Negotiation Behavior During Multiple Agent Specification: A Need for Automated Conflict Resolution

William N. Robinson
CIS-TR-89-13
September 6, 1989

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE
UNIVERSITY OF OREGON

Negotiation Behavior During Multiple Agent Specification: A Need for Automated Conflict Resolution

William N. Robinson

**Department of Computer and Information Science,
University of Oregon, Eugene, OR, 97403, U.S.A.**

ABSTRACT

Negotiation is part of specification. During specification acquisition, users negotiate amongst themselves and with analysts. During specification design, designers negotiate amongst themselves and with a project leader. Throughout the specification of a system, people communicate needs and constraints in ways which benefit the project or themselves. The study of such behavior has been valuable to disciplines such as contract negotiation and office management. However, software engineering has yet to address negotiation behavior. Here we show how negotiation applies to specification. We present automated means to promote integrative behavior during specification. We conclude that formal models of users' desires and resolution methods are necessary for integrative reasoning.

1. Introduction

Our research is focused on modeling and automating the software specification process. We have found this requires expanding the concerns of software analysts to include heretofore ignored issues. Multi-agent specification design is one such issue.

This paper reports on our work concerned with multi-agent specification design. We describe how various agents, often with conflicting goals, can resolve their differences, integrate their results, and produce a unified specification. Such bargaining behavior is both ubiquitous in complex specification and unrepresented by current methods. The following example will help illustrate this.

1.1. An Example

Consider the following transcript of an initial dialogue between a systems analyst (A), a manager (M), and a user (U) of a university admissions system.

79:U	As it is now, the variety at least, I get to take my eyes away from the computer. I'm spending half of my time writing letters on the computer, and to spend the other half putting information into the computer, ...
413:U	Is the purpose to eliminate the card file?
414:A	Yes. Yes.
415:U	Then I'll just restate my original objection...
416:A	Right.
417:U	...of sitting in front of the computer all day and entering this information is exceedingly boring...
418:A	Yeah.
419:U	...and very hard on the eyes...
420:A	Right. Well, for one thing you wouldn't be entering the information. And I admit that there's no way around looking at a computer screen if this is automated. So, if that's real distasteful to you, then that could be a problem.
421:M	Weren't you talking about having the computer print cards with all the information...
422:A	That's true.
423:M	Or, just generate a hard copy so that you have some backup.
424:A	That's true, but I guess we'd have to think about it more, it seems like the best reasons for doing this—ah, that's true, maybe a card file would be useful. It seems to me that a person that likes to use computers, one of the best reasons to have this automated is because it's a lot easier to get access to information rather than looking through the log file or card file.
425:U	I think I find it personally easier to pull out a card file, to get the name alphabetically than it is to punch it into the computer and wait for it to come...
426:A	Yeah, right.

The analyst, A, supports typical goals of an MIS system, e.g., automated filing. However, U has other objectives. She desires to work away from a computer at a variety of tasks. Perceiving the specified system as conflicting with her objectives, she negotiates. At issue is the amount of computing time in which U must engage. She attempts to reduce her computer time by stressing the need for a card file. The analyst gives in. Interactions 424-426 exhibit the compromise: a card file will complement the automated system.

The above interaction was typical in acquisition protocols we conducted[16]. Conflicts in the protocols arose mainly from the pursuit of goals, not because of errors. Such conversations are analogous to contract negotiations: parties interact to gain a mutually beneficial outcome. Unlike contract negotiation, specification negotiation is rarely investigated or even acknowledged. Here we give a basic account of negotiation behavior and present a design tool, **Oz**, which uses integrative reasoning. **Oz** provides ways to represent conflicting perspectives, like those above, and automated means to produce resolutions.

Oz assists a group of analysts engaged in the design of a specification. It embodies our multi-perspective specification design (MPSD) method. Simply put, MPSD suggests that a perspective be created to represent the desires of each type of person associated with a proposed system. Next, specifications are created expressing ideal systems from the view of each perspective. Finally, the specifications are integrated. Conflicts which arise during integration are resolved using negotiation techniques. Our model of integrative behavior, called *integrative reasoning*, conducts this process semi-automatically.

Integrative behavior, negotiation, and relevant decision science ideas are presented in section 3. **Oz** is discussed in section 4. As illustrated later, its use of integrative reasoning and a formal requirements model allows **Oz** to aid negotiations like that in the above protocol. Finally, section 5 presents conclusions. Next, section 2 summarizes research on specification negotiation.

2. Specification as Negotiation

Complex specification involves negotiation amongst a group of analysts. In many cases, one individual stands above the rest with skills of: interdisciplinary information integration, communication, and motivation. In one study, such individuals had knowledge which "...allowed them to integrate different, sometimes competing, perspectives on the development process."—p.1271[5]. Such analysts serve the role of arbitrator or mediator of conflicts. But, while they do apply negotiation knowledge, it is always done so without tool support. Automated negotiation support is needed. Analysts have "...lamented having no tools for capturing issues and tracking their status...Failure to resolve issues frequently did not become obvious until integration testing."—p. 1278[5]. Our research is aimed at filling this void.

Design tools employing negotiation techniques provide for the integration of knowledge. They do so by representing variant knowledge sources (e.g., user perspectives) and reconciling conflicts that arise during their integration. Next, we will show how current research addresses these issues.

2.1. Current Research

Until now, very little research has directly addressed the presence of negotiation behavior in specification. Even those that have, haven't addressed negotiation subprocesses[3, 17]. However, negotiation is generally recognized to exist; Ross typifies its early treatment.

> To succeed, the task of the analysis must be properly managed and coordinated, and the requirements definition effort must embody multiple viewpoints. These viewpoints may be overlapping and, occasionally contradictory—p. 10[37].

Later, Scacchi elevated the importance of negotiations.

> Problems found in specifications may be due to oversights in their preparation or conflicts between participants over how they believe the system should function...Each of these questions point to tacit or explicit negotiations between participants that must occur in the course of getting system specifications developed. Subsequently, the outcome of these negotiations will shape how the specifications will be.—p. 54[39].

Recently, a survey of large systems design concluded, "...developing large software systems must be treated, at least in part, as a learning, communication, and negotiation process."—p. 1282[5]. However, only now are empirical and modeling studies of specification concerned with negotiation.

Bendifallah and Scacchi[3] observed five student teams building similar software specifications over a ten day period. As part of their analysis, they hypothesized six categories of specification behavior. Three of these categories are particularly relevant: separating a problem into sub-tasks, resolving conflicts, and integrating results. This is encouraging and supportive of the MPSD method. However, we require a more formal and detailed representation than that given by Bendifallah and Scacchi to build specification design tools.

Finkelstein and Fuks view specification design as a multi-party negotiation problem[17]. They have developed a formal model of negotiation dialogue. Constrained by dialogue rules, knowledge sources remove inconsistent beliefs through communication. This view of specification as a multi-agent communication task is encouraging. However, such protocol oriented models speak to only a narrow aspect of negotiation. They avoid identifying conflicts, representing conflicts, and generating resolutions; these are basic concepts of *integrative reasoning*.

Our research concerning negotiation originated with an automated assistant, **Oz**, which embodied Feather's *parallel elaboration* specification methodology[33]. Simply stated, his methodology calls for independent development of separable functional specification aspects[12, 13]. With it, independently developed designs are not constrained to have consistent interfaces. While this simplifies design, it complicates design integration. In our approach, integration is assisted through negotiation techniques[11, 34-36].

We view specification design as an interplay between the acquisition of user goals and objectives and their representation in a specification language. The first process entails formalizing what users want to achieve and maximize; the second entails formalizing how their needs can be met.[1] Only through operational representations can goals be *discovered* to interact.[2] Hence, as the specification is created, it will become apparent that some goals and objectives have representations which interfere while others do not. Such interactions drive designers to seek alternative representations, relax goals, and drop goals and objectives. This is conflict management, a process of negotiation.

Conflict management and other negotiation processes have received little attention from SE researchers; AI researchers have explored only slightly more. Negotiation can be characterized as a multi-agent planning task. Each agent has his own set of goals. To achieve those goals, he must consider the actions of others. Research has focused mainly on multi-agent problem solving architectures[9, 10, 19, 20, 24] or multi-agent communication protocols[4, 6, 41]. Fewer projects follow the tradition of single-agent planners[38, 42, 46]. Those that do develop conflict resolution knowledge[23, 25, 29, 43, 44]. We too are focused on conflict resolution and management. We apply negotiation knowledge to design. That knowledge is presented next. Section 4 discusses its use in **Oz** which has been modified to assist integrative behavior.

3. Negotiation

Negotiation is a large field containing many schools of thought. Our presentation focuses on a search method used in integrative bargaining. Furthermore, we reject the use of utility functions for integrative reasoning. We explain why using normative and empirical decision science theories.

There are two basic types of bargaining: *distributive* and *integrative*[32]. Distributive bargaining reflects the intuition of, "your loss is my gain," i.e., a constant sum game. It typically involves only one issue. Each party describes a *utility curve* and a *reservation value* for each issue. A utility curve can be represented as a function $U_a(x)=y$, where y represents the satisfaction, 0 to 100%, that a party derives from the attainment of value x for attribute a. Often, opposing parties' utility functions are inversely related. However, even such negotiations can be resolved if there is a zone of agreement between reservation values. A *reservation value* is the lowest value of an issue a party will accept; any value lower and they will break off negotiations. A *zone of agreement* is the range of values between the reservation values of all parties which

[1]Such specifications do not (generally) include software design or implementation decisions, but describe relations of the system and its environment.

[2]Goal interaction descriptions, such as probabilities of competition[7] or interaction-resolution pairs[43], are useful. But, they combinatorically increase with the number of goals[29] and depend on ever changing technology[21, 47].

is acceptable by all parties.

Consider a buyer and seller of a product. If the buyer is willing to pay \$100 and the seller is willing to accept \$50, then the zone of agreement is the range of payments between \$50 and \$100. However, if the buyer's reservation value is \$50 and the seller's reservation value is \$100, then no distributive agreement is likely (except by coercion).

The distributive model can be generalized to m different attributes and n different parties. The total satisfaction that a party, p, derives from m attributes given various values for each attribute is often defined as, $U^p = \Sigma^m_{a=0} U_a$[22]. Then one simply engages in m different independent negotiations. However, when involved in such negotiations, parties exhibit *quid pro quo* behavior among issues. Such exchanges are mainly due to the varying importance of issues to each party; e.g., a buyer may weigh price as most important, whereas, the seller may weigh cash payment as most important.

When weights are taken into account, the distributive nature of negotiations diminishes. A situation in which it is possible for all parties to gain from varying values at issue is characterized as *integrative bargaining*[31]. The aggregate utility function, U^p, can be extended to consider *relative issue weights*, W_a, where, given all m issues $\Sigma^m_{a=0} W_a = 1$. Then, each party's aggregate utility is $U^p = \Sigma^m_{a=0} W_a U_a$.

Next, we present behaviors which facilitate integrative bargaining. We will not consider strategic posturing by individual parties[32], but instead focus on the search by an arbitrator for the "best" resolution.

3.1. Integrative Behavior

Productive integrative bargaining is more likely to occur in negotiations involving cooperative parties employing a *strategy of flexible rigidity*[31]. This strategy consists of incorporation, information exchange, and search. *Incorporation* is the act of augmenting a proposal with some element of an opposing, previously made, proposal.[3] *Information exchanges* are communications which provide insight into another party's motivational structure (goals, objectives, values, and constraints).[4] *Search* involves jointly considering a variety of proposals. Parties present them based entirely on their own perspective with little consideration as to why others favored or rejected previous proposals. Each of these individual aspects, and the strategy of flexible rigidity as a whole, have substantial empirical support[31].

We are assuming a single arbitrator of conflicts in **Oz**. Hence, some integrative behavior concerning how a group of individuals reach a resolution is not directly relevant. But, the content of information exchanges is particularly relevant because it points to structures that must be considered by an arbitrator.

It has been suggested that integrative agreements can only be achieved by communication of *real needs*[45]. Studies conducted or reviewed by Pruitt concur[31]. In most situations negotiations benefit from the exchange of priority and numeric information, be it explicit or implicit. More detailed information exchange involving

[3] A *proposal* consists of values for each issue involved in negotiations.

[4] *Goals* express the desire to obtain a particular value of a discrete or continuous attribute. *Objectives* express the direction in which attribute values are preferred to be maximized (or minimized). Such information forms *criteria* for the evaluation of a proposal.

"goals, values, and priorities is theoretically capable of transmitting rich information from which integrative formulas can be devised. However our data suggest that it has limitations..."—p. 171[31]. Pruitt postulates these difficulties are due to: parties uncertain of their own motives, lack of trust by listeners, and poor interpretation of statements by listeners.

Pruitt supports our belief that numerical and priority information, like relative issue weights, must be part of an integrative reasoning system. Moreover, Pruitt's postulate concerning the richness of motivational structures mirrors results derived from our own computational model[35, 36]. However, utility curves should not be considered an abstraction of integrative behavior. While such devices have been used by negotiators and computational models alike[22, 43], better abstractions exist[47].

Utility theory is only relevant when people maintain consistency in their preference of alternatives; utility functions must be transitive—this is a direct result of the *additivity assumption*, i.e., the summing of weights[1]. There is evidence to the contrary; people display intransitive preferences[1] and are poor at combining the relative strengths of different attributes in a consistent manner[40]. Zeleny's theory explains this intransitivity as changes in a displaced ideal which causes reordering of preferences during the exploration of extreme alternatives[47]. Similarly, integrative reasoning models must contend with combining criteria and intransitive preferences to effectively search for resolutions.

Arrow's axiom, influential and controversial in decision theory, states that *only feasible alternatives* have influence on a rational decision[2]; infeasible ideal resolutions should not be considered during search. However, while attempting to verify Arrow's axiom experimentally, Festinger and Walster obtained evidence to the contrary[14]. Zeleny bases his theory of the *displaced ideal* on such empirical studies, as well as normative decision science theories[47].

The theory of the displaced ideal defines a search space of resolution alternatives and a goal to obtain. The goal is the (infeasible) composite of each attribute's maximum achievable value within the feasible alternatives, i.e., the best known value for every issue without any of the negative interactions.

Figure 1 illustrates Zeleny's ideal using interactions 79-420 from § 1.1. From the protocol, we infer that U desires (1) *only a card file and no computer.* But, A desires (2) *only a computer and no card file.* The composite infeasible ideal, depicted x^*, includes both statements 1 and 2. Through search, the infeasible ideal x^*, is displaced in favor of the feasible ideal, x^{**}. The three remaining points are other feasible alternatives. Together, the feasible alternatives circumscribe the space of possible, yet undescribed, alternatives. The shaded region represents compromise alternatives. Such alternatives are *nondominated* since no alternative is between them and the ideal, x^*. Despite the unavailability of the ideal, it is useful to describe.

The search for the displace ideal is based both on moving away from an anti-ideal (the worst of all issues) and moving toward the ideal.

> As all alternatives are compared with the ideal, those farthest away are removed from further consideration. There are many important consequences of such partial decisions. First, whenever an alternative is removed from consideration there could be a shift in a maximum attainable score to the next lower feasible level. Thus, the ideal alternative can be displaced *closer* to the feasible set. Similarly, addition of a new alternative could displace the ideal *farther away* by raising the attainable levels of attributes. Such displacements induce changes in evaluations, attribute importance, and ultimately in the preference ordering of the remaining alternatives. — p. 143[47].

The strategy of combining extreme proposals assists in: (1) exploring alternatives, (2) ranking alternatives, (3) deriving maximally feasible solutions, and (4) predicting the existence of ideal solutions[36]. This last case is of particular importance; the very description of the ideal alternative may point to its existence or suggest its achievement; cf. Zwicky's morphological analysis[21] or Lenat's discovery mechanism[26-28].

In sum, integrative behavior consists of incorporation of opposing proposals; communication of goals, objectives, values, and constraints; search through extreme alternatives; intransitive preferences; and multiple issues. These complex interacting behaviors result in high profits when individual participants are stubborn in their aspirations and seek innovative alternatives. If participants seek an expedient resolution by abandoning goals early, a correspondingly lower satisfaction will be achieved.

4. Multiple Perspective Specification Design

We have formed abstractions of integrative behavior, collectively called *integrative reasoning*. They include motive and conflict representations, and algorithms to generate resolutions. The following subsections will illustrate these components using the protocol of § 1.1 as an example.

In **Oz**, specification design consists of creating perspectives (§ 4.1), creating specifications (§ 4.2), and integrating specifications (§ 4.3). Generally, perspectives represent the motives of people, e.g., users, managers. Specifications are created which fulfill the needs described in each perspective. Next, the specifications are integrated. Conflicts which arise are resolved using integrative reasoning. The final specification is the result of integrative negotiations between the various perspectives; it is rationalized by design and negotiation records taken throughout the specification process.

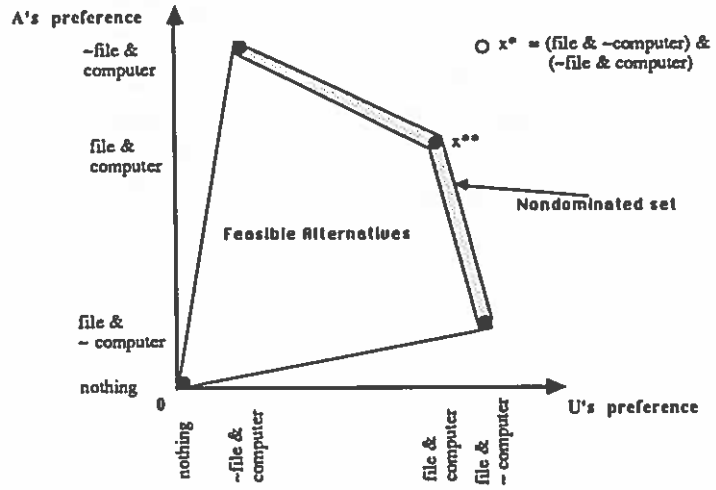

Figure 1. Searching for an Ideal Retrieval Alternative.

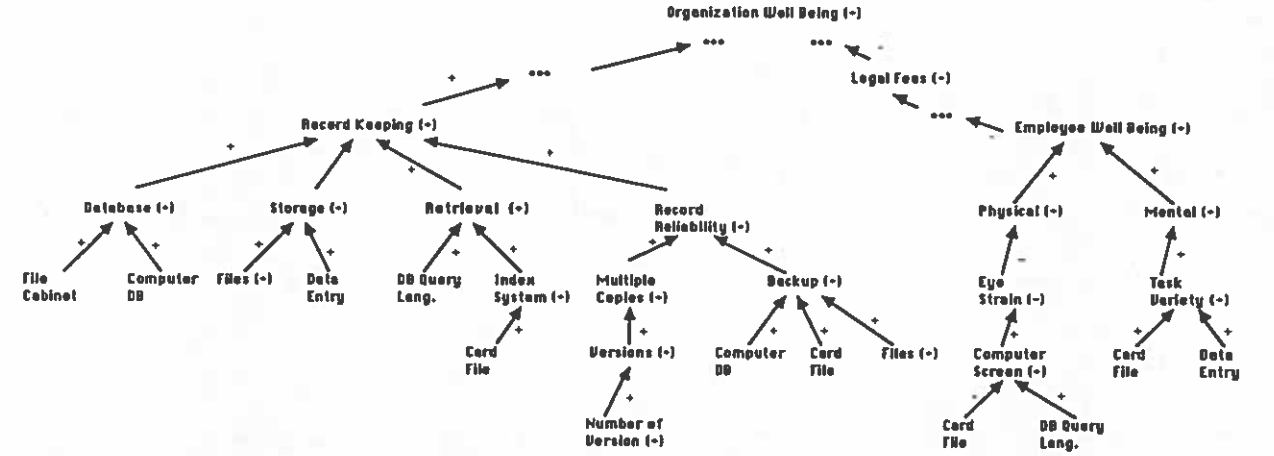

Figure 2. A Portion of an Information Management Attribute Graph.

4.1. A Domain Model

To represent integrative behavior, one must allow for variant belief sets. In **Oz**, *perspectives* fill this role. Perspectives are not projected views of a consistent belief set, but are alternative and conflicting beliefs; they represent a portion of an individual's beliefs, called *motives*. Perspectives are created by marking a domain model with goals and objectives. As such, they represent formal system requirements. Throughout design, integrative reasoning will cause them to be altered, thus reflecting the situation dependent nature of motives[14, 47].

A domain model represents causal relationships. Figure 2 illustrates a domain model for an information management system. It appears to be a goal hierarchy. However, it differs in the representation of "goals", hence the term *attribute*, and it differs in the representation of relations.

Domain attributes are descriptors of behavior ranges found in a domain; for example, Number of Versions has a behavior range from 0 to infinity. Behavior ranges need not be mutually exclusive, as in Retrieval's behavior range; it lists behaviors which can be aggregated to increase retrieval capabilities.

A domain attribute becomes a *domain goal* when behaviors in its range are marked for achievement; it becomes a *domain objective* when just the direction of achievement is given.[5] For example, the maximization of Number of Versions is an objective. (Only ranges whose elements form a partial order can be marked as an objective.)

There are many "+" and "-" signs strewn throughout figure 2. One sign category is associated with attributes via parenthesis. These signs indicate objective directions (e.g., "+" for maximize) and vary across perspectives. For example, from the analyst's perspective Computer Screen (times in which employees view a monitor)

[5]Ranges are one dimensional, discrete or continuous.

should be increased. However, if U's perspective were depicted, Computer Screen would be a minimizing objective.

The other category of signs is associated with links. Attributes are linked to other attributes with which they directly compete (depicted with a "-") or compensate (depicted with a "+"). For example, the link between Physical and Eye Strain represents the negative causal effect eye strain has on physical health. Similar correlation associations have been employed by Deutsch[7]. They indicate *a priori* knowledge of goal interactions. Relative issue weights (§ 3) are also associated with links (not depicted). They indicate the relative effect sibling attributes have on the support (detraction) of a parent. Such links form a tangled lattice among attributes.

4.2. Specification Design

Mapping from a perspective to a specification is achieved by applying operators which match on domain attributes and create specification components; each application can *incorporate* one or more goals and objectives. Next, we model the transcript from § 1.1. to illustrate the use of perspectives.

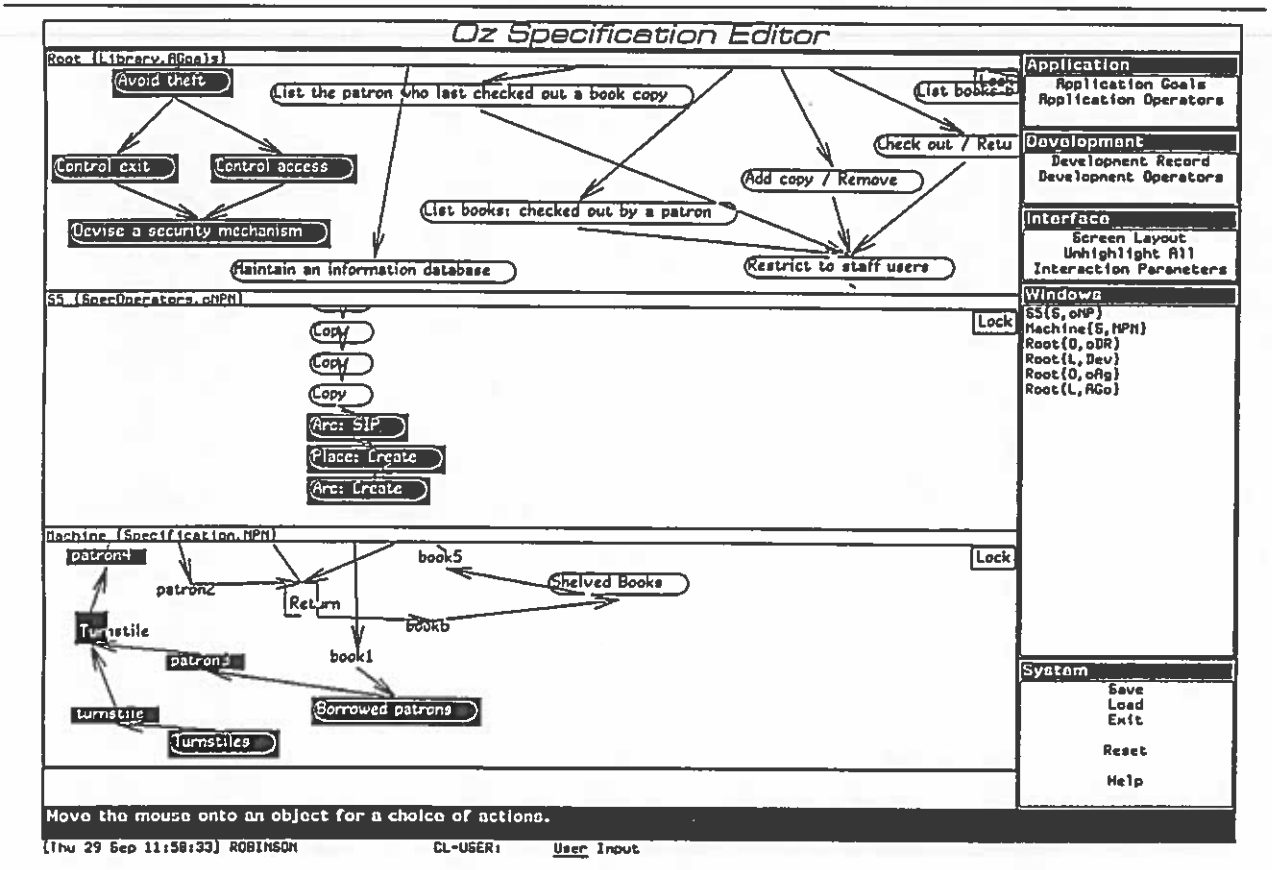

Figure 3. An Oz Depiction of Incorporation Links

From the transcript we infer that U desires a card file. Besides record retrieval, apparently it reduces U's physical and mental stress. Her motives are represented with a perspective where Card File is a goal and Eye Stain and Task Variety are objectives to be minimized and maximized, respectively. Similarly, we create a perspective for the motives of the analyst. Less importance is placed on the substructures of Employee Well Being. However, attributes supporting automation are emphasized, e.g., Data Entry, DB Query Language, and Backup. Specifications are constructed for both A's and U's perspectives. Specification U_{spec} represents a manual file and card index system. The other specification, A_{spec}, represents a database management system. In both cases, the design operators are recorded.

Operator records chronicle the incorporation of perspectives into a specification. These *incorporation links* can be traced upwards to find domain justification for specification components. Attempting to trace downward from an important attribute may determine that the specification does not describe the desired behavior. With incorporation links one can answer the following queries:

- Which domain attributes are *supported* by specification components?
- Which specification components are *justified* by domain attributes?

Such queries are generated during explanation, critiquing[15], and integration[35].

Figure 3 shows how **Oz** can highlight incorporation links. The top pane contains domain attributes, the middle pane design operators, and the bottom pane specification components.[6] The highlighted operators incorporated the highlighted attributes by creating the highlighted specification components. Thus, specification components are tied to the attributes which led to their creation.

4.3. Integration

Once each perspective has a specification, integration can begin. Specifications can be combined via a series of 2-way integrations, or a single N-way integration. In any case, the same four step algorithm applies.

Step 1: Correspondence Identification

Components from different specifications must be identified as similar or different. From our example, components from U_{spec} and A_{spec} are compared. Some components are equivalent, e.g., users, records; others are not. Since this is a difficult problem involving aspects of concept learning[8] and graph isomorphism[18], we rely on the analyst to carry out this process.[7]

Step 2: Conflict Detection & Characterization

Next, conflict detection derives differences from components that are marked as equivalent in step 1. One difference between the file retrieval process found in both U_{spec} and A_{spec} is the means of file lookup; A_{spec} employs a computer index, whereas U_{spec} employs a card file. To characterize conflicts, incorporation links from conflicting components are traced to their attributes. A Most Specific Common Attribute (MSCA) is then found by tracing up attribute links. MSCA's become issues for negotiation. Retrieval is a MSCA of the U_{spec} and A_{spec} integra-

[6]The specification language is an extension of Petri nets[30].

[7]We do provide a tool which creates correspondences between components with the same name. The analyst can edit this structure.

tion.

Step 3: Conflict Resolution

Conflict resolution attempts to remove conflicts between specification components by appealing to compromises or substitutions within perspectives. We combine two heuristic methods with an analytic method[36]. Zeleny's multi-criteria simplex method has been adapted to generate resolutions from N attributes. Searching through combinations of attribute N-tuples, plus resolutions created by the following heuristic methods, it returns the highest ranked resolution.

Our first heuristic method attempts to dissolve conflicts. It does so by applying general *Dissolution Heuristics*. For example, one resolution for resource contention is to time or space multiplex contenders. A heuristic in this category dissolved the retrieval conflict the same way as the analyst: both a card file and a query language can be used to access files.[8]

Our second heuristic method brings in new issues for negotiation. When conflicts can't be dissolved, one or more "losing" perspectives must drop goals so that goals from "winning" perspectives are satisfied. Losing perspectives are compensated by greater satisfaction of related attributes; e.g., providing U with greater Task Variety. Such attributes are determined by a *Search for Compensation*, derived from Sycara[44]. It traces through ancestors of attributes involved in conflict to determine which attributes to compensate.

Pruitt has defined three basic types of compensation: specific, homologous, and substitution. These run the gamut from compensation directed to the specific needs blocked in a conflict to general compensation unrelated to the original needs expressed in the conflict. Searching ancestors of conflicting attributes is one way to derive compensations in order of increasing generality. This is a good strategy since general substitute compensations are difficult to accept and require greater satisfaction than direct compensation[31].

Step 4: Resolution Implementation

Once the conflicts have been resolved at the attribute level, their resolution must be mapped back to the specification level. We assume that difficult interaction problems have been resolved and the remaining step involves simple merging and patching of specification components.

Generating resolutions is the core of the integration problem. It cannot be solved by communication protocols or problem solving architectures. To solve it one must use negotiation knowledge—knowledge providing conflict dissolution, compromise, compensation, and resolution evaluation.

We have partially automated these methods in our experimental design tool[35]. Besides providing negotiation support, **Oz** tracks the status of attributes. During design, attributes can be supported or unsupported (§ 4.2). Furthermore, one may explain the rationale for system features based on the negotiations which created them. For example, one could explain that the admission system's card file was provided to add variety to U's workday and to provide an index backup. If U were replaced or another backup mechanism available, negotiations (design) could be profitably reopened.

---

[8]The conflict, as expressed in the specification differences, was due to U's inability to simultanenous access files using the card file and the computer; time multiplexing U dissolved it.

4.4. Discussion

Our integrative reasoning methods assist designers engaging in integrative behavior. The domain model captures attribute relationships. Perspectives capture the motives of participants. The integration algorithm models processes of conflict recognition, search, dissolution, compromise, and compensation.

The modeling of perspectives is particularly compelling. First, it suggests how specification design can be decomposed. Second, methods employing perspectives are more likely to generate superior and novel designs, than those that do not; this is a corollary of Zeleny's theory concerning extreme alternatives. Finally, multiple perspectives provide the opportunity to explicitly address the interacting goals that surround any design.

While **Oz** does have many integrative techniques, it does not model all integrative behavior. Negotiators do use many of the same techniques; but, negotiators have richer domain representations, broader knowledge, and experimental knowledge[43]. Nevertheless, even **Oz**'s limited knowledge usefully supports the expanding concerns of software analysts.

5. Conclusions

People involved in the specification process negotiate. Design methods now available to analysts ignore such behavior. As a result, conflicts are resolved with poor resolutions: goals are relaxed too far, or even dropped; even good designs are not rationalized. Incorporating negotiation ideas into design methods will alleviate these problems.

Negotiators understand the basis of conflict management: *means* are separate from *ends*; issues define an *initial* search space of resolutions. Designers employing integrative reasoning can concentrate on design. Conflicts can be resolved effectively. As a result, designs will better reflect the desires that lead to their development. Moreover, when designs must be reopened, their design and negotiation records can be used to reconsider past rationale.

Specification can benefit from the use of negotiation techniques. Our abstraction of integrative behavior, *integrative reasoning*, brings assisted negotiation to design. Its components include formal users' perspectives and resolution methods. With it, we have been able to model specification protocols and generate resolutions.

ACKNOWLEDGEMENT

Thanks to John S. Anderson and Stephan Fickas for their comments on previous drafts of this paper. Also, thanks to Dave Meyer and Mary Mitchoff who assisted formatting it. We also thank the National Science Foundation for their support via grant CCR-8804085.

REFERENCES

1. E.W. Adams and R. Fagot, "A model of riskless choice," in: Eds. W. Edwards, A. Tversky, *Decision making*, (1967) 284-289.

2. K.J. Arrow, "Public and private values," in: Eds. S. Hook, *Human values and economic policy*, New York University Press (1967) 3-31.

3. S. Bendifallah and W. Scacchi, "Work structures and shifts: an emperical analysis of software specification teamwork," *11th International conference on software engineering*, (May 1989) To appear.
4. S.E. Conry, R.A. Meyer, and V.R. Lesser, "Multistage negotiation in distributed planning," in: Eds. A.H. Bond, L. Gasser, *Readings in distributed artificial intelligence*, Morgan Kaufmann , San Meteo, California (1988) 367-384.
5. B. Curtis, H. Krasner, and N. Iscoe, "A field study of the software design process for large systems," *CACM* **31** (November 1988) 1268-1287.
6. Randall Davis and Reid G. Smith, "Negotiation as a metaphor for distributed problem solving," *Artificial Intelligence* **20** (1983) 63-109.
7. M. Deutsch, *The resolution of conflict: constructive and destructive processes*, Yale University, New Haven (1973).
8. T. Dietterich and R. Michalski, "A comparative review of selected methods for learning from examples," in: *Machine learning: an artificial intelligence approach*, Tioga Publishing (1983) 41-82.
9. Edmund H. Durfee, Victor R. Lesser, and Daniel D. Corkill, "Coherent cooperation among communicating problem solvers," *IEEE Transactions on Computers* **C36** (1987) 1275-1291.
10. E.H. Durfee and V.R. Lesser, "Using partial global plans to coordinate distributed problem solvers," *Transactions on computers* **C-36** (1987) 1275-1291.
11. M. Feather, S. Fickas, and W. Robinson, "Design as elaboration and compromise," in: *Proceedings of the Workshop on Automating Software Design*, Kestrel Institute , AAAI-88, St. Paul, MN (August 25, 1988) 21-22.
12. M.S. Feather, "Language support for the specification and development of composite systems," *Transactions on Programming Languages and Systems* **9** (April 1987) 198-234.
13. M. S. Feather, "Constructing specifications by combining parallel elaborations," *Transactions on Software Engineering* **15** (February 1989) To appear (Also available as Technical Report RS-88-216 from ISI).
14. L. Festinger, *Conflict, Decision, and Dissonance*, Tavistock Publications, Ltd., London (1964).
15. S. Fickas and P. Nagarajan, "Being suspicious: critiquing problem specifications," *Proceedings of the 7th National Conference on Artificial Intelligence*, (August 1988) 19-24.
16. S. Fickas, S. Collins, and S. Olivier, "Problem acquisition in software analysis: a preliminary study," CIS-TR-87-04, University of Oregon (January 1988).
17. A. Finkelstein and H. Fuks, "Multi-party specification," *5th International workshop on software specification and design*, (1989) 185-195.
18. M.R. Garey and D.S. Johnson, *Computer and intractability: a guide to the theory of NP-completeness*, W.H. Freeman, New York (1979).
19. M.P. Georgeff, "A theory of action for multiagent planning," in: *Proceedings of 1984 conference of the AAAI*, Morgan Kaufmann Publishers (1984) 121-125.
20. Carl Hewitt, "Offices are open systems," *Trans. on Office Information Systems* **4** (1986) 271-287.
21. E. Jantsch, *Technological Forecasting in Perspective*, Organization for Economic Cooperation and Development, Paris (1967).
22. R.L. Keeney and H. Raiffa, *Decisions with multiple objectives*, John Wiley and Sons, New York (1976).
23. M. Klein and S. C-Y Lu, "Run-time conflict resolution in cooperative design," *AI and Design Workshop*, (1988) To appear.
24. W.A. Kornfeld, "The scientific community metaphor," *IEEE Transactions on Systems Man Cybern* **SMC-11** (January 1981) 24-33.
25. Susan Lander and Victor Lesser, "Negotiation among cooperating experts," *AI and Design Workshop*, (1988) To appear.
26. D.B. Lenat, "The nature of heuristics," *Artificial Intelligence* **19** (1982) 189-249.

27. D.B. Lenat, "The nature of heuristics II," *Artificial Intelligence* **21** (1983) 31-59.

28. D.B. Lenat, "The nature of heuristics III," *Artificial Intelligence* **21** (1983) 61-98.

29. Marc Luria, "Goal conflict concerns," *IJCAI-87*, (1987) 1025-1031.

30. J. L. Peterson, "Petri nets," *Computing Surveys* **9** (September 1977) 223-252.

31. D.G. Pruitt, *Negotiation Behavior*, Academic Press Inc. (1981).

32. Howard Raiffa, *The art and science of negotiation*, Harvard University Press (1982).

33. W.N. Robinson, *Towards formalization of specification design*, Masters thesis, University of Oregon (June 1987).

34. W.N. Robinson, "Automating the parallel elaboration of specifications: preliminary findings," Technical Report CIS-TR-89-02, University of Oregon (February 1989).

35. W.N. Robinson, "Integrating multiple specifications using domain goals," *5th International workshop on software specification and design*, (1989) 219-226 (Also available as Technical Report CIS-TR-89-03 from the University of Oregon).

36. W.N. Robinson, "Integrative reasoning: a new paradigm for acquiring and combining knowledge structures applied to specification design," Technical Report, University of Oregon (Forthcomming 1989).

37. D.T. Ross and K.E. Schoman Jr., "Structured analysis for requirements definition," *Transactions on Software Engineering* **SE-3** (January 1977) 6-15.

38. E.D. Sacerdoti, *A structure for plans and behavior*, American Elsevier (1974).

39. W. Scacchi, "Managing software engineering projects: a social analysis," *Transactions on software engineering* **SE-10** (January 1984) 49-59.

40. R.N. Shepard, "On subjectively optimum selections among multi-attribute alternatives," in: Eds. W. Edwards, A. Tversky, *Decision making*, (1967) 257-283.

41. Reid G. Smith, "The contract net protocol: high-level communication and control in a distributed problem solver," *IEEE Transactions on Computers* **12** (December 1980) 1104-1113.

42. G.J. Sussman, *A computer model of skill acquisition*, American Elsevier (1975).

43. Katia Sycara, "Resolving goal conflicts via negotiation," *Proceedings of the AAAI-88*, (1988) 245-250.

44. K.P. Sycara, "Resolving adversarial conflicts: an approach integrating case-based and analytic methods," GIT-ICS-87/26, Georgia Institute of Technology (1987).

45. R.E. Walton and R.B. McKersie, *A behavioral theory of labor negotiations: an analysis of a social interaction system*, McGraw-Hill, New York (1965).

46. R. Wilensky, *Planning and understanding*, Addison-Wesley (1983).

47. Milan Zeleny, *Multiple criteria decision making*, McGraw-Hill (1982).