and illustrating a system from three complementary perspectives: object diagrams show data structures and their relationships, functional diagrams show data flow between processes, and dynamic models show events, states and causal dependencies.

To start, Figure 1 is a functional model for an example planning procedure with four participants with different professional backgrounds.

The participants communicate with each other through the mediation system. The mediation knowledge available to the system includes some general purpose model of argumentation and, in a fully developed version, knowledge about the specific administrative procedures applicable, such as an environmental review procedure.

The project record stores the messages exchanged by the participants, as well as the evolving model of this particular discussion, using the ontology of the general
The mediator actor represents the human mediator or moderator of the discussion. The mediation system is conceived as an intelligent support system for human mediators. Our goal is to support human mediators, not automate the mediation task. The human mediator remains responsible for interpreting the natural language text of messages and identifying the argumentation elements of these texts, applying his knowledge of the Zeno formal model. Modeling the argumentation structure of a natural language text is an interpretative task comparable in principle to modeling the propositional structure of a text using some formal logic. The task is much simpler, however, because the argumentation structure of a text is at a higher level of granularity. In practice, the mediator may sometimes need to collaborate with the authors to try to reach agreement about how to model the argumentation structure of some messages. During this interaction the author may be given an opportunity to revise the message. This is but one small part of the mediator’s general responsibility to facilitate the discussion, helping to find solutions which are, ideally, acceptable to all parties.

The next figure, Figure 2, is an OMT object model of the mediation system. Each box in this diagram represents a class, in the sense of object-oriented programming. The links between the boxes represent associations and inheritance relationships.

The Project Record and the classes below it are more relevant. They show the internal structure of the Project Record persistent store we saw above in the functional model. A project record consists of the discussion and a queue of incoming messages. (The mediator will have a chance to review incoming messages before they are made public.) A discussion object is associated with some (zero or more) messages and models of the discussion. Notice that, in principle, there can be several models of the discussion, to support different views or interpretations of the original messages.

A discussion model is constructed by creating a marked message for each message
Figure 2: Object Model of Mediation

to be included in the model. A marked message is, if you will, a model of the message showing its argumentation structure. The marked messages are parsed to create and extend a relational model of the whole discussion. Each record in this relational model is called an **element**, as shown in the object model.

Elements are of several kinds, as shown in the next figure, Figure 3. This figure provides a high-level view of the structure of the Zeno Argumentation Framework. One objective of this paper is to refine this high-level model in a more detailed algebraic model of the framework.

Our model is a formal variant of Horst Rittel’s informal Issue-Based Information System (IBIS) model of argumentation [Kunz and Rittel, 1970, Rittel and Webber, 1973]. The basic elements of the IBIS model may be linked together in almost every imaginable way, to produce finite argumentation graphs. Using hypertext techniques, nice graphical user interfaces have been built for browsing IBIS graphs. The nodes of the graphs can contain arbitrary natural language expressions and other forms of media. Such systems can be quite useful for structuring and organizing information, despite their lack of formal semantics, and are particularly easy to build today using the protocols and document formats of the World-Wide Web.

The next figure, Figure 4, is a **speech act graph** showing a hypothetical discussion between a husband and wife about which car to buy. The speech act graph shows a particular state of argumentation, at one moment in time. The precise timing of speech acts is not important in our model of argumentation. The speech act graph only encodes time to the extent that responses to prior speech acts necessarily occur at some later time. That is, implicitly the speech acts are partially ordered by a reply relation, as in the threads of typical Internet discussion forums.

The speech act graph illustrated in Figure 4 could have resulted from the following discussion:
Figure 3: Object Model of Argumentation Elements

**Husband.** Honey, we’ve been thinking about buying a new car. Do you have something particular in mind? \(I_1\)

**Wife.** Well, yes. I think we should buy a Volvo station wagon. \(P_2\)

**Husband.** But, that’s such a family car. \(P_3\) Let’s buy a nice fast \(P_3\) sports car. A Porsche would be great. \(P_1\)

**Wife.** Isn’t a Porsche pretty expensive? \(P_4\) And besides, I think we should buy a safer car. Volvos are built like a tank. \(P_8\)

**Husband.** What makes you think Volvos are so safe? \(I_2\)

**Wife.** Don’t you watch TV? Haven’t you seen the advertisements? \(P_7\)

**Husband.** Oh, come on Honey. I read a report in Auto Sports Today which cited some government accident statistics. \(P_6\) Do you know what? Volvos were said to be involved in more fatal accidents than almost any other brand. I think these government statistics are more objective than the advertisements of the manufacturer. \(P_11\) And besides, having a fast car is more important to me than having a safe car. \(P_9\)

**Wife.** Why? \(I_3\)

**Husband.** Look. I’ve been wanting a fast sports car ever since I finished law school. An attorney in this town has to have a dynamic image. \(P_{12}\)

**Wife.** Yes, dear. But what about Betty and Susan. We have to think of the safety of our kids first. \(P_{13}, P_{14}\)

**Husband.** I guess you’re right about that. But where does that leave us? I still think I rather pay a few thousand dollars more for a Porsche than drive such a boring family car \(P_{10}\).

Even though this is a short, if somewhat construed example, it already may be becoming difficult to keep track of all of the arguments and their interrelationships.
Figure 4: A Speech Act Graph

Which position has the most support at the moment? The Volvo or the Porsche? A good mediation system should make it easy to quickly obtain an overview of the state of the debate.

3 Dialectical Graphs

Given the information in the model of a discussion, using the Zeno model of argumentation introduced in the previous section, it becomes possible to label the alternative positions of an issue, using a kind of reason maintenance procedure [Doyle, 1979]. This support for a kind of inference is an important advantage of our system over the informal Toulmin [Toulmin, 1958] and IBIS models of argumentation.

This section presents a formal, algebraic reconstruction of our model of argumentation, ending with a definition of our labelling function. This definition will be a recursive, mathematical specification, not an efficient algorithm for computing labels. Although it could be implemented using a functional style of programming in a straightforward way, a real implementation should use techniques from the reason maintenance literature for caching labels and incrementally propagating changes through the dialectical graph as it is extended with new issues and positions.

The examples below will be drawn from the Porsche example as displayed in the speech act graph, Figure 4.

Regarding our notational conventions below: Upper case identifiers, such as $P_1$ will be used to name object-level constructs in our formalism. Distinct letters will be used for each kind of object-level construct. Lower case identifiers, such as $p_1$, will be used as meta-level variables ranging over objects of the type indicated by the letter.

**Definition 1 (Positions)** Let $P$ be a set of positions, $P_1, \ldots, P_n$. 
**Example 1.** Let $P_1$ to $P_{14}$ be the set of positions shown in the speech act graph of the Porsche example. For example, $P_1$ is *Porsche*, $P_2$ is *Volvo*, $P_3$ is *Fast*, and so on. Just as when intuitive names are used for propositional letters in formalizations of propositional logic, the intuitive names for positions here simply serve to make this presentation of the system more comprehensible, but are of no importance to the formalization itself.

Notice that positions are not associated in the formal model with the participants who assert or claim them, since our labelling function for assigning acceptability values (*in* or *out*) to positions does not depend on this information. (See Definition 11.) Our Zeno mediation system, a discussion forum for the World Wide Web based on this theoretical framework, does of course store and manage this and other information, just as in other groupware discussion forums.

**Definition 2 (Terms).** Terms are defined inductively as follows:

1. Every position in $P$ is a term.
2. If $t_1$ is a term, then $\overline{t_1}$ is a term.
3. If $t_1$ and $t_2$ are terms, then $t_1 + t_2$ and $t_1 - t_2$ are terms.
4. Nothing else is a term.

For the moment, this is just syntax. The meaning of these operators is defined below, in Definition 11. Also, please excuse the overloading of the “-” symbol, as both a unary and binary operator. We appeal to the precedent set by arithmetic.

**Example 2.** *Porsche*, *Volvo*, *Fast*, *Expensive* and *Safe*, *FamilyCar* are all terms. Intuitively, terms are used to combine and weigh the arguments pro and contra some position.

**Definition 3 (Preference Expressions).** Let $t_1$ and $t_2$ be two terms. Then

1. $t_1 > t_2$ and $t_1 = t_2$ are preference expressions.
2. Nothing else is a preference expression.

**Example 3** *Fast > Safe* and *OurKids > Prestige* are preference expressions. Although not used in the Porsche example, (*Fast = Expensive*) > (*Safe = FamilyCar*) and *Porsche = Fast = Expensive* are also preference expressions.

**Definition 4 (Arguments).** Let $p_1$ and $p_2$ be two positions in $P$. There are two kinds of arguments:

1. $\text{pro}(p_1, p_2)$ is an argument.
2. $\text{con}(p_1, p_2)$ is an argument.
3. Nothing else is an argument.

In both $\text{pro}(p_1, p_2)$ and $\text{con}(p_1, p_2)$, $p_1$ and $p_2$ are called the antecedent and consequent of the argument, respectively.

**Example 4.** $\text{pro}(\text{Fast}, \text{Porsche})$, and $\text{con}(\text{FamilyCar}, \text{Volvo})$ are arguments. In these examples, the positions *Fast* and *FamilyCar* are antecedents; and the positions *Porsche* and *Volvo* are consequents.

Notice that the arrows of the arguments in the speech act graph lead from the consequent to the antecedent, reflecting the goal directed way arguments are introduced to justify positions only after they have been put at issue.

**Definition 5 (Constraints Table).** A constraints table is a finite mapping from positions to preference expressions. A position is a member of a constraints table if it is mapped in the table to some preference expression.
Example 5. \{P_8 : Fast > Safe, P_{10} : FamilyCar > Expensive\} is a constraints table.

Definition 6 (Issues). Let \( C \) be a set of positions and \( K \) be a constraints table. An issue is a pair \((C, K)\). Intuitively, \( C \), the choices, represents the alternative solutions to the problem or question raised by the issue and \( K \), the constraints table, contains the positions about the relative value of the arguments made for and against these choices.

Example 6. \( I_1 : (\{Porsche, Volvo\}, \{P_8 : Fast > Safe, P_{10} : FamilyCar > Expensive\}) \) is an issue.

Note: Not all choices of an issue are explicit in the example speech act graph. When a speaker makes an issue out of an existing position, we assume that the party who asserted the position implicitly reasserts it by “proposing” it as a “solution” to the issue raised about it. For example, when the husband made an issue out of the wife’s claim that Volvos are safe, we assume the wife reasserts that, yes, Volvos are indeed safe. More formally, we use the following interpretation rule: if raise\((p_1, i_1)\) then choice\((i_1, p_1)\). Some guidelines for using the IBIS model recommend requiring these links to be made explicit in the speech act graph. Here is an example of an issue where the choice is a position called into question by a later speech act:

Example 7. \( I_2 : (\{Safe\}, \{P_3 : GovtStats > Advertising\}) \) The positions in a constraints table may also be members of the choices of some issue. This allows argumentation about constraints.

Example 8. \( I_3 : (\{Fast > Safe\}, \{P_{14} : OurKids > Prestige\}) \) is an issue.

Definition 7 (Dialectical Graphs). Let \( P \), \( A \) and \( I \) be finite sets of positions, arguments and issues, respectively, such that the antecedents and consequents of all arguments in \( A \) are members of \( P \). Then \((P, A)\) forms a directed, finite graph in which the positions are nodes and the arguments in \( A \) are edges. Arguments link the positions by their antecedent and consequent. The tuple \((P, A, I)\) is called a dialectical graph if the positions of the choices and constraints of every issue in \( I \) are members of \( P \).

Example 9. Figure 5 depicts a dialectical graph for the Porsche discussion, given the issues shown in Examples 6-8 above. The labels \( (P_2 \text{ and } P_3) \) be discussed below in Definition 14. The argument edges of the graph are the pro and con links. The issue nodes and other links in the figure are, strictly speaking, not part of the \((P, A)\) graph, but are shown in the figure to make it easier to check whether the positions of the choices and constraints of every issue are members of \( P \), as required by the definition of dialectical graphs.

This dialectical graph was produced from the speech act graph by simply selecting its position nodes. We also used the interpretation rule discussed previously to replace two issue links by choice links: between \( P_0 \) and \( I_2 \) and between \( P_8 \) and \( I_3 \).

Definition 8 (Well-Formedness of Dialectical Graphs). A dialectical graph is well-formed if and only if:

1. The graph \((P, A)\) is acyclic;
2. No position is the antecedent of more than one argument;
3. At most one of the choice positions of an issue is the antecedent of an argument in \( A \); item All of the positions used in the preference expressions of the constraints \( K \) of an issue are antecedents of arguments pro or contra the choices \( C \) of the same issue.

The four well-formedness conditions, taken together, assure that the issues in a dialectical graph form a forest of trees, in which the nodes are positions and the edges encode arguments about positions on these issues. (Issue trees are defined formally
The final condition also assures that the constraints of an issue all directly concern the pros and cons of its own choices, and not those of issues in other parts of the dialectical graph.

It should be emphasized here that the Zeno Argumentation Framework is not a descriptive, predictive or empirical model of actual argumentation practice. We do not claim that real arguments always or even typically conform to this model. Rather, our model is a “logical” or normative model of argumentation. Our long term goal is to formalize the pragmatics of rational argumentation, comparable to the ambition of classical logic to formalize correct deductive reasoning.

There is a somewhat philosophical argument we could make here to justify these well-formedness restrictions. The argument turns on the distinction between a proposition and a position. A proposition is a declarative statement; it is either true or false, independent of its use by a party in some particular discussion. A position, on the other hand, is defined by its role and use in a discussion. A position records a speech act and is embedded in the discussion.

There are many examples of this distinction in the Porsche discussion. The wife asserted that Volvos are safe. This seemingly unqualified statement must be understood in the context of this discussion about which car to buy. There is a common understanding between the husband and wife that they will buy some car. They are concerned only about the relative safety of various cars, not about the principal safety of cars. In a discussion about the relative virtues of various modes of transportation, it would be perfectly acceptable for the wife to claim that cars are not safe. Interpreting these claims as propositions, rather than positions, could easily lead to a contradiction:

\[ \forall x, \text{if car}(x) \text{ then not safe}(x) \]
\[ \forall x, \text{if Volvo}(x) \text{ then car}(x) \]
\[ \forall x, \text{if Volvo}(x) \text{ then safe}(x) \]
\[ \text{Volvo}(a) \]

This is not an example of defeasible reasoning or reasoning with exceptions. The
intended interpretation is not that cars in general are unsafe, except for Volvos. Rather, the point here is that the acceptability of what we are calling positions depends on the problem solving context. Deciding whether to fly, drive or take a train somewhere is a different context from deciding which car to buy. Formalizations of deductive logic typically do not handle this task dependency within the formalization. This is what we are trying to achieve here, and use a different term, position instead of proposition, to emphasize this distinction and help avoid confusion.

**Definition 9 (Subissues).** Let \( \langle P, A, I \rangle \) be a well-formed dialectical graph and \( i_1 \) and \( i_2 \) be two issues in \( I \). \( i_1 \) is a subissue of \( i_2 \), denoted \( \text{subissue}(i_1, i_2) \), if and only if:

1. There exists an argument \( \text{prop}(p_1, p_2) \) or \( \text{cont}(p_1, p_2) \) in \( A \) such that \( p_1 \) is a member of the choices of \( i_1 \) and \( p_2 \) is a member of the choices of \( i_2 \), or
2. There exists a position \( p_1 \) which is a member of the choices of \( i_1 \) and a member of the constraints of \( i_2 \).

**Example 10.** In the Porsche example, the issue about the safety of Volvos, \( I_2 \), is a subissue of the root issue, about which car to buy. This is a subissue of the first kind. Also in the Porsche example, the issue about whether or not being fast is more important than being safe, \( I_3 \), is also a subissue of the root issue. This is a subissue of the second kind.

**Definition 10 (Issue Tree).** If \( \langle P, A, I \rangle \) is a well-formed dialectical graph, then the subissue relation partitions the issues of \( I \) into a set of maximum issue trees, \( \{\{I_1, \text{subissues}\}, \ldots, \{I_n, \text{subissues}\}\} \). By “maximum” here, we mean that each issue in \( I \) is a member of the set of nodes of exactly one of these trees, \( \{I_1, \ldots, I_n\} \).

**Example 11.** The three issues of the Porsche example, form an issue tree. The root issue is about which car to buy. The other two issues are subissues of this root issue, and are leaves of the issue tree.

The definitions above completely specify the syntax of dialectical graphs. What remains is to define the labelling function, which serves the function of a semantics. We begin by defining the interpretation of terms and preference expressions, using a hypothetical evaluation function. This evaluation function plays the same role as a “truth function” in defining the semantics of propositional logic. Since it is not known whether a particular evaluation function exists which is agreeable to all parties to the discussion, the label of a position, \( p \), considers all possible evaluation functions, just as propositional derivability considers all possible truth values of each propositional letter.

**Definition 11 (Interpretation of Terms)** Let \( \nu \) be some (unknown) mapping from the set of positions, \( P \), to the domain of integers, \( Z \). The interpretation of a term, denoted \( I[t] \) for some term \( t \), is defined as follows:

1. \( I[p] = \nu(p) \), if \( p \) is a member of \( P \).
2. \( I[-t_1] = -I[t_1] \), where \( - \) is the unary negation operation on integers.
3. \( I[t_1 + t_2] = I[t_1] + I[t_2] \), where \( + \) is the addition operation on integers.
4. \( I[t_1 - t_2] = I[t_1] - I[t_2] \), where \( - \) is the subtraction operation on integers.

**Definition 12 (Interpretation of Preference Expressions).** The \( = \) and \( > \) predicate symbols of preference expressions are defined to mean the equality and (strictly) greater than relations on the domain of integers, as follows:

1. \( t_1 = t_2 \) is true if and only if \( \nu(t_1) = \nu(t_2) \).
2. \( t_1 > t_2 \) is true if only if \( \nu(t_1) > \nu(t_2) \).
The integer values of positions are not asserted by users in this system. Indeed, there is no syntax supporting this in the definitions for dialectical graphs above. Rather, users only assert constraints on the values of positions, using preference expressions.

**Zeno is a qualitative, not quantitative model of argumentation.** The constraints are used in the labelling function to determine whether a position is acceptable or not, i.e. in or out.

One might object at this point that allowing the addition and subtraction of positions in preference expressions assumes that positions are independent, which may not always be the case. Although we accept this point, it is also true that two positions may in fact be independent, so we do not want to disallow combining them in preference expressions entirely. When two positions are independent, it should be possible to combine them to defeat some other position, which may be stronger than each of them separately, but not in combination [Hage et al., 1994]. Our partial solution to this problem is to allow the participants to make an issue out of this, i.e. to argue and decide for themselves whether or not to accept the combination of positions in a constraint. No additional machinery is required for this, since our framework already supports just this kind of meta-argumentation. In practice, clearly most users will not have the technical sophistication to oppose arguments on the basis of a lack of independence. This is another place where expertise of the human mediator is required to assist the users.

Now we are just about ready to define the labelling function for the positions of a dialectical graph. Following the nomenclature of Doyle’s “Truth Maintenance System”, each position which is not at issue will be labelled in, applying the “silence implies consent” principle. Each position of an issue will be labelled in or out to indicate whether or not it meets the burden of proof standard selected for the issue. A large variety of proof standards are imaginable. For a while we tried to find the right standard, which would be universally applicable for all kinds of issues. But we have come to believe that no standard is suitable for all purposes, drawing support from jurisprudence, with its variety of types of burden of proof for different kinds of proceedings and the work of [Farley and Freeman, 1995] in the field of AI and Law. We define some proof standards here, but do not claim that these exhaust all reasonable possibilities. Also, although the names we borrow from the law for some of these standards are intended to be suggestive and mnemonic, we do not claim that our formal definitions adequately capture the legal meanings of these concepts.

A few auxiliary functions will be useful several times below, to sum up the arguments pro and contra some choice. Precisely, let \((p_1, \ldots, p_n)\) and \((c_1, \ldots, c_n)\) be the antecedents labelled in of the arguments pro and con some choice \(x\) in a dialectical graph. Let \(\text{sum}(\text{pro}, x)\) denote the sum of these in pro antecedents, \((p_1 + \ldots + p_n)\), and \(\text{sum}(\text{con}, x)\) denote the sum of in con antecedents, \((c_1 + \ldots + c_n)\). Equations of the form \(x = \text{sum}(\text{pro}, x) - \text{sum}(\text{con}, x)\), called **choice equations** below, constrain the choice itself, and are just syntactic sugar for a preference expression of the form \(x = (p_1 + p_2 + \ldots + p_n) - (c_1 + c_2 + \ldots + c_n)\).

**Definition 13 (Proof Standards).** Let \((P, A, I)\) be a well-formed dialectical graph and \((C, K)\) be an issue in \(I\).

**Scintilla of Evidence.** A choice satisfies this standard if and only if there exists at least one in position which is the antecedent of a pro argument for it in \((P, A, I)\).

**Preponderance of the Evidence.** A choice \(x\) satisfies this standard if the in antecedents of the pro arguments supporting it outweigh the in antecedents of the con arguments against it. To be more precise, recall that the constraints \(K\) map positions to a set of preference expressions representing inequalities over these positions. The choice \(x\) satisfies the preponderance of the evidence standard if and only if the union
of the preference expressions of the in positions in $K$ and the choice equations entails $\text{sum}(\text{pro}, x) > \text{sum}(\text{con}, x)$.

**No Better Alternative.** A choice meets this standard if no other alternative currently has the better arguments. Formally, some position $c_i$ in the set of choices $(c_1, \ldots, c_i, \ldots, c_n)$ meets this standard if and only if for each $c_j$ in $(c_1, \ldots, c_{j-1}, c_{j+1}, \ldots, c_n)$ the union of the preference expressions of the in positions in the constraints $K$, the choice equations and $\{c_i > c_j\}$ is consistent.

**Best Choice.** A choice meets this standard if it is currently better than all its alternatives. Formally, some position $c_i$ in the set of choices $(c_1, \ldots, c_i, \ldots, c_n)$ meets this standard if and only if for each $c_j$ in $(c_1, \ldots, c_{j-1}, c_{j+1}, \ldots, c_n)$ the union of the preference expressions of the in positions of the constraints in $K$ and the choice equations entails $c_i > c_j$.

**Beyond a Reasonable Doubt.** A choice satisfies this standard if and only if the antecedent of every con argument against it is $\text{out}$ and none of the antecedents of the pro arguments in favor of its alternatives is $\text{in}$.

We leave open the question for future research about how best to decide which proof standard to apply for some issue. An obvious approach would be to allow the users to argue about this, ideally using the same argumentation framework. The problem is to find and justify a base case, so as to avoid an infinite regress.

Although we do not have a theory telling us which proof standard to apply, intuitively it seems clear that more than one such standard could be appropriate in our Porsche example. For the top level issue about which car to buy, the “no better alternative” or “best choice” standards seem to be appropriate, unless of course this is a brainstorming session, for which the less strict “scintilla of evidence” might be preferable. For the safety issue, about whether or not to accept that Volvos are safe cars, the “preponderance of the evidence” test is probably best. For such “yes or no” issues, where there is only one choice under discussion, the “best choice” or “no better alternative” standards are always trivially satisfied and therefore not particularly useful. The “preponderance of the evidence” test also appears appropriate for the final issue of the example, about whether or not speed is more important than safety.

Given a dialectical graph and a mapping from issues to proof standards, the labelling function is defined recursively using the tree structure of the issues in the graph. All positions which are not at issue are labelled $\text{in}$. Working up from the leaves of each issue tree, the proof standard of each issue is applied to determine the labels of its choices until we reach the root issue of each tree.

**Definition 14 (Dialectical Graph Labelling).** Let $(P, A, I)$ be a well-formed dialectical graph. The labelling function, $\text{label}(T, p)$, maps an issue tree $T$ and a position $p$ to one of $\{\text{in}, \text{out}\}$ and is defined as follows:

1. $\text{label}(T, p) = \text{in}$, if $p$ is not at issue or if it satisfies the burden of proof standard for its issue;
2. $\text{label}(T, p) = \text{out}$, otherwise.

**Example 12.** The positions of the Porsche example are shown labelled in Figure 5. The names of the out positions are shown struck out and the names of the in positions are bold. Recall that all positions that are not at issue are in. Thus all the leaf positions which are not a member of some issue are in. In the example, these are all of the positions except $P_1$, $P_2$, $P_6$ and $P_9$. Turning to the positions at issue, let us assume that the “preponderance of the evidence” standard is used for the two subissues. The wife’s position that Volvos are safe ($P_6$) fails to meet this standard, and is therefore $\text{out}$, because of the uncontested constraint, $P_{11}$, claiming that government
statistics are more to be trusted than advertising. Not only is it not entailed that the sum of the arguments in favor of safety are greater than those against, as required by the preponderance standard, worse still it would be inconsistent with the constraint to even presume this to be the case. Using the same kind of reasoning, the husband’s position regarding speed being more important than safety, $P_0$, is also out.

This leaves the top level issue, about which car to buy. Although the husband is behind in the speed vs. safety debate, his Porsche choice is still preferred to the Volvo. This is because the wife’s position about Volvos being safe is also currently out, as just shown.

The speed argument in favor of Porsche has some (unknown) integer value. This is reduced by some amount by the cost argument, $P_4$. But however many points are lost because of this cost argument, the Volvo suffers even more, because of the “family car” argument and the husband’s (still) uncontested constraint that the family car argument against Volvo is stronger than the cost argument against Porsche. Thus, for the moment, the Porsche meets both the “no better alternative” standard and even the stronger “best choice” standard, whereas the Volvo meets neither.

This result would be somewhat different if the safety argument were still in. In that case both cars would meet the “no better alternative” standard, but neither would meet the stronger “best choice” standard.

The labelling function of Definition 14 is our formal semantics of the acceptability of positions. We have yet to develop an algorithm for computing labels or to do a proper computational complexity analysis of the problem. These are interesting tasks we might try to tackle in future work. Regarding computational complexity, we anticipate that whatever the theoretical worse case complexity may be, the computation in practice will be quite efficient, because of the small, finite number of constraints we expect to be asserted for each issue.

4 Related Work

Formal models of argumentation fall into two main categories: relational (aka “declarative”) and dialectical (aka “procedural”). As its name suggests, the relational model views argumentation as a mathematical relation between some representation of information or knowledge (typically in the form of rules of various kinds, facts and cases) and the conclusions which are “warranted” or “justified” by this information.

The specification of a mathematical relation as such implies little about the process of reasoning, let alone argumentation. The same relation may be used in various ways: deductively to generate theorems from premises and abductively or inductively to generate sets of premises from goal “theorems”.

Although the implementation of a theorem prover may well contain complex flows of data, this is beside the point here. Our concern is not with implementation details, but with the “functional” features of the problem domain, in the software engineering sense. These are, from a software engineering perspective, analysis issues, rather than system design or implementation issues.

Beginning with Dialogue Logic, [Lorenz, 1961, Lorenzen, 1969] many authors have cloaked their proof theories in a dialectical style, with proponents and opponents exchanging arguments trying to defend or defeat some claim. Whatever the advantages of this approach to proof theory, the resulting models of argumentation remain squarely in the relational camp. The only process modeled is the generation or testing
of warranted conclusions from a static representation of undisputed, and within this framework undisputable, premises.

Computational Dialectics is concerned with modeling this larger process of argumentation, as a foundation for the design and implementation of computer systems suitable for supporting argumentation, negotiation and decision-making in groups.

Several abstract “frameworks” for computational dialectics research have been proposed. Our initial proposal [Brewka and Gordon, 1994, Gordon, 1996] consisted of three levels: 1) logic, 2) speech acts and 3) protocols. The logic layer is responsible for formalizing the notions of (necessary) consequence and contradiction. That is, its subject belongs to the field of mathematical logic, as it has become in this century. The speech act layer, which might better be called the “process” layer, defines the notions of state and action, but specialized on processes of disputation. This layer defines the space of possible actions. The protocol layer defines the rights and responsibilities of the participants, i.e. the norms regulating the procedure or “game”.

Most dialectical or dialogical models of argumentation have not distinguished between the process and norm layers. Since these are formal games, arguably there is no need for a norm layer: all of the norms can, in principle, be collapsed into the definitions of the possible actions at the process level. But models which fail to make this distinction are not suitable as a foundation for computer systems which guide, or even regulate, human activity. As experience with “work flow” systems in the Computer-Supported Work (CSCW) community has shown, they lead to rigid, brittle systems which do not give their users the freedom they require for behaving effectively in a changing world. (See the Winograd/Suchman debate in CSCW Journal [Suchman, 1994, Winograd, 1994] for further information.) This flexibility can only be obtained at the price of enabling users to violate the formal norms encoded in the system. Once violations become possible, a model of norms becomes useful, perhaps essential, as a way to specify how rights and responsibilities become modified in the event of violations [Jones and Sergot, 1992].

At the ICAIL-95 (International Conference on AI and Law), Henry Prakken introduced the notion of an argumentation framework, hereafter AF, which adds some useful structure to the process layer [Prakken, 1995]. In Prakken’s view, an AF defines four elements:

1. The concept of an argument.
2. The notion of conflict between arguments.
3. Preference relationships between conflicting arguments.
4. The status of an argument, such as whether it has “won”, “lost”, or left the dispute undecided.

Although there is not yet a formal definition of Argumentation Framework, comparable to the way a consequence relation, for example, can be defined by a set of algebraic properties, Prakken’s informal definition does give us a reasonable starting point for evaluating the adequacy of proposed Argumentation Frameworks. As we hope to have shown in this article, the Zeno Argumentation Framework does satisfy these requirements.

Using this notion of an Argumentation Framework, Prakken proposed another three layer model for dialectical systems, consisting of a logic layer, the Argumentation Framework, and finally a protocol layer. In this model, the process layer of Gordon’s model is moved to within the protocol layer. The idea of a separate argumentation
framework layer is attractive, but we would prefer to preserve the distinction between the process and protocol layer, for the reasons mentioned above.

These considerations lead us to propose splitting the process layer into an argumentation framework and action layers, which yields the following four layer abstract model of dialectical systems: 1. logic, 2. argumentation framework, 3. actions and 4. protocol.

If we think of argumentation in terms of language games, an AF can be viewed metaphorically as the game board or playing field. At any given time during the game, it is the structure which represents the state of play. The action layer defines the operations or moves which are possible on this structure, to change its state. The protocol layer, as before, defines the norms of the game.

To help make these distinctions clearer, consider the game of chess. The argumentation framework is like the playing board. The action layer would define the possible ways of moving chess pieces on the board. Note that it is possible to move a pawn as if it were a queen. (Some have challenged us on this point, but the daughter of the first author, Caroline, does this all the time.) The protocol layer would define the rules of the game, disallowing the movement of pawns in this way.

In the Pleadings Game [Gordon, 1995], Gordon used the nonmonotonic logic of Conditional Entailment [Geffner and Pearl, 1992] as an AF. Like several other nonmonotonic logics [Pollock, 1988, Simari and Loui, 1992], the proof theory of Conditional Entailment is in the argumentation style of Lorenzon’s Dialogue Logic. This approach suffers from two difficulties: conceptual and computational complexity. The conceptual complexity results from the embedding of an argumentation game for the AF within the larger dialectical game. Although from a formal point of view there is nothing objectionable about this, it is surely too confusing and unwieldy for practical mediation systems. The computational complexity is due to the warrant or entailment relation of these nonmonotonic logics, which are not even semi-decidable. Thus no algorithm can exist, let alone an efficient algorithm, for deciding whether a move is permitted at some stage of dialectical game, using this kind of logic as an argumentation framework.

Perhaps the computational complexity issue can be resolved by restricting the object language, at the cost of some expressiveness, or by restricting the entailment relation, by replacing the underlying monotonic consequence relation within something weaker, as was done in the Pleadings Game with the known relation. But these measures do not solve the conceptual complexity problem. On the contrary, they can make the whole system still more complex.

The use of Conditional Entailment as the AF in the Pleadings Game created an additional difficulty. All forms of preference relationships between competing arguments were encoded using specificity. This technique gave specificity higher priority than any other preference criteria, which is not universally correct. A newer law may take precedence over a conflicting more specific but older law, for example. The Zeno Argumentation Framework was designed to overcome these difficulties. Rather than adapting a nonmonotonic logic for use as an AF in a dialectical model, the Zeno AF was designed from scratch for just this purpose. As the AF has a restricted function in the multi-tiered dialectical model, it can be much simpler, both conceptually and computationally, than a nonmonotonic logic.

Considered in isolation, the Zeno AF may look like a relational model of argumentation, since its labelling function defines a relation between dialectical graphs and in positions. But Zeno was designed to meet the requirements for an AF in the multi-layered procedural or dialectical model of argumentation. Proponents of the relational
approach aim to completely model argumentation within a relational framework, without recourse to procedural notions at the system analysis level.

One limitation of the Zeno AF compared to AFs based on nonmonotonic logic is that our AF does not model the process of generating arguments. We do not consider this price to be very great, because in the application scenarios we are interested in the main problem participants face is not the generation of arguments from a common set of premises, but the generation of arguments from multiple, heterogeneous information sources and models, most of which will be completely informal. To use a formal logic to generate arguments, participants would first have to encode their premises in a common formalism. We do not consider this to be a realistic possibility in our intended application domains.

5 Conclusion

The Zeno Argumentation Framework is a formal version of Horst Rittel’s Issue-Based Information System (IBIS) conceptual model of argumentation. In addition to a syntax for dialectical graphs, it includes a “semantic” labelling function supporting a kind of inference.

The concept of a position is modeled as a primitive, atomic element in this model, similar to the way atomic propositions are primitive, undifferentiated elements of propositional logic. But positions should not be confused with propositions. The intuition behind these concepts is quite different. A proposition is a context-independent declarative statement having a truth value. (That we speak of the logical consequences of a set of premises as being “true” only if all of the premises are “true” supports this point.) A position, on the other hand, records the performance of a speech act in a particular thread of discussion. Its meaning is defined by its role in the discussion and is dependent on its location in the argumentation graph. Things about the world which would have to be made explicit to turn the position into a context-independent proposition are left unsaid, because the participants in the discussion make assumptions about shared knowledge and only make explicit points which seem contentious or otherwise relevant for the particular issues being discussed.

For these reasons, we have imposed a strong restriction on the use of positions in arguments: they may only be used once. That is, only a single argumentation link may flow from one position to some other position. We also prohibit cycles. The resulting argumentation graph is thus a forest of trees. To reuse a position in another argument, one can “copy” the position. This makes it possible to argue about the implicit assumptions of the two copies of the position independently. Perhaps assumptions appropriate for one use are inappropriate for the other.

Prakken’s concept of an Argumentation Framework requires some representation of the idea of conflict between arguments. Rittel’s IBIS model includes pro and con arguments, but they have no effect on the status of positions within the model. Zeno extends IBIS with a means to express preferences and compute position labels, overcoming this limitation.

Preference expressions are a particular kind of position, with some internal structure. Arguments about these constraints are supported, in the same way as arguments about any other positions. This may involve arguing about preferences between the positions supporting and opposing the constraint at issue, and so on, to any “level”.

Given a set of preference expressions, it becomes possible to make inferences about the relative quality of alternative proposed solutions of an issue. A set of burden of
proof standards was defined for this purpose. Given a dialectical graph and a mapping from issues to proof standards, the main task of the mediation system is to determine which of the positions of each issue satisfy its standard. The system is nonmonotonic. Further argumentation may cause some positions to now satisfy the selected test, and others to no longer make the grade. Several competing positions can satisfy the proof standard for an issue, even though the users of the system may ultimately have to choose one when deciding the issue.

This ability to qualify positions is the most significant advance over the informal IBIS model. It transforms IBIS from a lifeless method to organize and index information into a playing field for stimulating debate. The interested parties can see immediately whether their positions are currently “winning” or “losing”, given the arguments which have been made so far, motivating them to marshall still better arguments in favor of their positions.

6 Acknowledgments

This work was partially funded by the European Commission for the GeoMed project, IE 2037 D13. We would like to thank Gerhard Brewka, Eckehard Gross, Ronald Loui, Henry Prakken, and Hans Voss for some interesting and helpful debates on the argumentation in general, and on the pros and cons of Zeno system in particular, and thank Sylvia Johnigk for help with editing and typesetting the final version of the paper. This article is a revised version of a paper presented at ICAIL-97 [Gordon and Karacapilidis, 1997].

References


