

A Commitment-Based Framework for Describing Informal Cooperative Work

by

Richard E. Fikes
Cognitive and Instructional Sciences Group
Xerox Palo Alto Research Center

June 1981

Abstract

In this paper we present a framework for describing cooperative work in informal domains such as an office. We argue that standard models of such work are inadequate for describing the adaptability and variability that is observed in offices, and are fundamentally misleading as metaphors for understanding the skills and knowledge needed by computers or people to do the work. The basic claim in our alternative framework is that an agent's work is defined in terms of making and fulfilling commitments to other agents. The tasks described in those commitments are merely agreed upon means for fulfilling the commitments, and the agents involved in the agreement decide in any given situation how and whether a given commitment has been fulfilled. We also claim that in informal domains, descriptions of tasks, functions, and procedures are necessarily incomplete and imprecise. They result from negotiations among the agents and serve as agreed upon specifications of what is to be done. The descriptions evolve during their use in continuing negotiations as situations and questions arise in which their meaning is unclear. These claims imply that for a given situation an agent using such descriptions must be capable of interpreting imprecise descriptions, determining effective methods for performing tasks, and negotiating with other agents to determine task requirements.

Introduction

In this paper we present a framework for describing cooperative work in domains where there is no agreed upon formal model of the tasks and functions to be done, nor of the procedures for doing them (i.e., in informal task domains). Most of the work people do is in such informal domains where it is not feasible to create precise statements of the tasks to be done, the situations in which they are to be done, the resources available, nor the capabilities of the processors doing them. For example, people are regularly confronted with task descriptions such as "write a progress report on the project", "describe the items to be purchased", "yield right of way", "keep your eye on the ball", or "slice chicken breasts very thinly into julienne strips"

The project that produced this framework has been focused on the problems of automating office work involving the use of prespecified procedures, and our experiences with those problems motivate and provide examples for the discussions in this paper. However, the results reported here are intended to be generally applicable in any task domain where tasks and procedures are informally specified, and agents enlist each other's help to achieve their individual goals.

We began our office automation efforts by attempting to develop a model of the work being automated that would provide a basis for our system design efforts. We were particularly interested in describing the skills and information needed to do the work, and in accounting for the adaptability and variability of methods used by people in performing their tasks.

Our initial thesis was that office procedures are like computer programs and they are "executed" by a collection of office workers in a manner analogous to a collection of computers executing a program. It seemed a simple matter to automate office procedures by storing them in a computerized data base as if they were programs. Then, at each step in the execution of the procedures in an office the computer could do the step itself, or tell the person doing the step what operation to do and then monitor the results.

As we proceeded, fundamental problems arose that led us to question our thesis [Fikes and Henderson]. One such problem was that our model did not account for the variability in the way tasks are accomplished. For example, an office worker has more options in following a procedure than our model described. He can choose to ignore some of the requirements of a task (e.g., leave some fields of

form blank), renegotiate the requirements of a task (e.g., request extension of a deadline), or use some method other than the standard procedure for performing a task.

A second problem was that our model did not account for the difficulties related to working with informally specified tasks, functions, and procedures. For example, the informality of office work makes infeasible the specification of precise algorithms for performing tasks. Situations occur in which the available procedures do not indicate what to do (e.g., a vendor claims that goods were delivered, but no record of their arrival can be found), what is indicated cannot be done (e.g., a deadline has already past), or what is indicated is not the preferred method for performing the task (e.g., because an unexpected resource is available). Hence, the work involved in using office procedures is qualitatively different from the work involved in executing a formal algorithm.

We concluded, then, that modeling procedural office work as simple program execution is an inadequate basis for automating it and is misleading as a metaphor for understanding the skills and knowledge needed by computers or people to do the work. That conclusion led us to search for alternative ways of modeling cooperative work that would account for the way office tasks are actually performed, and would inform us regarding the required skills and knowledge. This paper presents the initial results of that search, an alternative based on the agreements made by agents performing the work and agents for whom the work is done (see [Flores and Ludlow] for another analysis of office work based on such agreements).

Analyzing Cooperative Informal Work

The Social Nature of Tasks and Functions

Tasks

We begin by analyzing a simple work situation in which an agent has a task that he wants done. For the purposes of this discussion we will consider a task to be defined as a set of goals to be achieved while maintaining a set of constraints. The basic tenet of our model is that tasks are essentially social in nature in that they are done by one agent (a contractor) for some other agent (a client). The situation where an agent does a task for himself is the special case in which the client and contractor are the same agent.

A client (i.e., an agent who wants a task done) can choose to enlist some other agent to be the contractor for a task. The client accomplishes that enlistment by establishing a "task contract" with the contractor (see the work on contract nets [Davis and Smith] [Smith] for a detailed model of how such contracts are established). We model a contract as consisting of a collection of commitments, each made by one of the contracting agents to another of the contracting agents. A task contract is an agreement between two agents containing a commitment by one of the agents (the contractor) to do a task for a second agent (the client). The contract may also contain other commitments, such as a commitment by the client to remunerate (i.e., achieve some goal for) the contractor in return for accomplishing the task.

In order for a task contract to be established, the client and contractor must agree on the task that is to be performed. Their negotiations will produce an agreed upon *description* of the task, and the commitment will be a statement of intent to do the *described task*. Therefore, we consider cooperative tasks as being defined by a social process, and as representing a negotiated agreement between the client and contractor.

Note that a task contract establishes a goal for the contractor of fulfilling his commitment to the client, and performing the described task is only a means for achieving that goal. That observation is one of the bases for our explanations of the behavior variations observed in human cooperative work situations. That is, the agreed upon task description provides the contractor with a set of sufficient conditions for fulfilling his commitment, and therefore represents one method of achieving his goal. However, *any* actions by the contractor that result in the client agreeing that the commitment has been fulfilled will achieve the contractor's goal. For example, the contractor may choose to achieve some variation of the task's goal, ignore some of the constraints, convince the client that the task shouldn't be done, etc. He is free to use whatever method he thinks will succeed and is most desirable for him in the context of his other goals and constraints.

In order for a contractor to make use of the flexibility available to him in fulfilling his commitment, he must know who the client is, the client must be accessible to him for negotiation, and the contractor must be capable of planning and performing alternative courses of action to those described in the task contract.

Functions

Often in human work situations, a person will agree to perform a given type of task whenever a given set of conditions occur; that is, he will agree to perform a "function". For example, a buyer in a corporate procurement department may agree to issue a purchase order whenever a properly executed purchase request is received. Also, procedures are typically designed as methods for performing functions rather than individual tasks and are used whenever the function's task is to be done (e.g., the procedure for issuing purchase orders). Hence, in order to describe those situations, we will generalize our discussion to include functions as well as tasks.

As with tasks, one method available to a client for performing a function is to establish an agreement (a "function contract") with a contractor in which the contractor commits to do the function for the client. The contract will contain an agreed upon description of the function to be performed by the contractor. For our discussion, we will consider a function description to consist of a parameterized task description and parameterized set of preconditions such that any given instance of the preconditions defines an instance of the task. Whenever an instance of the preconditions becomes true, the contractor agrees to perform the corresponding instantiated task.

As with tasks, the contractor's goal is to fulfill his commitment and the agreed upon function description provides him with a set of sufficient conditions for achieving that goal. Each time the function's preconditions become true, the contractor can choose to do whatever actions he thinks will satisfy the client.

Note that in the transition from task to function a new subtask has been introduced; namely, recognition of the occurrence of the preconditions. Hence, an agent who has committed to perform a function must

establish monitors that recognize situations in which an instance of the function's task is to be performed.

Tasks and Functions in Informal Domains

An agent performing a function depends on the function description to specify each situation in which he is to do something and in each of those situations the task that he is to perform. In informal domains, use of those descriptions becomes problematic because of their imprecision and incompleteness (What is a "properly executed purchase request?") [Suchman]. Hence, the contractor is confronted with the new subtasks in each situation of interpreting the function description to determine whether a task is to be done, what the task would be, and then after doing something whether the task has been accomplished.

We claim in our model that the sole criteria for an acceptable interpretation of these descriptions is agreement by the contractor and client. That is, the function and task descriptions are part of the contract between the contractor and client, and those agents are the final authority as to what those descriptions mean and whether they have been satisfied. For example, the meaning of "describe the items to be purchased" on a purchase requisition form is worked out in each case by the requisitioner and the procurement department buyer, for whom the description is being created.

The interpretation of task and function descriptions in any given situation is therefore a subject for negotiation between client and contractor. That is, if a commitment description is not sufficiently precise or complete for the contractor to determine what he should do in a given situation, then additional negotiations with the client are necessary. Hence, in informal domains, the negotiation processes that produce commitment descriptions continue during the fulfillment of those commitments and become an integral part of the work required to fulfill them.

Agents performing functions in informal domains must be capable of determining appropriate interpretations of imprecise descriptions and of recognizing when the description is sufficiently inadequate to warrant renegotiation with the client. When agents are skilled in those capabilities, the difficult and time consuming process of creating comprehensive function and procedure descriptions can be avoided. Descriptions can be allowed to build up incrementally by generalizing the experiences gained in particular situations.

Functions as Operators for Planning

Functions play the same role as operators in standard Artificial Intelligence planning and problem solving frameworks (for example, [Fikes and Nilsson]) in that they can be used by agents to achieve goals. We said earlier that an agent who wants a task done can enlist a second agent to do the task by establishing a task agreement with the second agent. Functions provide an alternative method of enlisting a second agent to do a task. That is, if the second agent is a contractor who has made a commitment (it doesn't matter to whom) to provide a function, and the task that the first agent ("the consumer") wants done is an instance of that function's task, then the consumer can cause the contractor to do the task by persuading him that the appropriate instance of the function's preconditions are true. If the contractor refuses to do the task, then the consumer can appeal to the function's client, attempting to convince him that the preconditions were satisfied and that the contractor did not fulfill his commitment to accomplish the task.

For example, if an employee of a small company wants to obtain some equipment for use in his work, then he can achieve that goal by obtaining the appropriate authorizations and submitting the appropriate forms to the company's procurement department. The procurement department has made a commitment to the company president to be the contractor for the function of purchasing equipment, and receipt of the appropriate forms and authorizations is the precondition for that function. The employee becomes a consumer of that function by convincing the contractor that an instance of its preconditions have become true. If the procurement department refuses to provide the advance, the employee can complain to the company president that they are not performing their function.

In deciding to use a function, the consumer has replaced his original task with the new task of persuading the contractor to do the original task. Notice that the method for accomplishing the new task is to convince the contractor that an instance of the preconditions have been satisfied, rather than simply to make an instance of the preconditions true. The consumer is free to negotiate with the contractor as to what he will accept as satisfactory evidence that the preconditions are true. For example, the employee requesting an equipment purchase might convince the procurement department that a phone call from the employee's manager is sufficient in that case to authorize the purchase. If the preconditions are informally described, then there is the additional issue to be resolved in those negotiations of determining an agreed upon interpretation of the descriptions in the situation. For example, the employee might ask the procurement department to accept a memo requesting the purchase rather than the standard form. This is another case where negotiations during the performance of a task are vital to its completion and where variability is introduced by the one-time agreements that result from those negotiations.

Subcontracting to Perform Tasks and Functions

Consider again the basic situation in which a client wants a task done and has obtained a commitment from a contractor to do the task. We could then describe the contractor's situation as one in which he wants a task done, and that he has the option of persuading yet a third agent (a "subcontractor") to do some or all of the task for him. The subcontractor then is in the same situation and has an option to enlist a fourth agent, etc. The same description holds for functions as well as tasks.

We are interested here in examining the role that the contractor's client plays in the work of a subcontractor. For that purpose it is sufficient to consider the three agent case where a contractor and client have an agreement in which the contractor commits to perform a function, and the contractor instead of performing the function himself establishes an agreement with a subcontractor in which the subcontractor commits to perform the function. In that case, the contractor's client then becomes the consumer for the subcontractor's function.

We can augment our purchasing example by considering the function contract between the company president and the employee. In that contract, the president commits to purchase equipment for the employee whenever he submits an authorized request. Instead of doing the purchasing himself, the president contracts with the procurement department to do it. Hence, a subcontracting relationship exists in which the employee is the consumer, the company president is the contractor, and the procurement department is the subcontractor. Figure 1 indicates the structure of the two contracts that establish those relationships.

Main Function Contract:

Client: The employee
 Contractor: The company president
 Function Description: Purchase equipment for the employee whenever he submits an authorized request.

Function SubContract:

Client: The company president
 Contractor: The procurement department
 Consumer: The employee
 Function Description: Purchase equipment for the employee whenever he submits an authorized request.

Figure 1: Example subcontracting situation in an office

The contractor wants the function done in order to fulfill his commitment to the consumer. The commitment of the subcontractor to perform the function can therefore be considered as being to fulfill the contractor's commitment to the consumer. Satisfaction of the consumer

is a sufficient condition and important method for the subcontractor to fulfill his commitment. The subcontractor can therefore do whatever he thinks will convince the consumer that the contractor's commitment to him has been fulfilled.

The consumer therefore plays an important role in the subcontractor's work and is an additional agent with whom the subcontractor can negotiate to determine what is required of him in a given situation. As before, if the work is being done in an informal domain, then determining agreed upon interpretations of the descriptions in particular situations is an additional issue for negotiation. If the subcontractor and the consumer agree on what is to be done, then the contractor need not enter into the negotiations or even know what was agreed on because his commitment to the consumer is being fulfilled and the commitment to him by the subcontractor is being fulfilled.

If, in a given situation, the subcontractor and consumer cannot agree on the task to be done, then they both can appeal to the contractor for help. The subcontractor can argue that his commitment to the contractor does not include what the consumer is asking for, and the consumer can argue that the contractor's commitment to him is not being fulfilled. Hence, the contractor needs to enter into the negotiations only when the subcontractor and the consumer cannot agree.

For example, when the employee requests the equipment purchase, the procurement department buyer may attempt to satisfy the employee by convincing him that he should use previously purchased equipment or that he should rent equipment. He may persuade the employee to help find an appropriate vendor, and in return agree to obtain the authorizations for the purchase that are normally part of the employee's responsibility. Such localized one-time agreements between agents occur regularly in office settings, and are an important aspect of the variability and adaptability that characterize office work. Standard computer program description techniques (e.g., flow charts) are hopelessly inadequate for describing such activity.

So, we see that the consumer is a source of information for the subcontractor about what is to be done and an authority on when the task has been completed. Also, the consumer acts as a monitor for the contractor as to whether the subcontractor has done his job, since it is the consumer who cares whether or not the task is accomplished. The interdependencies among the consumer, contractor, and subcontractor discussed in this section are summarized in Figure 2.

For the consumer:

The subcontractor:
 Performs the desired task.
 The contractor:
 Settles disputes with the subcontractor.

For the contractor:

The subcontractor:
 Fulfills the commitment to the consumer.
 The consumer:
 Provides remuneration for doing the task, and monitors the subcontractor's work.

For the subcontractor:

The consumer:
 Helps interpret the task description, and Indicates when the task is completed.
 The contractor:
 Provides remuneration for doing the task, and helps settle disputes with the consumer.

Figure 2: Summary of the Consumer, Contractor, Subcontractor Relationships

The Social Nature of Procedures

Now consider a situation in which an agent has a function he wants done and a procedure describing how to do it. We will call the agent who has the function "the procedure's manager" and the function "the procedure's function". A procedure describes a method for doing a function in terms of a collection of steps to be done in a specified order, and thereby provides a means for the procedure's manager to organize a collection of agents to perform the procedure's function. That is, the procedure's manager has the option for each step of the procedure of obtaining a commitment from some other agent to do the step (i.e., of "installing the step"). If he obtains such a commitment for each step of the procedure (i.e., if he "installs the procedure"), then the agents who agreed to do the steps (i.e., the "step contractors") will do the function for him. For example, if a procurement department manager is assigned the function of purchasing equipment for employees, then he can either find or create a procedure for performing that function and install the procedure by obtaining commitments from the people in his department to be step contractors for each of the procedure's steps.

In formal domains, operation descriptions can be provided for each step in a procedure that are guaranteed to satisfy the designer's intention for the step (e.g., add x to y). Then the commitment of a step contractor is to perform the step's operation in a manner that satisfies the formal description. The contractor need not have any model of the results expected from his step or of the role they play in performing the procedure's task. His total sphere of concern is to perform the operation as specified. That is the style of procedure execution done, for example, by a typical programming language interpreter.

In informal domains, there are no guarantees that a procedure will successfully accomplish its task. Those guarantees are lost because the procedure, its task, and the situations in which it will be used are imprecisely described. Hence, procedures in informal domains are only prototypes of methods for performing tasks. They suggest a way of decomposing a task into steps, and perhaps indicate how the task is typically performed, but they do not alleviate the need for problem solving in each specific situation to determine how to perform a task. The user of an informal procedure is confronted with the subproblems of determining the meaning of the procedure in the specific situation and whether it will be applicable or effective.

A basic problem in informal domains with installing procedures to perform functions is that one must commit at the time of installation to the decomposition specified by the procedure. If indeed as we argued above, that decomposition is only suggestive and needs to be reexamined each time the procedure is used, then the strategy of installing a procedure is an ineffective means of transferring the work from the procedure's manager to the step contractors. The challenge then is to describe and install procedures in a manner that maximises their adaptability and flexibility.

Procedure Steps as Functions

An important way of meeting the challenge of compensating for the inadequacies of informally specified procedures is to add to the description of each step a description of the *function* to be accomplished by that step (i.e., the goals to be achieved and constraints to be maintained each time the step is performed). For example, add to a step described as "Submit to procurement an authorized purchase request" the function description "Whenever an employee wants equipment purchased, achieve the goal: Procurement knows the employee wants equipment purchased and has the information and authorizations necessary to make the purchase".

A function description specifies the requirements of a step without reference to *how* those requirements are to be performed and therefore provides the option of using whatever method is appropriate in a particular situation to accomplish the function's task. The agent performing a step can use the function description to evaluate whether the action described for the step is an appropriate method in a given situation, to plan alternative methods for performing the step, and to evaluate whether his actions accomplished the step.

Adding function descriptions to steps results in procedures applicable to a wider range of situations because it allows the agents performing the steps to take into consideration properties of the situation such as resource limitations and interactions with other tasks that may not have

been known at the time the procedure was designed. The work involved in using those function descriptions is significantly different from the work of performing steps described as operations. In particular, it involves subtasks of *planning* to determine a method to use, and *monitoring* to determine whether the method accomplished the function. However, an agent capable of effectively performing those subtasks can better determine the appropriateness of his results and can successfully perform his step in unexpected situations [Fikes].

Subcontracting Within Procedures

Our description of procedure installation thus far would predict that each time a procedure step is activated and the step contractor does something other than the task described in his agreement with the procedure's manager, that the contractor must obtain an acknowledgement from the manager that what he did satisfies his commitment. In actual practice in offices, there is a broad variability of behavior in the performance of procedure steps, and only rarely is that behavior accompanied by interaction with the procedure's manager (typically the step contractor's supervisor). Instead, there are frequent negotiations among the agents doing the steps of the procedure. Those agents are not generally working for each other and have made no apparent commitments to each other. How do we explain their negotiations and the role those interactions play in their work? In this section we model those interactions by extending our description of procedure installation to include the subcontracting relationships that are established among the step contractors.

We can apply our analysis of subcontracting to the performance of procedure steps by identifying the commitments made during a procedure installation and considering the "functional role" played by procedure steps. A step's functional role is the rationale used by the procedure designer for including the step in the procedure (e.g., achieve a goal of the procedure's task, satisfy a precondition of some other step in the procedure). That rationale is therefore the defining basis for the function to be performed at that step [VanLehn and Brown].

The function to be performed at each step of a procedure has a set of preconditions as part of its description. The designer of a procedure must assure that when a given step is to be performed, its preconditions are satisfied. That design goal is satisfied by including other steps earlier in the procedure that will cause those preconditions to be satisfied. The functional role of those earlier steps, therefore, is to satisfy the preconditions of the later step.

We can characterize a function's preconditions as consisting of "activation conditions", the occurrence of which signals the contractor that an instance of the function's task is to be done, and "enabling conditions", the satisfaction of which provides the context needed by the contractor to perform the task. For example, the function performed by a buyer in a procurement department is activated when he receives a purchase request and is enabled when he receives authorization to make the purchase. We distinguish, therefore, between steps whose functional role is to *activate* other steps and those whose role is to *enable* other steps.

We make use of that distinction in describing the contract that installs a procedure step. That contract contains a commitment by the step contractor to perform the step's function whenever the step's activation conditions occur and a commitment by the procedure's manager to satisfy the step's enabling conditions whenever the activation conditions occur. For example, an accounting department clerk (the step contractor) may make a commitment to his manager (the procedure's manager) to respond to vendors' invoices whenever one is received. The manager would, in turn, agree to provide the clerk with the purchase order, packing slips, and other supporting documents needed to respond appropriately to the vendor.

The procedure's manager satisfies his commitment to satisfy a step's enabling conditions by installing those procedure steps whose functional role is to *enable* that step. Hence, an agent who is performing a step whose functional role is to enable some other step is in effect a subcontractor whose consumer is the agent performing the step he is enabling. In the accounting department example above, the agents who supply the clerk with the supporting documents are subcontractors whose consumer is the clerk.

Our earlier comments about the role that a consumer plays in the work of a subcontractor therefore apply here. The agent doing the step being enabled and the agent satisfying the enabling condition negotiate with each other to determine what the enabler's task is in problematic situations, and the procedure's manager is brought into the negotiations only when they cannot agree. Also, the agent being enabled acts as a monitor on the enabler for the procedure's manager.

The analysis of subcontracting applies to any procedure step whose functional role involves providing a result to some agent other than directly to the procedure's manager. In those cases the agent providing the result is fulfilling a commitment made by the procedure's manager to the consumer of that result (or to a client of that consumer). Hence, the consumer and producer can work out together what is to be provided.

We conclude from this discussion that an important way of increasing the adaptability of a procedure is to include in the description of each step the functional role of the step. If that functional role involves fulfilling a commitment of the procedure's manager to some third agent, then the description should include the identity of that agent, and the step contractor should have access to him for ongoing negotiations.

Summary and Conclusions

In this analysis we have described a framework that identifies the sources of variability and adaptability observed in human cooperative work situations. Our basic claim is that an agent's work is defined in terms of making and fulfilling commitments to other agents. The tasks described in those commitments are merely agreed upon means for fulfilling the commitments. The agents involved in the agreement are free in any given situation to decide how and whether a given commitment has been fulfilled. Hence, nonstandard methods and outcomes may be considered acceptable even though they do not correspond to the described tasks, functions, and procedures.

We claimed that descriptions of tasks and functions result from negotiations between clients and contractors, and serve as agreed upon specifications of what the contractor is to do. In informal domains, those descriptions are necessarily incomplete and imprecise. Determining their intended meaning in specific situations is an important component of the work. That determination involves continuing negotiations as situations and questions arise in which the meaning of the descriptions is unclear.

Procedures provide a means for organizing a collection of agents to perform a function. In informal domains, procedures represent only prototypes of methods whose meaning and applicability in specific situations is unclear. Their use requires problem solving and negotiation to determine an effective method in a given situation.

Information Needed To Do Cooperative Procedural Work

This framework characterizes the information needed by agents doing cooperative work and the role that the information plays in their work. In general, it indicates that an agent needs descriptions of the task and function contracts to which he has agreed, and the functions available to him.

For each task or function commitment that an agent has made, he needs to know the agreed upon task or function description (because it provides a set of sufficient conditions for fulfilling the commitment), the agent to whom the commitment was made (so that the contractor knows whose satisfaction he is trying to obtain), and the consumer of the results of the task or function in the case where the commitment is a subcontract (because satisfying the consumer is a sufficient condition for fulfilling the commitment).

An agent needs to know the functions available to him so that he can use them as steps in plans he forms to accomplish his tasks. In order to use a function, he needs a description of its task (so that he can determine whether the function can be used to accomplish his task), its preconditions (because they describe a means for initiating performance of the task), the identity of the contractor (so he will know who he must persuade to perform the task), and the identity of the client (so he will know who to appeal to if he feels that the contractor is not adequately performing the function).

Information Needed From a Procedure Description

We have also indicated information that is needed from the description of an informal procedure in order for the procedure to be used adaptively and flexibly. The description should identify the procedure's manager (so that each step contractor knows whose satisfaction he is trying to obtain), and each step of the procedure should be described as a function (so that the step contractor can choose his own method of performing the step). If satisfaction of an enabling condition of a step is subcontracted to another step, then the description of the step being enabled should identify the enabling step and who is performing it (so that the contractor for the enabled step can negotiate with the enabler and monitor his performance). Finally, as noted above about all functions, if a step is a subcontract, then its description should identify the consumer of the subcontract (because satisfying him is a sufficient condition for fulfilling the commitment).

Implications for Office Automation

This framework is serving as a basis for our exploration of how computer-based systems can effectively participate in procedural work in offices. We have reported in earlier papers our preliminary results in this regard ([Fikes] and [Fikes and Henderson]) and will not attempt to describe our current efforts in detail here. Instead, we will conclude this paper with some general remarks on office automation to suggest the uses we are making of the commitment-based framework.

Our discussion indicates that in informal domains, "intelligent" capabilities such as planning, plan monitoring, and negotiation are required to do even simple cooperative work. Current computer-based systems that claim to automate such work in offices do not have those capabilities. They require precise descriptions of their function and how to perform it. Therefore, they can "commit" to doing only a formalizable approximation of the function desired by the client. They are incapable of performing the function in situations that do not match the assumptions of the formalization, and can not adapt their methods to account for unexpected features of a particular situation such as resource limitation changes or interactions with other tasks. In addition, they require more effort by the client to establish their task or function contract since they have no capability of interpreting vague descriptions and only very limited capabilities for recognizing situations where a description is inapplicable.

All too often, designers and installers of office automation equipment do not realize the unformalizable subtleties of the work being automated, and therefore do not anticipate the differences between what the equipment is going to do and what the people did whom it is replacing. Those differences often cause major upheavals in an organization because they change the work requirements of all the agents who interact with the equipment. A major goal of the analysis described in this paper has been to provide a model of the unformalizable aspects of office functions being overlooked by current automation efforts so that the differences in functionality introduced by the automation can be predicted and compensated for.

Automation can increase productivity in an office by supporting, as well as replacing, people in their performance of office functions. For example, the framework we have described suggests ways of supporting office work by providing agents with the information they need when they need it. It also suggests a facilitator role for a computer-based system using knowledge of who the clients, contractors, and consumers are for each task being performed. By knowing who must be satisfied by each result, a system would be able to monitor and track the performance of a task without needing to understand the methods being used or the semantics of the task itself.

Acknowledgements

Lucy Suchman is a major participant in the research effort from which this paper emerges and has made significant contributions to the ideas presented herein. Also, I would like to thank Danny Bobrow, Austin Henderson, Tom Malone, Al Newell, and Kurt VanLehn for providing valuable feedback and ideas in support of this paper.

References

- Fikes, R. "Automating the Problem Solving in Procedural Office Work" In Proceedings of the AFIPS Office Automation Conference, Houston, Texas, March 1981.
- Fikes, R., and Henderson, A. "On Supporting the Use of Procedures in Office Work" Proceedings of The First Annual National Conference on Artificial Intelligence, Stanford, Calif., August 1980.
- Fikes, R., and Nilsson, N. "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving". *Artificial Intelligence* 2 (1971), pp 189-208.
- Flores, F., and Ludlow, J. "Doing and Speaking in the Office". In Fick, G., and Sprague, R. (eds), *DSS: Issues and Challenges*. London: Pergamon Press, 1981.
- Smith, R. G. "A Framework for Problem Solving in a Distributed Processing Environment". Dept. of Computer Science, Stanford Univ., Stanford, Calif.. STAN-CS-78-700 (HPP-78-28) Dec. 1978.
- Smith, R. G., and Davis, R. "Frameworks for Cooperation in Distributed Problem Solving" *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-11, No. 1, January 1981.
- Suchman, L. "Office Procedures as Practical Action: Theories of Work and Software Design" To appear in a special issue on Office Semantics of the *IEEE Transactions on Software Engineering*.
- VanLehn, K., and Brown, J. S. "Planning Nets: A Representation for Formalizing Analogies and Semantic Models of Procedural Skills". In Snow, R., Frederico, P., and Montague, W. (eds) *Aptitude Learning and Instruction: Cognitive Process Analyses*. Hillsdale, N.J.: Lawrence Erlbaum Associates.