Computational Dialectics

Thomas F. Gordon

Artificial Intelligence Division
Institute for Applied Information Technology (FIT)
GMD — National Research Center for Information Technology
Sankt Augustin, Germany

Abstract. The central task in practical problem solving is to identify and choose among alterative courses of action. Computer science has failed to provide adequate tools for supporting rational, effective and fair decision-making under the conditions which usually prevail. Especially, computer science has yet to develop models of rational decision-making in groups which adequately take into consideration resource limitations or conflicts of interest and opinion. This paper provides an informal overview of Zeno, a mediating system for supporting discussion, argumentation and decision-making in groups, which explicitly takes these considerations into account. Also, a new subfield of computer science is proposed, "computational dialectics", whose subject matter is computational models of norms for rational discourse. Zeno is a contribution to this field, based on the thesis that rationality can best be understood as theory construction regulated by discourse norms.

Now is the time to shift our view of computers from communications medium to negotiation medium, from knowledge processing to interest processing.

Carl Adam Petri

1 The Rationality Crisis

The central task in practical problem solving is to identify and choose among alternative courses of action. A couple must decide which car to buy. The designers of the Dylan programming language had to decide whether its syntax should be more like Lisp or Algol. Volkswagen must decide whether to manufacture the new "Beetle" shown at a recent international automobile show. The editors of the General Anzeiger had to decide whether to put the story about the burning of a housing complex for asylum seekers on the front page or bury it near the back.The Social Democratic Party had to decide whether or not to include an Autobahn speed limit in their platform for the upcoming election.The German parliament had to choose between Bonn and Berlin to be the capital city of the reunited German state. The United Nations and NATO must decide whether or not to use military force in Bosnia.

The main purpose and promise of computers and information technology is to improve the procedures for making choices of this kind in industry, government, and other kinds of organizations and groups. The improvement may be in effectiveness, efficiency or, when the conflicting interests of multiple parties are involved, fairness.

The different subfields of computer science contribute to this abstract goal in complementary ways. When there is perfect information about a problem, an efficient algorithm or theorem

prover may be used to compute or search for a solution. Large data bases make a wealth of relevant information readily available. Knowledge-based systems are useful for tasks where there is sufficient consensus about the knowledge required and the costs of knowledge acquisition and maintenance can be amortized over the expected life time of the system. High capacity networks and hypermedia technology are making it cheaper and easier to disseminate and access all kinds of information, including text, sound, color graphics and video. So-called “virtual reality” systems and other kinds of computer-simulation make it possible to explore and vividly imagine the likely effects of alternative courses of action. Even applications as banal as word processing, spreadsheets, and electronic mail flourish in the end because of their role in the processing and distributing information to be used in making decisions.

As useful as these technologies have been shown to be, none of them squarely confronts the problem of supporting effective, fair and rational decision making procedures under the conditions which usually prevail. Either they only deal with a part of the problem, such as providing access to relevant information, or they restrict their attention to special problem solving contexts where certain simplifying assumptions, such as perfect information, can be made.

Under what conditions must decisions usually be made? Here are some of the more salient factors:

1. There is both not enough and too much information. For some parts of the problem relevant information which would be useful for making a decision will be missing. For other parts, there will be more information than the persons responsible for making the decision will have time to even retrieve, let alone comprehend.
2. The resources which can be applied to finding a solution are limited. Time, in particular, may be “of the essence”: a solution must be found before the issue becomes moot.
3. The expected value of the known alternative decisions is not high enough to make it cost effective to invest substantial resources in implementing a program, knowledge base, or other kind of elaborate computer model to use in helping make the decision.
4. However much information is available, opinions differ about its truth, relevance or value for deciding the issue.
5. Arguments can and will be made pro and contra each alternative solution.
6. Reasoning is defeasible. Whatever choice seems best at the moment, further information can cause some other alternative to appear preferable.
7. Factual knowledge about how the world functions and its current state is not sufficient for making a decision. Value judgments about ethical, political, legal and aesthetic factors must not only also be taken into consideration, but are the critical issues requiring the most attention.
8. Several persons have a role to play in making the decision and will be affected by it. Conflicts of interest are inevitable; support for negotiation and other procedures for achieving consensus and compromise are required.
9. Finally, the persons responsible for making the decision are not proficient in mathematics, logic or any other formal methods for solving problems.

Again, this is not a worst-case characterization, but rather a fair and realistic description of the conditions under which decisions must usually be made. Increasing awareness and acceptance of this fact, both among the general public and experts in fields such as philosophy, jurisprudence and mathematics, has led many people to cast doubt on the whole enterprise of rationality.

Computer science is built upon a conceptualization of rationality coming increasingly under fire. Preserving a proper role and justification for information technology will depend critically on developing the theory, methods, and applications for assisting individuals and groups to make effective and fair decisions under ordinary circumstances.

On the theoretical front, computer science desperately needs to intensify its dialogue with the humanities, including philosophy, law, history, literature and the arts. Effectiveness and fairness are normative concepts. The natural and engineering sciences provide us models of how the world functions and technology for changing the world in sometimes dramatic ways, but they address only the easier half of the general problem of making rational decisions. Knowing what can be done tells us nothing about what should be done. It is the humanities which provides standards and methods for evaluative judgment.

Regarding methods, the metaphor of an assisting computer system, the guiding idea of the AC research program at GMD, is a useful starting point. The mediator, moderator or arbitrator of a discussion, debate, brain-storming session, or bargaining meeting is a kind of assistant. He or she is not a principal participant in the discussion, but rather has an ancillary function, such as helping to assure that the speakers abide by the rules of the procedure. A mediator has little or no authority. It is neither cop nor judge. The function of a mediating computer system is not to automatically enforce some formal, and therefore rigid, set of procedural rules for resolving conflicts and deciding issues, but rather to advise the participants about the rules and provide other information about the state of the proceeding.

As for applications of this idea, several mediating systems for coordinating the activities of a group have been designed and implemented during the course of the AC program in the Computer Supported Cooperative Work (CSCW) research division of our institute [Kreifelts, et. al., 1991; Kreifelts, et. al., 1993]. These systems help groups with such tasks as scheduling appointments and meetings, creating and monitoring plans, and guiding the flow of forms through an organization.

Zeno will be a mediating system for assisting the more generic task of discovering and choosing among alternative courses of action. The Zeno system will be able to moderate a discussion or debate about any topic between ordinary persons with no particular technical skills in computer science or logic. Our ambition is to develop a practical system for supporting decision-making in groups under ordinary circumstances.

There is a trade-off between ease-of-use and functionality. Supporting deep reasoning requires complex formal logics. Ordinary users cannot be expected to express their positions in formal languages of any kind, and the state of the art of natural language processing has not yet reached the point where the translation to and from a suitable logic can be automated. Finding a good trade-off between ease-of-use and expressiveness which does not require natural language processing is one of the main problems to be addressed by Zeno. We call our current approach *lazy formalization*. The idea is that the participants in a discussion are free to choose the level of formalization they deem appropriate. In fact, a speaker may use any means of expression desired, formal or informal, textual, graphical, or multimedia. The discussion begins using a logic which is so simple that it can be hidden completely behind an intuitive user-interface.

To give a better idea of the kind of system we have in mind, the next section describes Zeno's current design, from the user's perspective. The next section is more theoretical; it discusses a proposal for a new field of computer science research, to be called "computational dialectics", whose subject matter is computational models of norms for rational discourse. This field is founded on a conception of rationality which, we claim, can withstand the criticism and concern of the skeptics. The final section discusses related work.

2 A Tour of the Zeno System

The Zeno system will be configurable for different kinds of deliberations about some topic, such as brainstorming sessions, council or board meetings, contract negotiations, design team discussions, and law suits.There will be two interfaces, one to configure the system and another for using a particular configuration to mediate a proceeding.The first interface can be viewed as a high-level programming language for implementing mediating systems. Compiling a program in this language generates a mediating system for a particular type of proceeding.We will have more to say about this "programmer's interface" later. Let us first take a look at the interface to be used by the persons taking part in a discussion.

Figure 1 shows a mock-up of a Motif version of the main window of the Zeno application. It appears to be a cross between an electronic mail program and a hypermedia browser, and indeed it has characteristics of both.

The "File" menu includes the usual commands for such things as opening, closing and printing documents. A Zeno document contains references to all the messages registered with the mediator for a particular proceeding or task. To open a document, the user must first log in to the mediator's machine on the network, providing his or her name and password. The rights of the user to view or send some message may depend on such factors as the type of the proceeding and the role of the user in this proceeding. Several participants can open and modify the same document simultaneously; as messages are only added during the discussion and never deleted, the usual synchronization problems of databases and multiuser text editors do not appear here. However, the rules of the proceeding may have to specify when each kind of speech act is to be legally effective; e.g., at the time it was sent or at the time of receipt by the mediator.

Instead of the usual "Save" there is a "Send" command. A user can modify the network of claims and arguments locally, playing "what-if" games to see the effects of alternative lines of argument, before sending his contribution back to the moderator. There will be unlimited undo and redo commands as well as a "Revert" command so as to facilitate this kind of private contemplation.

There will also be a "Save As" command for saving a local copy of the document and for exporting it to other file formats. Of particular interest would be the possibility to export an outline of the discussion, or selected parts of it, in the native formats of various word processing, "idea processing" and "presentation" applications. This would be quite useful for writing such things as the "minutes" of the discussion or the justification of a decision.

Just below the menu bar, in the center of the display, is the title of the proceeding, in this case "Miller vs. Smith", suggesting this may be some kind of legal discussion. In the area below the title is a scrollable transcript of the messages which have been registered with the mediator. These need not be all, or even most, of the messages which have been exchanged by the participants in the proceeding. It is not intended that the Zeno system be used to replace all other forms of communication within a group. On the contrary, it should be used primarily for those speech acts which are to have some kind of official or binding character.

This brings us to a problem Zeno, like other CSCW applications, must deal with: How to integrate the system with the other applications, to facilitate interoperability, data exchange and ease-of-use? Presumably most users will already be using some other program for electronic mail. Some may not want to use another system to send messages to a mediated discussion, with yet another set of user-interface conventions and quirks. While a complete solution to this problem will have to await the wide-spread use of distributed object-oriented programming environments, an intermediate approach is possible for the time being. First of all it should be possible to cut and paste data between Zeno messages and other applications, at least for the more popular data formats. Secondly, a simple command language, along the lines of the ones used by network mailing lists, will allow messages to be sent to the mediator using any electronic mail program.

Below the transcript in the Figure 1 is some "header" information about the message being displayed, including a short description of the "claim" being made in the message, the name of the sender, the date and perhaps time the message was sent, and a pointer to the message for which this message is a response. The claim can be any unique title for this message. It need not actually be a declarative sentence, although this might be a good practice. In the example, the message claims "This map shows where Escondido is." and is offered to support the claim of the message contributed earlier by Lynn Bild, who claimed that "Escondido is near San Diego."

To the left of the field naming the previous message is a label showing the type of this message, in this case an argument "pro" the claim of the other message. Although the full set of message types will be defined by the designers of a Zeno application, there may be a few "standard" types, such as:

Agree. Used to agree with or concede some other claim.

Disagree. Used to challenge, question or deny the other claim.

Pro. A claim which, if accepted, tends to make the claim of the prior message more likely or probable.

Con. A claim which, if accepted, tends to make the claim of the prior message less likely or probable.

Alternative. Proposes another solution to the problem, or takes another position with respect to the issue.

Utility. Makes an assertion about some effect or consequence of deciding to accept the claim of the prior message. For example, one could claim that a Porsche is a fast car, or that a Volvo is a safe car.

Relevance. Questions whether the prior message really is of the type asserted. For example, suppose Judy claims that it will rain next Saturday and Joe then argues that this is unlikely, because he has planned a picnic for then. Rather than arguing about whether or not he has in fact made such plans, Judy might prefer to question the relevance of his plans to her prediction.

Refinement. Registers a claim which only becomes an issue if there is a decision to accept the claim of the other message. For example, if it has been decided to buy a Chrysler, this kind of message can be used to propose buying a particular model, such as a Voyager.

Comment. Can be used as a "catch-all" message type when none of the other types available seem appropriate, or when the speaker wants to avoid the formal consequences of some other speech act. In some applications, it may also be permitted to send comments, and perhaps other types of messages, anonymously.

To compose a message, the user selects its type from the "Argue" menu. Another name may be preferable for some applications, so this will be configurable in the Zeno programming environment.

It might be objected that users will not want to take the trouble to label the type of their messages, or would prefer to remain vague or ambiguous about the intended pragmatic effect of some speech act. For example, in a message to the boss criticizing his plan to manufacture horseshoes instead of tires, one might prefer to tactfully couch the warning in the language of admiration and support. A large, bold label of **criticism** might be counterproductive.

This is admittedly a problem, but not an insurmountable one. One can use the innocuous comment label in such cases. Also, as the body of the message is subject to no formal restrictions, the user is encouraged to apply her rhetorical skills, to the best of her ability, here.

However there is a better response to this objection. Performative speech acts are often effective only if they have the proper form, regardless of their intended meaning. In business and government, one often has to say the magic words. There are sound reasons for this formality. The interests of the persons affected by a decision will differ; often they are diametrically opposed. The buyer of some product or service would like the price to be low; the seller would like it to be high. Whenever it is in the interest of one party to speak vaguely, so as to hedge his bets by delaying the determination of the message's performative effect as long as possible, there is probably another party to the transaction with exactly the opposite interest, to have the matter clarified definitely as soon as possible. Consider a letter offering to buy some product from a mail order distributor. The seller would like the assurance that it is indeed a binding offer before sending the goods. The buyer would prefer this question to be decided after the goods have arrived, to be able to inspect them before deciding whether or not to pay. He would like to be able to back out of the deal by arguing that his "offer" was really only an "inquiry". Formal procedures and "bright-line" criteria for categorizing speech acts provide the means to fairly allocate risks and responsibility in such situations. Clear conventions also dramatically reduce the cost of doing business, by avoiding lengthy and expensive conflict resolution procedures, such as law suits.

There is no universally optimal degree of formality, suitable for all kinds of group decision making contexts. In particular, CSCW systems which support *only* informal modes of communication are biassed; they cater to the special interests of only some of the persons affected by the decisions made using the system. The aim in Zeno is remain neutral by providing a configurable environment supporting a wide range of formality. Design choices about this and other aspects of the procedures for making decisions in a group or organization should be made by representatives of the various interests groups affected, through some fair political process.

Once a message has been sent, the rights and obligations of the other participants will change, depending on the rules of the proceeding. For example, in a negotiation, an "offer" message may give some other participant the right to "accept" within 30 days. Or the posting of an issue may require position statements to be made within six months, before the issue comes up for vote. One of the main responsibilities of the mediator is to maintain a calendar and agenda of such tasks. There are commands for displaying these documents in the "View" menu. The calendar shows the schedule of dates and times for various activities. One possible

service of the mediator would be to remind users of deadlines. The agenda is a prioritized list of issues to be resolved, where the criteria used to prioritize issues or tasks will depend on the application.

Returning to Figure 1, below the header information is a scrollable pane for the body of the message. In the example, this is a color map of San Diego. Again, there are no restrictions, in principle, on the kinds of data which may be included in messages. From the perspective of the Zeno's formal logic, each message is a proposition. As always in *formal* logic, the intended meaning or interpretation of the proposition is ignored when deriving consequences and other kinds of formal properties. However, the persons participating in the discussion will of course be quite interested in the meaning of a message, which will presumably play a dominant role in their contemplations about how best to respond.

Propositions in Zeno are situated; they are contextually embedded in a discussion between persons taking place in time. A proposition does not hang in the air, but is stated by a particular person at a particular time. Except of course for the opening proposition, every statement is made in response to some other claim made in the course of a discussion. One must be careful when carrying over arguments and claims made in one context to some other context in this or another discussion. Syntactically equivalent claims in different branches of a discussion are not presumed to be identical.

A message may also be a compound document, consisting of a combination of graphics, text and other objects, including hypertext links to other messages and documents. Unlike the message types discussed above, these hypertext links have no particular semantics for the logic of the Zeno system. They may be used in any way a user sees fit and help to reduce the "rigidity" of Zeno's formal logic.

While we are on the topic of hypertext links: as every message except the first is a response to some other message, they form a tree structure. The "Navigate" menu includes the usual commands, familiar from hypertext systems, for browsing this tree. For example, the "Top" command takes the user to the first message of the proceeding; the "Up" command moves to message responded to by the current message; and the "Next" and "Previous" commands cycle through the other responses at the same level. To move to a lower level, there are submenus for each type of response, such as "Pro", "Con", and "Relevance".

To perhaps belabor the point: this graphical interface provides an intuitive way to express the elements required by a formal logic (propositions and various kinds of relations between them) without requiring the use of some formal syntax.

At the bottom of the window in Figure 1 is the final pane to be discussed; it displays information about the current status of the claim. On the left-hand side there are two sliders, showing the logical status of the claim along two dimensions. The first dimension concerns the *quality* of the position, relative to the other proposed alternatives. Quality is computed using the utility arguments which have been made for each of the alternatives. (See below.) The other dimension concerns the likelihood, probability or feasibility of the position and is computed using the arguments pro and contra which have been made concerning it. Of course, these sliders can not be manipulated by the user, willy-nilly, to set the value of these parameters. Rather, they are continuously computed by the Zeno system, using a combination of theorem proving and constraint satisfaction techniques.

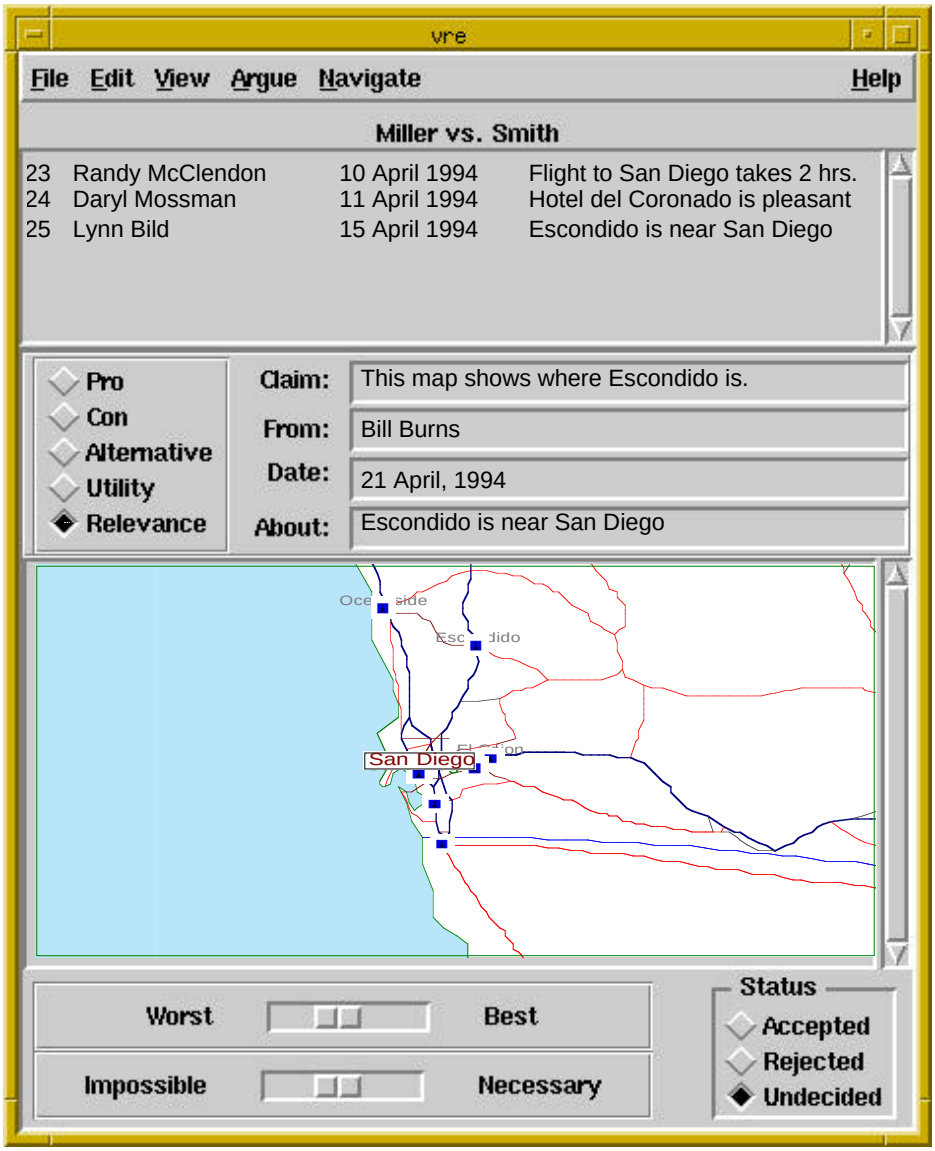

Figure 1. Main window of the Zeno application.

To the right of these two sliders is a group of buttons showing whether the claim has been accepted, rejected or yet to be decided. The procedure for making this decision will depend on the rules of the particular type of proceeding. Common methods include randomly selecting some alternative, using the best possible alternative computed by Zeno, voting, or granting the responsible manager or authority discretion to decide as he or she sees fit. Notice that the quality and probability measures computed by Zeno have only an advisory character; they may be taken into consideration by the persons responsible for making the decision, but need not determine it. This flexibility is perfectly reasonable. After all, the system is founded on the premise that reasoning is defeasible. The responsible person may have information which for various legitimate reasons he is unwilling to divulge to the group and which tips the scales in favor of some other alternative. Or he may simply prefer to follow his intuitions.

Associated with each claim are three other documents: 1) a worksheet for making and viewing claims about the relative weight or importance of the arguments pro and contra this claim; 2) another worksheet summarizing the arguments made about the relative utility of this

claim and its alternatives; and 3) a document for recording information about the decision. This latter document may include such things as the name and "signature" of the person making the decision, the date the decision was made, an explanation or justification of the choice, or a tally of the votes for and against each alternative, as appropriate.

The worksheet for presenting utility arguments is shared by all alternative positions for some issue. It has two parts. The first part is a list of utility claims which have been decided to apply to each alternative position. For example, when discussing which car to buy, the following utility claims may have been accepted

BMW 520i. good chassis, good styling, good interior, good safety, fair fuel economy.

Mazda Xedos. good chassis, fair styling, fair interior, good safety, fair fuel economy.

Opel Omega. good chassis, good styling, good interior, good safety, fair fuel economy.

These claims could be displayed in a table, but this will not generally be the case. A utility claim may be an arbitrary proposition about some effect of choosing the alternative; Factors or dimensions, along which the alternatives would be ranked, need not be first be systematically identified.

The second part of the utility worksheet is a list of "constraints" about the relative values of these utility claims. In the car buying example, the following evaluation constraints may have been accepted:

- good interior > fair interior
- good safety > good interior
- good safety > fair fuel economy
- fair interior + fair fuel economy > good safety

The main purpose of these constraints is to provide an easy, qualitative way to express and argue about preferences and value judgments. It is not necessary here to devise factors and utility functions, let alone assign numeric values to particular properties. Given this qualitative information, constraint satisfaction techniques can be used to rank the quality of the alternative solutions.

In the graphical user interface, there will be some intuitive and quick way to go to the message in which it was decided to accept some claim shown on this worksheet. For example, to find out why it was decided to believe that an Opel Omega has a good chassis, one might be able to just double click on that property on the worksheet to begin browsing any arguments there may have been about the quality of the chassis.This applies to the evaluation constraints as well, which are debatable just like other claims.

It remains to discuss the "programmer's" interface for configuring Zeno for a particular kind of discussion or proceeding. Some of these discussions will be primarily cooperative, others will be more adversarial. Other factors to consider when drafting the rules of the procedure include its goal and purpose, the types of speech acts required, and the roles of the participants. The rules of procedure will specify just what speech acts are permitted, obligatory or forbidden in each situation, and at what time, where a situation consists of the messages which have already been registered with the mediator.

A configuration also needs to specify what the mediator should do in the case of a violation, or attempted violation, of the rules. However complex the rules, situations are likely to occur which were not anticipated. One way to avoid rigidity when configuring Zeno is to use the legal system as a model. Unlike formal systems, legal rules are not self-applying. Persons must interpret, and reinterpret, the rules in the context of their current situation. In the worst

case, a law suit may be necessary to resolve disagreement about the meaning of the rules. In a Zeno application, this strategy could be realized by having the mediator send a private warning to the persons affected, who would then have the option of negotiating a "settlement" or initiating some quasi-legal procedure for resolving the conflict.

The language for defining these rules has yet to be designed. It is still unclear whether a simple and convenient graphical user-interface will be possible for configuring Zeno. Arguably, it is not quite so important for this interface to be easy for lay persons to use, as some small number of configurations will be adequate for most purposes. Experts could be hired to help design and implement a custom configuration. On the other hand, it is critically important that every user be able to understand the rules of theproceeding, so as to be able to effectively participate and decide whether or not others are "playing by the rules".

3 Computational Dialectics

Zeno is but one project in the field we call "computational dialectics". The subject matter of this field is the design and implementation of computer systems which mediate and regulate the flow of messages between agents in distributed systems, so as to facilitate the recognition and achievement of common goals in a rational, effective and fair way.

The term "agent" here is intentionally abstract. An agent may be a person or organization, or some computational entity, such as a process, task or object, in the sense of object-oriented programming. In a complex, distributed system consisting of multiple agents working together, some of the agents will be natural persons or organizations, and others will be artificial agents implemented by programs executing on one or more computers.

The field of computational dialectics has its analytic, empirical and normative aspects. The analytical task is to develop models of the structure of discourse and communication tuned to the task of group problem solving and decision-making. This distinguishes the models of dialectical processes from those designed for understanding natural language. The analytical task, as usual, consists in identifying, categorizing and analyzing the formal properties of these models along various dimensions. The empirical aspect involves developing and testing theories of how, in fact, groups of agents use language to make decisions. Finally, the normative aspect of the field is concerned with drafting and justifying principles and norms for regulating communication and decision-making in groups, where individual agents may have incompatible beliefs about the world and competing interests.

To be sure, much prior work has been done in this area, if not under this label. It is our hope and goal to bring together researchers who have been working implicitly on this subject in the fringe of other parts of computer science, including distributed systems, distributed artificial intelligence, nonmonotonic logic, case-based reasoning, machine learning, conflict resolution in concurrent engineering, artificial intelligence and law, issue-based information systems and computer-supported cooperative work. Presumably, research in computational dialectics would be more productive if the people interested in this subject would begin to form a research community. As a first step in this direction, we have organized, together with Ronald Loui, a workshop on computational dialectics for the Twelfth National Conference on Artificial Intelligence (AAAI-94). Additional work is needed to reach people outside the AI community.

The thesis of the Zeno project, which represents only one position in the field of computational dialectics, is that rationality can best be understood as a theory construction process regulated by discourse norms. The dominant conception of logic in analytical philosophy is limited to the study of the notions of consequence and contradiction given some set of premises. It says nothing about how the premises are or should be constructed.

However, by viewing rational discourse as a process of theory construction, a strong connection to logic is preserved. Our aim is to complement logic with norms regulating the pragmatic aspects of constructing and using theories.

4 Related Work

Prior work of the CSCW group at GMD on coordination systems was mentioned in the introduction [Kreifelts, et. al., 1991; Kreifelts, et. al., 1993]. Again, whereas these systems support the scheduling of meetings, the monitoring of tasks and activities and the flow of forms through an organization, Zeno mediates a discussion about the pros and cons of alternative solutions to a problem.

Several others hypertext systems have been constructed for organizing and browsing arguments, based either on the Conklin's Information-Based Issue Systems (IBIS) model [Conklin and Begeman, 1988] or Toulmin's model of argument structure [Toulmin, 1958; Marshall, 1989; Schuler and Smith, 1990]. The argument structure designed for Zeno is a synthesis of ideas from these systems. Unlike Zeno and the Pleadings Game, discussed below, these other hypertext systems do not use logical dependencies to constrain or facilitate the further development of the discussion. The goal in Zeno is to achieve the simplicity and ease-of-use of IBIS without sacrificing a solid, logical foundation, by drawing on the results of argumentation systems for nonmonotonic logic [Pollock, 1988; Simari and Loui 1992; Geffner and Pearl, 1992]. With the exception of the Pleadings Game, none of these other systems distinguish the roles or interests of the persons involved in the discussion, so the idea of regulating argumentation using discourse norms does not appear.

The Pleadings Game [Gordon, 1993a; Gordon, 1993b] is a computational model of a mediator for a particular kind of legal proceeding, the pleading phase of a civil case. Pleading is a two-party adversarial procedure, whose purpose it is to identify the issues of the case. The plaintiff has the burden of defending his claim against various kinds of attacks by the defendant.

The Zeno system generalizes the Pleadings Game in a number of ways. Whereas the Pleadings Game is a particular mediating system for one kind of decision-making procedure, the goal of Zeno is to provide a convenient language for specifying a broad range of mediating systems, for both cooperative and adversarial contexts. Another important difference is that the Pleadings Game model has an entirely theoretical purpose, to demonstrate how judicial discretion can be fairly and rationally limited by factors other than the literal meaning of legal texts. The purpose of the Zeno system, on the other hand, is to provide a practical tool for implementing systems which mediate actual discussions between persons.

Notwithstanding these differences, both Zeno and the Pleadings Game are based on insights from legal philosophy, especially the normative theories of legal argumentation of H.L.A. Hart [1961] and Robert Alexy [1989].

Hart and Alexy, in turn, both draw heavily on the "speech act" theory of language going back to (late) Wittgenstein. There is an ongoing controversy within CSCW about the suitability of speech-act theory as a basis for computer systems for coordinating human activity in organizations. A recent issue of the CSCW journal includes two articles on this very issue; one by a critic, Lucy Suchman [1994], the other by Terry Winograd [1994], who together with Fernando Flores first introduced the use of speech-act theory to CSCW in the influential "Understanding Computers and Cognition: A New Foundation for Design" [1986].

Legal philosophy provides another perspective on this issue, which reveals weaknesses in the arguments of both Suchman and Winograd.

Suchman takes the position, closely related to Grudin's in [1990], that "the adoption of speech-act theory as a foundation for system design, with its emphasis on the encoding of speakers' intentions into explicit categories, carries with it an agenda of discipline and control over organization members' actions". In other words, she claims this kind of CSCW system furthers the interests of management at the expense of workers. She proposes instead that CSCW systems be designed with "an appreciation for and engagement with the specificity, heterogeneity and practicality of organizational life."

Winograd counters by arguing, in essence, that a certain amount of rigidity and formality is a necessary evil in large organizations: "When people interact face to face on a regular day to day basis, things can be done in a very different way than when an organization is spread over the world, with 10,000 employees and thousands of suppliers". And further, "The use of explicitness makes possible coordination of kinds that could not be effectively carried out without it."

If we identify corporate interests with the interests of management and suppose that these interests conflict with those of employees, then Winograd may be thought to be conceding Suchman's main point here. However, he goes on to argue that coordination systems can be successful only if they are "grounded in the context and experience of those who live in the situation". To assure this is so, Winograd argues that users should participate in the design of the system.

At first glance, there may not appear to be anything new or interesting about this debate from a legal perspective. Surely it is noncontroversial that changes in the rules of an organization, whether or not brought about by the introduction of new technology, have a political dimension requiring fair procedures for negotiating an acceptable compromise balancing the interests of all concerned.

What does make this debate interesting from a legal point of view is its close relationship to an old debate in legal philosophy about the status of legal rules. In the previous century, German conceptualism (Begriffsjurisprudez) adopted a deductive view of legal reasoning. In modern terms, they sought to apply the axiomatic method to the law. The resolution to any conceivable legal dispute was contained, implicitly, in the axioms, waiting to be discovered by a process of deduction. This view depends critically on the "correspondence theory of truth", which underestimates the difficulty of deciding whether the concrete facts of a case should be subsumed under the general terms used in a statute. This is where Hart comes in. Hart recognized that the meaning of laws cannot and should not be fixed at the time of their enactment by a legislature. Rather, the meaning of the law must be continuously reinterpreted and re-evaluated in the context of deciding specific cases, in the courts. Hart noted that the ability of natural language to be imprecise is a feature, not a defect; it allows power to be delegated to the courts to decide issues in the context of concrete cases, when more information is available. This line of reasoning leads to a justification of the division of powers between the legislative and judicial branches of government.

Suchman's main mistake is to conclude that rules framed in terms of general "categories", only serve the interests of a particular class, management. There are at least two problems with this position. The first is that the rights and interests of employees, too, may be protected only by this kind of general rule. The "technology" of the language of laws, rules and agreements is interest neutral. The second problem is that the moral principle of "universalizability" requires norms to be expressed in terms of general categories, rather than concrete situations. This derives from the notion of equality under the law. The tension between equality and doing justice to the "specificity, heterogeneity and practicality of organizational life" is resolved by interpreting and reinterpreting general rules to decide issues raised by concrete cases.

Is the formal structure of speech acts in Winograd's kind of CSCW system like a system of laws? It should be but is not. The problem is that these formal structures have been used to define and create the space of actions, rather than the space of rights and obligations. They have been used to define what is possible, rather than what is ideal. It is not enough to allow users to participate in the design process. Users, too, are not omniscient; they cannot foresee all the possible consequences of an abstract set of norms, divorced from the concrete facts of particular situations. It should be possible to do what is best, and not merely that which is obligatory given a strict, formal interpretation of the rules.

Zeno is modelled after the legal system. The formal rules of a decision-making procedure are not used to limit the space of possibilities. Users remain free to take responsibility for their own actions. They may, at their own risk, violate the formal rules. The mediating system is neither the long arm of the law, nor of management. Its job is to advise users about their rights and obligations, not to enforce the rules. Procedures will be provided for resolving disputes about the meaning of the rules, analogous to court proceedings.

Acknowledgments

Earlier versions of this paper were read by my colleagues Barbara Becker, Gerhard Brewka, Peter Hoschka and Hans Voss. I would like to thank them all here for their constructive suggestions and encouragement.

References

[Alexy, 1989] R. Alexy, *A Theory of Legal Argumentation*, Claredon Press, Oxford, 1989.

[Conklin and Begeman, 1988] J. Conklin and M.Begeman. gIBIS: A Hypertext Tool for Exploratory Policy Discussion. *ACM Transactions on Office Information Systems*, 6(4):303–331, 1988.

[Geffner and Pearl, 1992] H. Geffner and J. Pearl. Conditional Entailment: Bridging Two Approaches to Default Reasoning. *Artificial Ingelligence* , 53(2–3):209–244, 1992.

[Gordon, 1993a] T. F. Gordon.The Pleadings Game: Formalizing Procedural Justice, In Proceedings of the Fourth International Conference on Artificial Intelligence *und Law*, pages 10–19, ACM Press, 1993.

[Gordon, 1993b] T. F. Gordon. *The Pleadings Game: An Artificial Intelligence Model of Procedural Justice*. Ph.D. diss., Fachbereich Informatik, Technische Hochschule Darmstadt, 1993.

[Grudin, 1990] J. Grudin. Groupware and Cooperative Work: Problems and Prospects. In B. Laurel, editor, *The Art of Human Computer Interface Design*, pages 171-175, Addison-Wesley, 1990.

[Hart, 1961] H. L. A. Hart, *The Concept of Law*, Oxford University Press, 1961.

[Kreifelts et al., 1991] T. Kreifelts, E. Hinrichs, K. H. Klein, P. Seuffert, and G. Woetzel. Experiences with the DOMINO Office Procedure System. In *Proceedings of the Second European Conference on Computer Supported Cooperative Work*, pages 117–130, Kluwer, Dordrecht, 1991.

[Kreifelts et al., 1993] T. Kreifelts, E. Hinrichs, and G. Woetzel. Sharing To-Do Lists with a Distributed Task Manager. In *Proceedings of the Third European Conference on Computer-Supported Cooperative Work*, pages 31–46, Kluwer, Dordrecht, 1993.

[Marshall, 1989] C. C. Marshall. Representing the Sructure of Legal Argument. In *Proceedings of the Second International Conference on Artificial Intelligence and Law*, pages 121–127, Vancouver, ACM Press, 1989.

[Pollock, 1988] J. Pollock. Defeasible Reasoning, *Cognitive Science*, 11:481-518, 1988.

[Simari and Loui, 1992] G. R. Simari and R. P. Loui. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence*, 53(2-3):125–157, 1992.

[Schuler and Smith, 1990] W. Schuler and J. B. Smith. Author's Argumentation Assistant (AAA): A Hypertext-Based Authoring Tool for Argumentative Texts. In A. Rizk, N. Streitz and J. Andre, editors, *Hypertext: Concepts, Systems and Applications*, Cambridge University Press, 1990.

[Suchman, 1994] L. Suchman. Do Categories Have Politics? The Language/Action Perspective Reconsidered. *Computer-Supported Cooperative Work*, 2:177–190, 1994.

[Toulmin, 1958] S. E. Toulmin. *The Uses of Argument*. Cambridge University Press, 1958.

[Winograd and Flores, 1986] T. Winograd and F. Flores. *Understanding Computers and Cognition, A New Foundation for Design*, Ablex, Norwood, New Jersey, 1986.

[Winograd, 1994] T. Winograd. Categories, Disciplines, and Social Coordination. *Computer-Supported Cooperative Work*, 2:191–197, 1994.