MASSACHUSETTS INSTITUTE OF TECHNOLOGY

ARTIFICIAL INTELLIGENCE LABORATORY

A. I. Working Paper 102 January, 1975

Bargaining Between Goals

Ira P. Goldstein

ABSTRACT

Bargaining is a process used to modify conflicting demands on an expendable resource so that a satisfactory allocation can be made. In this paper, I consider the design of a bargaining system to handle the problem of scheduling an individual's weekly activities and appointments. The bargaining system is based on the powerful reasoning strategy of producing a simplified *linear plan* by considering the various constraints independently and then debugging the resulting conflicts.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-70-A-0362-0003.

* An abbreviated version of this paper is to appear in the Proceedings of IJCAI IV.

Contents

Bargaining Between Goals

Introduction

Bargaining is a process used to modify conflicting demands on an expendable resource so that a satisfactory allocation can be made. In this paper, I consider the design of a bargaining system to handle the problem of scheduling an individual's weekly activities and appointments. From a practical standpoint, this problem is of interest as one of a class of computer applications that fall into the general category of personal assistants. From a theoretical standpoint, rational bargaining represents an important improvement to goal-driven systems that avoids unnecessary and time-consuming failure-driven searches. The proposed scheduler is an example of a *common sense system* in that it contains a wealth of knowledge about ordinary activities and people as well as a set of reasoning strategies that can deal with apparent conflicts and contradictions. Furthermore, the bargaining system is based on the powerful reasoning strategy of producing a simplified *linear plan* and then debugging it. Hence, it represents an extension of recent work on debugging [Goldstein 74, Sussman 73] applied to a new, non-procedural domain.

Our bargaining system will be built upon three basic ideas: the first is the use of *frames*, a generalized property-value representation that provides a rich description of acceptable assignments for each value. Frames are used by the scheduler to describe both people and activities. A vocabulary is developed for expressing default choices, legitimate variations, preferences and requirements. The second is the generation of a *possibility space* that provides a simplified overview of alternative solutions. The possibility space is constructed from linear plans, i.e. schedules for each activity developed independently of one another. The third is an explicit representation of various *bargaining techniques* that include strategies for relaxing defaults,

preferences and even requirements. These bargaining techniques are used to debug linear plans by resolving conflicts between activities scheduled at the same time.

The scheduler which I shall describe plans a week's activities for an individual as well as alters his tentative scheduling in response to unexpected appointment requests. It is member of a class of AI projects generally called Personal Assistants. Personal Assistants are an increasingly popular domain for AI research. At MIT, Fred Kern is programming and continuing the development of the scheduler described in this paper [Kern 74]; Mitch Marcus has considered the information retrieval problem [Marcus 74] and Dave McDonald has analyzed the task of English generation by the assistant [McDonald 74]. Projects for programming a Travel Budget Manager and an Apartment Finder are currently underway at Bolt, Baranek and Neuman [Woods 74] and Xerox Palo Alto Research Center [Bobrow 74]. This popularity is well-deserved. It is the author's opinion that in the coming years AI techniques will make a significant impact on the Personal Assistant domain and, in particular, on the design of personal resource (money, time, effort) and personal information (letters, papers, notes) management systems.

Before entering into the details of our analysis, the reader may wish to know what this paper has to contribute to the scheduling problem that is not already included in the various scheduling programs that exist for allocating time, money or space resources. The answer lies in the use of a much larger amount of knowledge in order to reach more intelligent accomodations between conflicting goals. The traditional scheduling situation is one in which there are a great many items to be scheduled, but, for each item, the system knows only a small number of absolute requirements. An example is allocating classrooms for lectures at a university, where for each lecture the system has been told the required length of time and frequency. This kind of problem is basically one of fitting together a jigsaw puzzle from pieces of predetermined shape. Our interest, however, is in a different kind of bargaining situation: namely one in which the number of events to be scheduled

is smaller; but where, for each event, we have available a rich set of knowledge regarding the relevant constraints. We do not expect to meet all of these constraints, but rather come to some compromise between conflicting goals. In our bargaining domain, the pieces of the jigsaw puzzle are over-specified and part of the solution lies in changing their shape.

A final caveat for the reader: the design of the scheduler described in this paper is as yet only tentative and the program is still to be written. I believe the paper should be of interest as a study of the kinds of knowledge and techniques that play a role in the common sense process of satisfying conflicting constraints. But there inevitably will be some fuzzy points in the ensuing discussion, especially with respect to the overall control structure, that represent the gap between design and implementation.

Scheduling Frames

Frames are data structures that provide expectations regarding assignments of values to various important properties [Minsky 1974, Winograd 1974]. For our scheduling system, frames are provided which describe the time requirements of various activities and the particular preferences of individuals.

Let IRA be a hypothetical individual using our scheduling system. In order to produce a plan for the expected activities in IRA's week, without excessive specification on the part of the user, *default* assignments are provided for the important properties of the expected activities such as the desired TIME and DURATION of meetings of the Personal Assistant Research Group.

```
(pa-meeting frequency (default (day 7)))
(pa-meeting duration (default (hour 1)))
```

Similar advice is provided regarding other activities such as LUNCH, RESEARCH, and TEACHING.

However, conflicts may arise such as those caused by the unexpected arrival of an important visitor. To cope with them, additional information is provided regarding acceptable *ranges* in which a property's value may fall, *preferences* which are desirable but not necessary and *requirements* which must be met if the activity is to be successfully scheduled at all. The following assertions provide such information regarding the scheduling of preparation time for IRA's lectures.

```
(teaching (prep time) (requirement (before (prep time) (lecture time))))
(ira (prep duration) (default (hour 6)))
(ira (prep duration) (range (between (hour 1) (hour 6))))
(ira (prep time) (preference (during (prep time) (day (lecture time)))))
```

Further examples of each of these different kinds of advice regarding value assignments is provided in Figure 1. This type of information allows the system to engage in various kinds of bargaining which we shall describe in greater detail in a later section.

Linear Plans and Possibility Spaces

In order to generate a reasonable default plan for the week's expected activities, the scheduler collects IRA's requirements, defaults and preferences for each of his activities. Figure 1 illustrates a subset of the database viewed from this perspective. The advice regarding each activity is collected from the person's frame, the activity's frame, and superiors of these frames pointed to by a-kind-of pointers.

Figure 1 -- Scheduling Advice for IRA's Weekly Activities

IRA'S PREFERENCES REGARDING THE SCHEDULING OF MEETINGS

```
(meeting duration (default (hour .5)))
(ira (meeting time) (default :friday))
```

IRA'S PREFERENCES REGARDING THE WEEKLY PERSONAL ASSISTANT MEETING

From the frame for PA-MEETINGS

```
(pa-meeting time (preference (during time :afternoon)))
(pa-meeting frequency (default (day 7)))
(pa-meeting frequency (range (day 6), (day 7), (day 8)))
(pa-meeting duration (default (hour 1)))
(pa-meeting duration (range (between (hour .75) (hour 1.5))))
```

From the frame for IRA

```
(ira (meeting time) (preference (during time :friday)))
```

IRA'S PREFERENCES REGARDING THE SCHEDULING OF LUNCH

From the frame for LUNCH

```
(lunch frequency (once :day))
(lunch time (default (am 12)))
(lunch time (range (between (am 11) (pm 2))))
(lunch duration (default (hour 1)))
(lunch duration (range (between (hour .5) (hour 1.5))))
```

From the frame for IRA (These properties take precedence over the more general LUNCH frame)

```
(ira (lunch time) (default (am 11)))
```

IRA'S PREFERENCES REGARDING THE SCHEDULING OF RESEARCH

From the frame for RESEARCH

```
(research duration (requirement (≥ duration (hour 1))))
```

From the frame for IRA

```
(ira (research duration) (default (workday 1)))
(ira (research duration) (preference (maximize (total duration))))
(ira (research time) (preference (during time :morning)))
(ira (research time) (preference (during time :monday)))
```

IRA'S PREFERENCES REGARDING SCHEDULING OF TEACHING

From the frame for TEACHING

```
(teaching (prep time) (requirement (before (prep time) (lecture time))))
```

From the frame for IRA

```
(ira (lecture time) (tues 3:30))
(ira (lecture time) (thurs 3:30))
(ira (lecture duration) (hour 1.5))
(ira (prep duration) (default (hour 6)))
(ira (prep duration) (range (between (hour 1) (hour 6))))
(ira (prep time) (preference (during (prep time) (day (lecture time)))))
```

A *linear plan* is then constructed for each activity, i.e. it is scheduled at some time, independently of the existence of other activities. The time is chosen on the basis of advice extracted from the frame for that activity, from superiors of that frame linked to it by chains of "a-kind-of" relations and from advice about that frame contained any person frames involved as participants.

The set of linear plans for each of an individual's activities forms a *possibility space* of alternative calendars for the week. Figure 2 shows such a possibility space for IRA. (Teaching actually consists of two activities, lecturing and preparation. However, for simplicity, it is shown as a single block of time.)

Figure 2 -- Possibility Space for the Week's Activities

				pa-meeting [1 2]
	teaching [9 5]		teaching [9 5]	meeting [9 5]
research [9 5]	research [9 5]	research [9 5]	research [9 5]	research [9 5]
lunch [11 12]	lunch [11 12]	lunch [11 12]	lunch [11 12]	lunch [11 12]
M	T	W	R	F

This structure is built by considering each activity of IRA independently of the others. RESEARCH, for example occurs on every day because of the preference that its total duration be maximized. LUNCH is scheduled at the default time of 11 rather than 12 because IRA's preference for an early lunch overrides the default time of 12 given in the lunch frame. TEACHING includes

both classtime and preparation and is therefore intended to last the entire day. The conjunction of a preference for meetings on Fridays and pa-meetings in the afternoon causes the PA-MEETING to be scheduled on Friday at 1.

The possibility space, viewed as a conjunction of linear plans, has bugs in the form of conflicting activities. Lunch and research, lunch and teaching, research and teaching, research and meetings -- all of these pairs of activities conflict. Thus, we have found the first instance in which bargaining techniques must be applied.

Debugging Linear Plans

A debugged plan for IRA is developed by resolving these conflicts. This section provides an informal trace of the bargaining analysis required to do this. For each conflict in the possibility space, the BARGAINER examines the conflicting activities, applies different strategies and ultimately arrives at a decision. The next section then presents the BARGAINER in a more formal way, outlining each of its current strategies.

Resolving the conflict between Teaching and Research on Tuesdays and Thursdays

Teaching and research are not designated as time-sharable activities. Therefore, a conflict exists on Tuesdays and Thursdays. Research is scheduled then as a preference to maximize the total research duration. Teaching consists of two activities: lecturing and preparation. It is a requirement that lecturing be scheduled from 3 to 5 on Tuesdays and Thursdays. It is only a preference that preparation occur on the same day. Using a coarse *utility measure*, a requirement dominates over any number of preferences; and hence the conflict is resolved in favor of the teaching. The BARGAINER, as it is currently designed, will confirm this with hypothetical

reasoning of the following sort: rescheduling the preparation will conflict with research scheduled at some other time. Therefore, it might as well be left as is. (The BARGAINER might equally well reason that a *maximizing preference* such as research should never dominate over an ordinary preference. This is explained in the next paragraph.)

-> *Decision*: cancel research on teaching days.

Resolving the conflict between Research and Meetings on Friday

Research and meetings conflict on Friday. It is required that a meeting day occur every 7 days. Thus, permitting research on Friday only results in MEETING being scheduled on another day and subtracting from research time then. It is a zero sum situation. Conflicts in which one event is justified by a maximizing preference while the other must occur at some point in the week (although the particular time chosen is not a requirement) are resolved in favor of the latter (since the former will be scheduled at all possible times). Rescheduling the required activity will simply result in it conflicting at another point with the maximizer.

-> *Decision*: cancel research on the meeting day.

The time set aside for meetings may not actually be needed. In this case, the scheduler should return to the alternative choice of research. The current design plans to achieve this by having the scheduler retain pointers from the decision back to the original conflict and consequently the rejected alternative. If the potential meeting time arrives and no actual meeting is scheduled, then the rejected linear plan for this time slot is pursued. The result will be that the system will remind the user that the time can now be used for research rather than meetings.

Resolving the conflict between Lunch and other daily activities

LUNCH [11 12] apparently conflicts with the other daily activities of research, meetings and

teaching. Lunch is an absolute requirement; however, its frame indicates that it is a sharable activity. Thus, one bargaining solution, *sharing*, is for lunch to happen at the same time as other overlapping activities. On the other hand, research and preparation are interruptible. Thus, a second solution, *insertion*, is to interrupt these activities long enough to have lunch. Sharing is preferred since it maximizes research time, although subsequent time demands might result in the latter solution being chosen.

-> *Decision*: Lunch can share time with other activities of day. No conflict.

Ignoring the apparent conflict between pa-meetings and meetings on Friday

PA-MEETING apparently conflicts with MEETING. However, PA-MEETING is a kind of MEETING. When activities are in this hierarchical relation, the conflict is only apparent. Instead, what this overlap actually represents is simply a further specification of how the MEETING time will be used. Conflicts only occur between activities on different branches of the hierarchy tree defined by the frame system.

-> *Decision*: No conflict. Example of further specification.

Figure 3 -- The Weekly Plan

				pa-meeting [1 2]
research [9 5]	teaching [9 5]	research [9 5]	teaching [9 5]	meeting [9 11], [12 5]
lunch [11 12]	lunch [11 12]	lunch [11 12]	lunch [11 12]	lunch [11 12]
M	T	W	R	F

Bargaining Techniques

Bargaining techniques are required to debug conflicts in the weekly plan of a single individual as well as to schedule appointments given conflicting constraints of the participants. These techniques fall into two classes. The first, which I shall call *resource-driven*, are experts at altering the particular interval chosen by some goal, while still satisfying the goal. They include *relaxing defaults, relaxing preferences, swapping intervals, time sharing* and *interrupting*. These strategies are expert at manipulating the "time" resource. They have obvious analogues for manipulating other resources like space and effort. These strategies do not question the justification for pursuing the goal, but rather attempt relatively local alterations of the various kinds of advice -- defaults, range, preferences -- that allow the interval to be altered without actually violating the overall goal.

The second class of techniques are *purpose-driven*. These strategies, as opposed to the resource-driven techniques, are capable of eliminating or modifying requirements. They do so by altering the least important goals, as chosen by analyzing the overall purpose of the event. The resource-driven techniques are ignorant of the relative importance of competing goals and are therefore unable to be as radical as the purpose-driven strategies.

These bargaining techniques can be viewed as *debugging strategies* for resolving unforseen interactions between linear plans. As such, they represent a further analysis of the *simplify and debug* problem solving paradigm explored in recent papers by [Goldstein 74] and [Sussman 73].

The following paragraphs describe each of the techniques that we currently plan on implementing in the scheduler. The control algorithm which administers these techniques is discussed in a later section on *Control Structure*.

Compromise by Relaxing Defaults

Default choices represent only estimates. Conflicts can sometimes be resolved by changing the default, while still remaining within the required range.

Compromise by Insertion

Some activities are indicated as being interruptible, e.g. research. For such activities, the negotiation strategy is available of interrupting one activity long enough to insert another. A heuristic restriction limits any activity to at most two interruptions. (A more adaptable system might inquire of the person being modeled how many interruptions he is prepared to tolerate for each of his "interruptible" activities.)

Compromise by Sharing

Some activities are indicated as being sharable, e.g. lunch. Conflicts between such activities can be eliminated by scheduling them at the same time, i.e. the conflict is dismissed. A more intelligent scheduler might be able to reason that some activities like eating and meeting can be shared while others like eating and sleeping cannot. For simplicity, I currently intend to handle this by simply grouping activities into two classes: sharable and unsharable. A sharable activity can share time with any activity not explicitly labeled unsharable. However, a desirable extension would be to represent in the frame for a given activity, exactly those other activities with which it can share time and those with which it cannot.

Compromise by Swapping

Sometimes the requirements and preferences of some activity imply only a duration and a frequency, but do not actually specify the day. Hence, some conflicts can be resolved by swapping

one such activity with another. The virtue of the swap is that the new activity scheduled in the blocked time may be relaxable, interruptible or sharable, whereas the previous occupant of the time slot was not. A swap is prevented if the event is scheduled at the original time due to requirements of the activity or the participants that entail a specific day.

Compromise by Relaxing Preferences

Conflicts between competing requirements and preferences are resolved in favor of requirements while conflicts between competing preferences are decided in favor of maximizing the number of satisfied preferences. This strategy of relaxing preferences is tried only after the others listed above. It generally fails when requirements make removing the block impossible.

Possibly a more subtle use of numerical utilities will be necessary. For example, a common sense system ought to be able to take account of the advice that a particular individual prefers a meeting (1) as early in the day as possible, (2) as soon as possible and (3) in the event that these two preferences conflict due to pre-existing appointments, to consider one day earlier to be worth making the meeting one hour later in the day.

Compromise by Request

This and the following two strategies represent techniques that can relax or eliminate requirements. They do this by questioning not simply the local choice of time, but the overall justification for the activity or for some of the participants. I shall call these techniques *purpose-driven* bargaining.

A measure is provided for the relative importance of various participants. When absolute requirements of individuals conflict and there is no resource-driven settlement possible, the system asks the less important individual to compromise by relaxing his requirements. Figure 4 illustrates

the frames which define the MORE-IMPORTANT relation.

Figure 4 -- Frame definition of the More-Important Relation

```
;Frame for a relation
    (comparison a-kind-of relation)
    (comparision 2-place relation)
    (comparison transitive)
    (comparison anti-symmetric)


;Frame for the MORE-IMPORTANT relation
    (more-important a-kind-of comparison)
    (more-important arg1 person)
    (more-important arg2 person)
    (more-important leader participant)
    (more-important teacher student)
```

This static partial ordering represents, of course, a gross simplification. Ultimately, I would like to see the system be able to change the "importance" with which the preferences of an individual are regarded in accordance with the context-dependent role that he happens to be playing, e.g. host versus employer versus friend. Purpose-reasoning is introduced in the next section to partly meet this need of a more dynamic importance relation.

Compromise by Elimination

Again the strategy is to relax requirements, except in this case, the BARGAINER attempts to decide which requirements can be eliminated, rather than asking the participants involved. This is accomplished by utilizing the MORE-IMPORTANT metric described above plus *purpose checking*. Purpose checking is necessary to prevent the pre-defined importance relation from causing a person to be dropped from a meeting even though without him the meeting is pointless.

An example is the problem of scheduling an Oral Examination for three professors and a student. The pre-defined ordering of importance declares professors to be more important to

students, in the sense that given conflicting desires for a meeting time, the preferences of the professor are maximized. Hence, without purpose reasoning, conflicts in scheduling the Oral Examination would probably result in the student being dropped from the list of participants. This is prevented by declaring that the purpose of the group meeting is for the student to be present.

Currently, the system is capable of only very elementary purpose reasoning based on "required participants". Such requirements are recorded either as assertions in the appropriate activity frames or as additional advice given to the system at the time of the appointment request. Typical examples are:

```
(meeting participant (requirement (≥ (number participant) 2)))
(pa-meeting participant (requirement (participant leader)))
(oral-exam participant (requirement (participant examinee)))
```

Purpose reasoning is a type of common sense logic that AI programs must have to be able to debug conflicts. The scheduler, as currently planned, will be capable of it in only an elementary way. This is clearly an area which merits further study.

Compromise by Substitution

Substitution applies the swapping technique described above for intervals to participants. Again, purpose reasoning is involved. If a visitor requests an appointment with a faculty member of the laboratory, the request is generally routed to some particular individual. If he cannot schedule an appointment, then the bargainer seeks a substitute participant from its list of known people and *requests* that he accept the appointment.

Compromise by Division

Resolve a conflict between participants by attempting to schedule the meeting twice for different subsets of the intended participants. This represents an instance of the general planning strategy of breaking a goal into sub-goals whose conjunction satisfies the original purpose. In the scheduling context, it is often an inefficient technique because some subset of the participants usually must attend both meetings. Hence, it is generally not preferred. Some activities such as oral-exams are not divisible into separate meetings. This is indicated in the frame for this activity under the TYPE property.

Comparison to Traditional Scheduling Algorithms

Recall our earlier reference to traditional scheduling programs. We can now be more precise in stating their limitations. Because such algorithms are knowledge-poor and do not know anything but the basic requirements of the various items to be scheduled, they cannot *relax defaults* or *preferences*, nor know whether a particular item is *interruptible* or *sharable*. Similarly, they are not knowledgable enough to examine purposes and decide upon *eliminations* or *substitutions*. On the other hand, the bargaining system proposed here has available to it a richer set of strategies for reaching an accomodation because it has far more knowledge about each item, knowing both preferences, ranges and defaults in addition to requirements.

A Meeting Scenario

I shall further illustrate the use of these bargaining techniques by noting that an individual's weekly plan is not absolute. An unexpected visitor may arrive and request an appointment on a day previously planned for some other activity. In such a case, we would like the system to

consider its options and reschedule its planned activies if possible. The scheduler's strategy for accomplishing this is again to build a possibility space of linear plans. This space is then examined for the least blocked times and bargaining techniques are applied in an attempt to remove the constraints responsible for the conflict.

Suppose a visitor sends IRA a telegram requesting an appointment on the next Monday. This results in the following *possibility space* of appointments. "(+ <PERSON>)" indicates that the person can make an appointment during the indicated time while "(- <PERSON>)" indicates that he cannot. The first line of Figure 5 is the linear plan for when IRA can make the appointment and the second line is the linear plan expressing MV's choice of meeting time.

Figure 5 -- Possibility Space of Appointments for IRA and MV

(- IRA)	(- IRA)	(- IRA)	(- IRA)	(+ IRA)
(+ MV)	(- MV)	(- MV)	(- MV)	(- MV)
M	T	W	R	F

The BARGAINER examines this Possibility Space and looks for the best point at which to consider a compromise. This is the least blocked interval. Monday and Friday are least blocked intervals with blockage equal to 1. The following paragraphs trace the alternative bargaining strategies that are considered in an effort to find an acceptable appointment time. Ultimately, the BARGAINER chooses the strategy that violates the fewest preferences and requirements. (If there is more than one such "best" solution, the user is asked to choose among them.)

Resource-driven compromise by relaxing defaults, swapping, sharing, interrupting and relaxing preferences are all possible if the blocking event event is not occupying its interval as a result of a requirement. Hence, all of these strategies are applicable to the block to the appointment on Monday caused by IRA's RESEARCH. On the other hand, the Friday block is due

to the requirement, at least as expressed to IRA's scheduler, that the visitor will only be in town on Monday. To this block, only purpose-driven strategies, i.e. compromise by request, by elimination, or by substitution, are possible.

Three of the strategies which can be applied to eliminate the Monday block are bargaining by relaxing defaults, by insertions and by swapping. That research on Monday has a duration of all day is a default. Hence, the boundaries of the research interval can be *relaxed* by one hour to make room for the appointment. This produces the plan: *schedule the appointment from [9 10] or [4 5]*. Research is also *interruptible*. Therefore, it is possible to insert the appointment into the Monday research, producing the plan: *schedule the appointment for 1 hour any time during Monday*. Once this is done, the research is marked as no longer interruptible. Currently, the bargaining system prefers relaxing boundaries to interrupting an activity, in the absence of any governing preferences or requirements. (This is based on the introspective observation that people prefer not to be interrupted once engaged in an activity. Another user, however, could obviously alter this default choice.)

Swapping produces the plan of interchanging research time on Mondays with meeting time on Fridays. This approach would be preferred to either of the first two if none of the affected Friday activities had governing preferences that tied them to that day of the week. For example, using Friday as an appointment day is a default but not a preference for IRA. Hence, the swapping plan would be chosen to avoid violating the preference for maximizing research time since the first two plans decrease research time by an hour.

Purpose-Driven Bargaining

The above resource-driven techniques would be inapplicable if IRA had a requirement that research occur all day on Monday (e.g. a funding deadline is drawing nigh). In such an event, the

bargainer would apply various purpose-driven techniques to alter either Ira's or the Visitor's requirements. The following paragraphs illustrate the application of such techniques to the Visitor's request.

The strategy of attempting to compromise by *request* results in the Monday Visitor being asked by *IRA*'s scheduler if he can reschedule his trip to be in town on Friday. In a world of personal assistant programs, this would involve a call to the visitor's personal assistant program.

Compromise by *substitution* is possible if the visitor's orginal appointment request did not REQUIRE a specific faculty member. If so, the plan would be to *reschedule the appointment with another equivalent participant.*

The two bargaining strategies of compromise by *Elimination* or *Division* are inapplicable. There are only two participants so the number of people attending the meeting cannot be relaxed. Similarly, sub-division does not apply with only two participants.

Control Structure

This section develops in greater detail how the system preserves the rationale of scheduled blocks of time (in order to support subsequent interrogation by the BARGAINER) and how the BARGAINER mediates between its various relaxation and reformulation strategies.

Representation of Rationales

Rationales are preserved by back-pointers from the assigned interval (e.g. `pa-meeting [1 2] Friday`) to the set of database assertions used to schedule the activity at that time.

Figure 6 -- Rationale Pointers

```
(pa-meeting [1 2] Friday)
  -rationale->
      ( (pa-meeting time (preference (during time :afternnon)))
        (pa-meeting frequency (default (day 7)))
        (pa-meeting frequency (range (day 6), (day 7), (day 8)))
        (meeting duration (default (hour 1)))
        (meeting duration (range (between (hour .75) (hour 1.5))))
        (ira (meeting time) (preference (during time :friday))) )
```

To modify the scheduled time or cancel the activity entirely, the bargainer examines these assertions. If requirements exist which necessitate the activity at this particular time, then the resource-driven techniques are inapplicable and the BARGAINER immediately applies the various purpose-driven strategies. On the other hand, if the rationale indicates that the activity was scheduled solely because of various preferences and defaults, then the resource-driven strategies are applicable. Preferences and defaults can be relaxed. Swapping, insertion and time sharing are possible only if the activity is of the appropriate type.

Actually, I plan to use a somewhat more sophisticated representation using the generalized tag mechanism of CONNIVER [McDermott 74]. The "rationale" will actually be represented by a pointer back into the environment of the goal which originally scheduled the activity at that time. That goal will contain a variable set to the list of relevant database assertions. The environment of the goal will also contain information regarding previous decisions (reentries) that were made. This use of tags to allow reentry to goals that have already returned is based on the architecture of the BUILD program [Fahlman 74].

Control of the Bargaining Strategies

Under the current design, all of the applicable strategies are tried, each producing an alternative plan for eliminating the a given scheduling conflict. The final solution is chosen on

the basis of which plan violates the fewest preferences and requirements. This preserves the philosophy of maintaining a global perspective in the form of a Possibility Space of alternative plans and not becoming trapped by an excessively local perspective.

In those cases where there are many conflicts (such as arise in planning an entire week for a single individual or a meeting among a group of people), our current implementation plan is to apply the bargaining strategies as search operators. Each strategy operates on the current calendar, producing a new calendar with fewer conflicting events. Terminal nodes of the resulting tree are those with calendars containing no conflicts. Procedurally, the current plan is to use a Coroutine Search which initially explores the search space breadth first, but suspends those paths whose "utility" is less than the current maximum, where the utility is defined simply as the sum of satisfied preferences. If the paths currently being explored encounter difficulties and their utility drops below that of some of the suspended nodes, then those nodes are reactivated. (This is the same kind of search as was used to find the plan of uncommented simple programs in [Goldstein 74] and is also similar to the kind of search used by Woods in the LUNAR system [Woods 74].)

An alternative control strategy would be to have the blocking activity engage in bargaining with the other activities scheduled at the same time in the possibility space. This represents a *negottatton* model of the bargaining process and might be implemented in an ACTOR-like formalism [Hewitt 73]. Currently, I lean towards making the BARGAINER an expert in its own right and representing the common sense of bargaining expertise in a single package. (An *arbitratton* model of the bargaining process.) The activities are simply static packets of knowledge. However, as the system is implemented, the relative merits between the distributed and centralized bargaining models should become clearer.

Conclusions

Our general approach has been one of generating a global overview and then debugging conflicts until a possible solution is obtained; rather than searching through a space of solutions by generating single possibilities until some choice meets all of the constaints. This latter approach, common in many AI programs, is non-optimal in the sense that the first plan derived which satisfies all of the REQUIREMENTS may very well not maximize the PREFERENCES. The use of a *possibility space* both for the weekly plan and for possible meeting times avoids this myopia.

The bargaining techniques described above are an attempt to represent a kind of common sense reasoning that people commonly engage in as they juggle the various demands being placed upon their limited resources of time, money, and energy. Few AI programs in the past have evinced any kind of robustness when the initial request is "unsolvable" and either the problem must be relaxed or reformulated. Combining *bargaining* with a rich frame-oriented description of ordinary activities and people is a beginning towards permitting this new dimension of problem solving.

Finally, the last point which should be made about the scheduler is that it is a knowledge-rich system. Its success in finding a reasonable solution to organizing a week's activities is fundamentally based upon a detailed description of the various people and activities involved. Future extensions of the scheduler will include a natural language discourse component. The knowledge-base will then be doubly important, as it would be expected to support the required natural language ability of having reasonable expectations about what the speaker will say. Such expectations are necessary to provide a guide for the parsing and generation processes.

Much further work needs to be done in the Personal Assistant domain (I have not touched upon the problem of how the system might create and modify its own frames), but I hope that this paper serves as an interesting discussion of some of the knowledge and reasoning strategies that

will be required to achieve a system capable of common sense.

Bibliography

< 1 > [Bobrow 1974] Bobrow D.G. et. al. "Steps Towards Language Understanding", Internal Memo, XEROX-PARC, Palo Alto, California.

< 2 > [Fahlman 74] Fahlman, S. "A Planning System for Robot Construction Tasks", *Artificial Intelligence*, Volume 5 (Spring, 1974), 1-49.

< 3 > [Goldstein 74] Goldstein, I.P. *Understanding Simple Picture Programs* AI-TR-294, MIT AI Laboratory (September 1974).

< 4 > [Goldstein 75] Goldstein, I.P. (ed.). "Progress on the Personal Assistant Project", working paper in progress, MIT Artificial Intelligence Laboratory, 1975.

< 5 > [Hewitt 73] Hewitt, C., et. al. "Actor Induction and Meta-evaluation", Conference Record of ACM Symposium on Principles of Programming Languages Boston, Oct. 1973.

< 6 > [Kern 1974] Kern, F. "An Implementation of a Personal Assistant Scheduling System", in Goldstein, I.P. (ed.), "Progress on the Personal Assistant Project".

< 7 > [Marcus 74] Marcus, M. Document Retrieval from an AI Point of View in Goldstein, I.P. (ed.), "Progress on the Personal Assistant Project".

< 8 > [McDermott 74] McDermott, D. and Sussman, G.J. "The CONNIVER Reference Manual", AI Memo 259A, MIT.

< 9 > [McDonald 74] McDonald, D. Language Requirements of a Personal Assistant in Goldstein, I.P. (ed.), "Progress on the Personal Assistant Project".

< 10 > [Sussman 73] Sussman, G.J. *A Computational Model of Skill Acquisition* AI-TR-297, MIT AI Laboratory (August 1973).

< 11 > [Minsky 74] Minsky, M. "A Framework for Representing Knowledge." To appear in Winston, P. (ed.), *The Psychology of Computer Vision*, McGraw Hill, New York, 1975.

< 12 > [Winograd 74] Winograd, T. "Frame Representations and the Declarative/Procedural Controversy", to appear in Bobrow and Collins (eds.), *Representation and Understanding: Studies in Cognitive Science*, Academic Press (in press).

< 13 > [Woods 74] Woods, W. A. "Speech Understanding Research at BBN", BBN Report 2976, Volume I.