On Supporting the Use of Procedures in Office Work

Richard E. Fikes and D. Austin Henderson, Jr.

Systems Sciences Laboratory
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, California 94302

Abstract

In this paper, we discuss the utility of AI techniques in the construction of computer-based systems that support the specification and use of procedures in office work. We begin by arguing that the real work of carrying out office procedures is different in kind from the standard computer science notions of procedure "execution". Specifically, office work often requires planning and problem solving in particular situations to determine what is to be done. This planning is based on the goals of the tasks with which the procedures are associated and takes place in the context of an inherently open-ended body of world knowledge. We explore some of the ways in which a system can provide support for such work and discuss the requirements that the nature of the work places on such support systems. We argue that the AI research fields of planning and knowledge representation provide useful paradigms and techniques for meeting those requirements, and that the requirements, in turn, present new research problems in those fields. Finally, we advocate an approach to designing such office systems that emphasizes a symbiotic relationship between system and office worker.

Introduction

We are interested in developing office systems that would make use of a knowledge base describing *what* tasks are to be done, *who* is to do them, and *how* they are to be done. Such descriptions specify the functions of an office and how it is organized to achieve that functionality. We claim that such a knowledge base can form the basis for a broad range of system support in an office. In this paper, we discuss some of the ways in which AI paradigms and techniques are relevant to the support of office work by such computer-based systems.

We begin by describing some of the support functions we have in mind, and then address what we consider to be the primary issue; namely: what is the nature and structure of the information in such a knowledge base? We are guided in addressing that issue by considering the nature of the work that occurs in an office and how such information is used in that work.

We first argue that the work involved in carrying out office procedures is different in kind from the "execution" of a procedure that one might expect by drawing analogies with the behavior of a computer executing a program. We illustrate and support this claim by presenting a typical case of office work and analyzing the actions that take place there. From this argument we derive a requirement for systems which support office work; namely, that they be flexible enough to support the variety of behavior occasioned by the unpredictable details of particular situations.

We then turn to the relevance of AI for achieving this functionality. We develop the idea that the paradigms from the AI literature for automatic planning and execution monitoring of plans provide a more suitable alternative to the procedure execution model of office work; and furthermore that the demands of supporting office work require extensions to those paradigms. Second, we argue that the knowledge representation problems presented by the open-ended office domain are unsolved and challenging. We suggest that they can be attacked by the use of specialization-based representations and facilities for storing "semi-formal" structures in which uninterpreted text is intermixed with data whose semantics is understood by the system.

Finally, we argue that the whole enterprise of supporting office work can only hope to succeed if we regard the office systems as functioning in a partnership with the office workers. Due to the open-endedness of the domain, the system cannot hope to "understand" the full import of the information which it is handling, and so must rely on human aid. Furthermore, to fully support the users, the system must be able to represent, although not necessarily understand, any of the information in the domain. We conclude by advocating an approach of "symbiotic processing" between system and office worker and the use of AI techniques in constructing systems to support office work.

Supporting the Production and Use of Procedural Descriptions

We begin by considering some of the ways in which computer-based office systems could facilitate the effective production and use of descriptions of what tasks are to be done, who is to do them, and how they are to be done. There are two groups of people whom an office system dealing with such descriptions can support: the producers and the users. However, the production and use phases are often tightly interwoven, with the same people often involved in both (despite what managers may choose to think).

The producers of these "what, who, and how" specifications (typically managers and planners) are engaged in a process of organizing the work in the office so that the office's goals and commitments will be met. That process involves defining the tasks to be done, designing procedures for doing those tasks, and assigning individuals to carry out those procedures.

A system can support these specification processes by providing a descriptive framework in which to express the specifications and by helping to manage the complexity that arises from the interactions of the tasks, constraints, procedures, and policies being specified.

The descriptive framework would provide a guide as to what information needs to be specified (based on the intended purpose and uses of the specifications) and a terminology for expressing that information. For example, the system might provide a template for describing a task that would include fields for the task's goals, inputs, outputs, responsible agent, activation conditions, etc., and a description language for filling those fields. The system could also indicate direct implications of a description, such as the subtasks implied by a task description of recognizing the task's activation events, obtaining the task's inputs, or communicating its outputs.

The system could aid in managing the complexity of the specifications primarily by monitoring interface requirements among interacting components to help assure that those interfaces are well specified and the requirements are met. For example, if the description of a task included the source of each of the task's inputs and the destination of each of its outputs, then the system could alert the specifier when those input-output connections between tasks are inconsistent (e.g., when some input is not an output of the source task), and could prompt for a specification of how those input-outputs are to be communicated.

The other group that the system can support is the users of the "what, who, how" specifications. That support would include facilities for accessing the specifications in useful ways, for adding informal notes to the specifications, for monitoring the carrying out of tasks, and for doing some of the procedural steps.

One useful way in which the system would act as an information source is in providing "how-to" information to a person who has a task to do and doesn't know how to carry it out. For example, when a secretary is given a task that he is not familiar with (such as "order some business cards for me" or "obtain a consulting contract and arrange the travel plans for this person"), the system could provide him with a description of what needs to be done, how to do it, and who the people are who will play a role in getting it done.

One could amplify the system's usefulness in this role as a how-to resource by allowing its users to add informal notes to the descriptions. Then the system also becomes a repository for the accumulated societal wisdom concerning the realities of actually carrying out the tasks.

The functionality we have discussed thus far has only required knowledge of the procedures in general. The system's usefulness can be further enhanced by providing it with the capability of knowing about *specific* instances. With this capability the system could participate in the work in one of two ways: by tracking the progress of tasks, and by carrying out tasks itself.

A task tracking facility would allow the system to:

* be a source of information regarding the task's status, history, and plan;
* send requests to the agents who are to do the next steps, and make available to them a description of what they are to do, what has been done, and pointers to the relevant documents;
* send out reminders and alerts when expected completion times of task steps have passed; and
* ask for intervention by the appropriate agent when problems arise.

A system which is tracking tasks in this way is participating as a partner in doing the work. Once that symbiotic relationship has been established between system and office worker, there are many steps in office procedures that the system could do itself. Such tasks would certainly include communication activities (e.g., using electronic mail), and maintenance of consistency in structured information bases (e.g., automatically filling in fields of electronic forms, see [Fikes], 1980).

Office Work and Office Procedures

With this class of intended systems in mind, we now turn to the question of the nature of the office work that we hope to support. In so doing, our goal is to determine the nature and structure of the information needed by our intended systems to support that work.

A Procedure Execution Model

A common model of office work considers an office worker to be a processor with a collection of tasks to be done and a procedure for doing each task. The work, in this model, involves executing each procedure and "time sharing" among them. However, studies of office work reveal a complexity and diversity of behavior far beyond what would be predicted by this model (For example, see [Suchman], [Wynn], and [Zimmerman]). In this section we explore the nature of this apparent discrepency as a way of exposing characteristics of office work that we think have important implications in the design of systems to support that work. {Note: The potential usefulness of this discrepancy was suggested to us by [Suchman].}

First, consider the office worker's ongoing (meta-)task of determining how to allocate his resources among the collection of tasks assigned to him. His work involves the planning, scheduling, and context switching associated with time sharing among those tasks. However, he can exercise options in carrying out his scheduling task that are not available to the scheduler in a computerized time sharing system. In particular, he can modify tasks themselves. For example, the worker may choose to

* ignore some of the requirements of a task,
* renegotiate the requirements of a task,
* get someone else to do a task,
* create and follow a new procedure for doing a task.

Hence, office work includes, in addition to the carrying out of tasks, the determination of when a task should be done, how the task is to be done, and whether the task will be done at all.

{Note: The office worker also has goals other than the completion of assigned tasks. For example, he has career goals (try to get ahead in the company), company goals (maximize profit), personal goals (keep from being bored), social goals (be regarded as good company), and societal goals (be honest).}

Second, we take it as obvious that the domain with which office systems must deal is open-ended: truly anything may become relevant to the workings of offices at one time or another. This fact implies that a procedure which implements a task is necessarily an inadequate description of all the actions which must be done to achieve the task's goals in all the various situations that can (and inevitably *will*) occur. That is, at the time the procedure is defined which implements a task, one cannot predict either the range of situations that will be encountered in an office or the extent of the knowledge, activities, and considerations that will be needed to carry out the task. Hence, for any given procedure, situations may occur in which the procedure does not indicate what is to be done, or in which what is indicated in the procedure cannot be done. For example, situations may occur in which:

* a case analysis in the procedure does not include the current situation,
* assumptions about the form and availability of inputs for a step are not met.
* resources required to do a step are not available,
* the action described in a step will not have the intended effects.

The procedures associated with each task serve as a guide in that they indicate one way of doing the task under a particular set of assumptions. The office worker has the responsibility of deciding in each particular situation whether the procedure's assumptions are satisfied and whether he wants to carry out the task in the way specified by the procedure.

Third, the office worker has the problem of interpreting abstract specifications of what is to be done. For example, it is not uncommon in procedure specifications to find phrases like "include any other pertinent data", "send forms approximately six weeks in advance of the deadline", and "arrange for employee to receive benefit checks at home". What is "any other pertinent data", when is "approximately six weeks in advance of the deadline", and how is one to "arrange for the employee to receive benefit checks at home"? The specification of the procedure doesn't say. Hence, a necessary part of the work of following office procedures is determining what the abstract specification implies is to be done in each particular case.

We conclude from these observations that the standard model of procedure execution is inadequate for describing office work. The original procedure specification serves only as a guide in this process and can be thought of as the first approximation to a plan for the particular task at hand. It is the responsibility of the office worker in each particular case to determine the suitability of the procedure, fill in missing details, and modify it where necessary to achieve the goals of the task.

An Example Of Office Work.

To make these points more tangible, let us now look at an example of the everyday work which goes on in an office (This is an elaboration of an actual case of office work reported by [Wynn], p. 49). This example exhibits the problematic nature of the work, and the need for reflecting upon the specifications of the procedures.

Xerox sells supplies for its copiers - paper, toner, and such. Customer orders for supplies are taken over the phone by a "customer order entry clerk" (COEC). The COEC talks to the customer and fills out a form which records the order. This order form is used by other clerks to bill the customer and to deliver the supplies. The form has a field for recording the address at which the copier is located, and there is an underlying assumption that this is the address to which the supplies are to be delivered.

In the particular incident of interest, the customer informed the COEC that he could not supply an address for the copier because it was located on an ocean-going barge(!). This situation, of course, raised the question of what should be put into the address field of the order form.

The clerk realized that the intended use of the address was to specify where the supplies were to be delivered, and that because the copier was on a barge that the needed address was dependent upon when the delivery of the supplies was to be made. Since he could not predict that date, he obtained from the customer a telephone number that could be called when the delivery was about to be made to obtain the address of the current location of the barge. He entered that telephone number into the field of the form and added a notation indicating how the number was to be used.

The story continues: When the billing clerk was making up the bill, the question arose as to whether or not to charge California sales tax. The answer depends on whether or not the supplies were to be delivered out-of-state. The address field of the order form was examined, as per the usual procedure for answering the question, and of course no information about the state was available. What now?

The billing clerk read the notation, called the telephone number, and ask the respondent whether the delivery was to be made in or out of California. Again, the date of the delivery was crucial in determining the answer. However, the billing clerk knew approximately when the supplies would be available, and therefore was able to determine from the person called that the delivery would be made in California, even though the precise delivery address was still not known. An addition was made to the information in the address field of the order form indicating that the delivery was to be made in California, and the bill was prepared and sent.

Finally, the shipping clerk, with the supplies in hand, repeated the telephone call when preparing the shipping label. The address was then known, the address was added to the form, and the supplies were delivered.

Analysis of This Example.

What we have here is a case of a blown assumption. The procedures in which all three of these clerks were playing a role were designed on the assumption that copiers do not move and therefore have a fixed address. The particular case violated that assumption.

The COEC was confronted with a problem because he could not carry out a step of a procedure (i.e., he could not fill in an address for the copier). There are several things he could have done at that point, including ignoring the step or telling the customer that unless he provided an address that the order could not be taken. Instead, he chose to stop "executing" the procedure and to step back and reason about it. In particular, he considered what were the intended uses for the problematical address; i.e., what was the goal of filling in the form's address field. Using that information, he created a *plan* involving both himself and the shipping clerk that was within the spirit, although not within the letter, of the established procedures. That is, he devised an alternative that would satisfy the goals of the intended users of the address, as he perceived them. Hence, those goals were the crucial information that the COEC needed in order to determine suitable alternative actions when the unexpected situation occured.

Note that the COEC was apparently not aware of the billing clerk's use of the address field to determine state sales tax. Hence, the COEC's alternative plan did not indicate how the billing clerk was to deal with this situation. The billing clerk, like the COEC, was confronted with a problem of not being able to carry out a step in a procedure (because the address field of the order form did not contain an address). Again, as was the case with the COEC, he did not ignore the problem or reject the situation as unacceptable. Instead, he attempted to find suitable alternative actions that would satisfy his task goals and allow the billing to proceed. His planning involved understanding the alternative procedure for the shipping clerk that had been formulated by the COEC, and realizing that he could use the telephone number included in that formulation to satisfy his goals.

Consider the nature of the information involved in this example. Note the *unpredictability* at the time the form was designed of the kinds of information that would be put on the form. Note also that the information on the form regarding the address was *changing* throughout the procedure. First there was a note describing a procedure for obtaining the address, then a *partial* address containing only the state added to that note, and finally a complete description of the address. Another form of partial description that played a role in the example was *approximation*; in particular, the clerks' knowledge of the approximate delivery date. The strength and certainty of those approximations determined when and to what extent the delivery address was obtained.

Supporting the Work Requires Flexibility

We have presented the idea that the work that *actually* goes on in offices is not routine. It consists of many particular cases of applying the given procedures to the details of those cases. This work involves dealing with unsatisfied assumptions, doing planning, understanding goals, and using information that is partial, approximate, and changing. The illusion that office workers execute procedures in a manner that is analogous to the way computers execute procedures ignores these realities of the situation. Given that picture of office work, we now turn our attention to the requirements placed on the design of computer-based systems to support such work.

A primary design challenge is to find ways of providing the flexibility that is needed to allow for the application of established procedures to the circumstances of particular cases. With respect to information being supplied to the system by users, this flexibility involves dealing with cases where information is missing, information is provided in unexpected forms, and/or information in addition to what was expected is supplied. With respect to the procedural steps being carried out, this flexibility involves dealing with cases where steps are omitted, steps are done in different order, and/or additional steps are done. When office systems lack the flexibility to deal with these contingencies, they severely restrict the options of their users and thereby become yet another bureaucratic barrier to be overcome in "getting the work done".

Consider, for example, an electronic forms system for supporting the work of the COEC. When the "copier on a barge" problem arose, the COEC would have needed that system to be flexible enough to allow entries other than addresses in the form's address field. In particular, the COEC needed to be able to say to the system, in effect, "I can't give you an address in this case. Instead, I'll give you a note for the shipping clerk." If the system also used its descriptions of the procedures being followed to provide instructions to the clerks regarding what is to be done, then the system would need to be able to accept the COEC's decision to omit the step of providing an address in the form's address field, and to incorporate into the shipping clerk's procedure an instruction to read the COEC's note the first time the address was needed.

In addition to being able to accept such alternative inputs, any of the system's facilities for doing computations based on those inputs (e.g., to compute the state sales tax on customer orders using the address on the order form) must be designed to deal with cases in which those inputs have some unexpected structure or are not available at all. The challenge in the design of such processing facilities is to provide ways for the system, in cooperation with the office worker who is being supported, to overcome the difficulty posed by the failed computation so that work on the task can continue in a productive manner.

One often hears the argument that this need for flexibility and variation from established procedures could be overcome by doing a more thorough analysis of the office tasks and thereby producing complete procedures that would cover all of the cases that could occur. Our claim is that because of the open-ended nature of the office domain, one cannot anticipate all of the situations that will occur in the carrying out of a given task, and therefore cannot totally characterize the inputs that will be available or the actions that might be taken to satisfy the task's goals.

The Relevance of AI to Supporting the Work

An Alternative Model: Planning and Plan Execution

The observations we have presented on the characteristics of office work have led us to seek an alternative to the procedure execution model to guide us in building a knowledge base for office support systems. We have found what we think to be a suitable alternative in the paradigms from the AI literature for automatic planning and execution monitoring of plans. That is, we take the viewpoint that we are confronted not so much with the problems of representing and supporting the execution of procedures, but with the problems of representing plans and supporting the monitoring and replanning that occurs during their execution.

This viewpoint provides us with a conceptual framework for understanding the use of procedures in an office, an understanding that we feel is critical to dealing with the problems of designing systems to actually support that work. In the following paragraphs we present some of the key aspects of this point of view and discuss the ways in which it suggests that a system could provide useful support.

The basic requirement on a data base describing these plans are that they provide the information needed to monitor a plan's execution and to do whatever replanning might be required. What information is needed during those operations? By referring to the planning paradigm used in the STRIPS systems ([Fikes], et al, 1972), we obtain the suggestions that execution monitoring requires descriptions of the expected results of each operator, the intended use of each operator result, the preconditions of each operator, and the assumptions made by the planner about the world at each step of the plan. Planning involves the use of descriptions of the current state of the world, the operators available as potential plan steps, and the goals to be achieved by the plan.

This planning paradigm characterizes some of the information that might be useful in the doing of office tasks and therefore suggests what to include in the description of office tasks and their associated procedures. In particular, it suggests the inclusion of information about task goals, intended uses of operator results, and precondition assumptions of operators. For example, the COEC employed information regarding the intended use of the address by the shipping clerk to determine an alternative plan when the address was not available. If the COEC had also known about the billing clerk's intended use of the address, then he would have tried to obtain the information needed for that use (i.e., the state in which the delivery would be made) and, if successful, would have eliminated the difficulties that the billing clerk had in the example.

One of the major ways we see for a system to provide support is by serving as an information source for office workers during the execution of plans and during any replanning that may be required. Hence, the planning paradigm suggests what information to include in the system's representation of the tasks and procedures, and provides us with a basis for characterizing the questions that the user may ask of the system.

The paradigm of hierarchical planning (e.g., see [Sacerdoti]) also applies here and can be used in our characterization of office work. That paradigm would suggest that we consider each individual step in a plan as being a task with its own inputs, enabling conditions, goals, etc. There may or may not be a plan associated with any given step's task. In the cases where there are, these plans form a tree and then get combined in various ways to form a planning network. Such a network represents a hierarchical plan, where the top of the hierarchy decribes a top level task and each sucessive level of the hierarchy describes increasingly detailed subtasks. In the standard non-hierarchical planning case, there is a plan for each step and each plan consists of a single operator; hence, there is a one-to-one correspondance between plan steps and operators. In the hierarchical planning paradigm and in the office, that one-to-one correspondance need not exist.

Hierarchical planning networks appear to be an important device for representing office plans for several reasons. They are a useful structure for representing the task-subtask and goal-subgoal relationships that need to be known about during execution monitoring and replanning, and they provide the basic descriptive framework for indicating how the work is to be organized.

Also, since there are effectively no primitive operators in the office, there is a need for describing office plans at varying levels of detail, depending on the specific needs of the describer and users of the descriptions. That flexibility in the level of detail of specification is therefore needed in an office system's representation facilities. The system can then be involved in the office work at varying levels of detail. For example, the system may know that a travel request needs to be authorized, but know nothing about the subtasks involved in obtaining the authorization. Such flexibility is also an important tool for enabling the system to participate in situations that it does not understand. For example, if a plan that the system is helping to monitor fails, the user may not describe to the system the alternative plan he decides to use. However, the system knows about the goals of the original plan that the alternative must also satisfy and can therefore monitor the accomplishment of those goals, even though it now has no model of *how* those goals are being achieved.

An important way in which office work motivates extension of current AI planning paradigms is that office work is done in a multi-processor environment. That is, one can consider each agent (person or system) in an office to be a processor that is accomplishing tasks by carrying out plans that have either been given to it or created by it. Any given plan may involve the participation of several agents, each agent acting as an independent processor executing one or more of the plan's steps.

The processors make commitments to each other regarding the goals they will achieve during the execution of a plan [Flores]. Therefore, in creating a plan to be carried out by multiple agents, the commitments that those agents will make to each other are a crucial part of a multi-processor plan.

Furthermore, the commitments an agent has made and that have been made to him form for him a set of constraints within which he must work. In particular, these commitments form the context in which replanning takes place. in that any new plan must satisfy those commitments. However, the agent also has the options during replanning of renegotiating the commitments he has made or of ignoring them altogether.

Any system that is to participate in the replanning process needs to support this commitment-based view of planning and take them into consideration. In particular, a system could help an agent keep track of the commitments he is involved with that relate to a particular task and indicate his options for changing them during replanning. To support this tracking, the system's plan representation needs to include for each plan step both the commitments made by and to the agent responsible for doing the step.

The multi-processing nature of office work also implies that steps of a plan can be done in parallel. Hence, representations for office plans need to allow specification of partial orderings for plan steps. That requirement and the pervasiveness of replanning that occurs during plan execution suggest that task descriptions should include a set of necessary and sufficient "enabling conditions" for beginning the task so that it can be determined when the task can be begun irrespective of the order or nature of the steps that achieved those conditions.

Up to this point we have not considered perhaps the most immediate question that arises out of looking at office work from a planning point of view: to what extent can we expect a system to automatically do the planning and replanning that is needed for office tasks? The primary limitation on such automatic planning seems to be the open-endedness of the office domain. That is, the extent to which there are considerations relevant to the formation of the plan of which the system has no understanding will limit the system's effectiveness at determining an appropriate plan. For example, the system may not know about possible operators in the situation, the costs or chances for success of renegotiating or ignoring existing commitments, other goals that interact with the task's goals, or the implications of an unexpected situation. These limitations have lead us to focus on a symbiotic relationship between system and office worker during planning and replanning, where the system plays primarily a role of supporting the planning being done by the users by helping represent, manage, and communicate the resulting plans.

In conclusion, then, we are claiming that

* a multi-processor hierarchical planning model is a useful one for understanding office work and therefore for structuring an office system's knowledge base, and
* the demands of supporting office work motivate new research in multi-processor commitment-based planning.

Knowledge Representation Challenges

We turn now to the demands that supporting office work makes on the representation of the knowledge which systems have of the office domain. We then discuss two techniques that arise from work in the AI community that provide particularly promising starting points for confronting those demands.

The single most salient demand of such representation schemes is that they be able to respond to the need for change in conceptualization of the work. As we have seen, the domain of office work is inherently open-ended (e.g., before the barge case there was no notion of addresses being time-dependent; afterwards there was). Consequently there is no way to anticipate the full range of subject matter with which the system will have to deal. In consequence, the representation scheme must be able to handle any conceivable conceptualization which, over the course of the life of the office, the users of the office system choose to enter into the system.

Furthermore, as time passes, this conceptualization will change to meet the changing understanding of the office domain which the users of the system have. Sometimes these changes will be small; at other times, there will be major "re-thinkings" of the information. The system must not only be able to represent this changing pattern of thought, but must also be able to simultaneously represent the pattern of changing thought: to support office work, it will have to be able to support the history of what has happened previously, and consequently will have to be able to hold simultaneously the old conceptualizations - for supporting the understanding of the past, and the new.

The second demand placed by office systems on the mechanisms for representing the knowledge which is within them is to support partial knowledge of their domain. This incompleteness comes in at least three forms: the support of a subset of some expected body of knowledge (e.g., the state in which an address is may be known, but nothing more); the support of an abstraction of the knowledge (e.g., the supplies ordered are a paper product, but which one is unknown); and the support of an approximation (e.g., the date of delivery is between mid-April and mid-May). In particular, this ability to support partial knowledge will permit the entry into the system of all that one knows, even though that may only be part of what is desired in a complete description.

A resultant demand is that the mechanisms which access information must be prepared for the expected information not to be there (e.g., the state portion of an address is missing from the available information when the billing clerk tries to bill the barge-owner). This preparation involves the representation scheme in at least being able to detect the absense, and further, in having some means of coping with the resulting problems.

The third demand results from the fact that knowledge of things often accumulates over time. Sometimes such an accumulation of partial descriptions can be reformed into a coherent whole. But more often the pieces are better retained as independent, un-coordinated facts. Indeed, rather than think about *the* description of an entity, it is often useful to view the object as having *multiple* descriptions. Thus, for example, the knowledge about the address of the copier might at some point include three distinct descriptions of the address: as having California as its state portion, as being a changing thing, and as being something which can be furthur determined by carrying out the procedure "call this number and ask".

The final demand arises from the expectation that the system should provide a general model of office work. This model could be crafted by experts on the organizational structuring of offices, and would then be available as a conceptual framework to support the description of more particular details. In fact, these concepts become the *terms* in which the details are not only described, but understood. Thus, for example, the concepts of task, goal, procedure, plan, agent, post, authorization, commitment, and data repository might be provided as a very general framework for modeling offices. Particular offices would have their own particular tasks, goals, etc.

These demands pose challenging research problems in knowledge representation which we are not claiming to have solved. However, we discuss in the following paragraphs two starting points for confronting these problems that look particularly promising to us and that we are using in our work.

Our first starting point for responding to the demands of supporting office work is the use of a specialization-based knowledge representation formalism (see, for example, [Brachman]). This, and similar, schemes for formally and precisely representing knowledge take as their goals the first three of our needs: support for representation of a changing conceptual structure, partial description, and multiple description. The major structuring principle for their representations is *definition by specialization*: any concept in a knowledge base can be taken as a basis for defining a new concept which is a special case of the old one. Thus, describing the details of a particular office can be done in this formalism by specializing more general descriptions of offices. This specialization can be done in steps, thus permitting the tailoring of the conceptualization in various ways to produce progressively less abstract descriptions. In the end, the most specific descriptions are understood through their place in a taxonomic lattice (a concept can specialize more than one abstraction) of more abstract concepts. Not only does this enrich the understanding of the domain, for one can understand similarities between concepts in terms of common abstractions, but it also provides locations for attaching knowledge about abstractions which will immediately apply to all special cases of those abstractions.

Our second response to the knowledge representation needs presented by the office domain is the use of *semi-formal* descriptions. That is, we are using the formal knowledge representation mechanisms in a style which permits us to capture information that the system does not "understand" in such a way that it can be usefully employed by human users of the system. For example, paragraphs of English prose or diagrams can be associated with concepts; the only use the system will be able to make of them will be to present them to a human user, and permit him to read and modify them.

We view these mixtures of formal (the structure is understood by the system) and informal (the structure is understood only by humans) descriptions as an essential "escape valve" for our knowledge representation systems: if it were required that the system had to understand the conceptualization underlying all the informaion in its knowledge structure, then the cost of entering information which is currently beyond the system's understanding would be very high. Instead, by escaping into informal description, the system can still be used as a repository for all of the information about the situation at hand, and yet permit the work to proceed. The system becomes primarily a communications device for supporting interaction between people carrying out office work. However, because the informal descriptions are represented in the same description formalism as the formal descriptions, they can be integrated into the knowledge base in a consistent manner.

An Approach to System Building: Symbiotic Processing

In the account given above of our vision of the behavior and properties of a system to support the real work of offices, there has been repeated reference to interaction between the system and its human users. This will be an important aspect of successfully completing the planning which must be done in office work. Also in representing knowledge within the system, we have argued that a "semi-formal" mixture of information will be important for achieving practical systems. That is, it takes both man and machine to understand the information held within the system.

Changing the emphasis, we prefer to think of the humans and the computer as cooperating processing engines carrying out the "computations" of the system in partnership. The idea here is that both processing engines, each with its own processing capabilities, knowledge, and memory structures, are essential to getting the task done. Neither could effectively do the work without the other.

In biology, such interdependence is called *symbiosis*, and a system composed of two or more interdependent organisms is called a *symbiotic system*. We therefore use the same term to refer to the sort of office systems envisioned here: there is a symbiosis of human and machine.

Why is it that this quite obvious co-operation between man and machine has not so far been the dominating pattern of computer use in office (as well as other) systems? Our theory is this: when batch processing was the only economically feasible form of business (and therefore office) system, such interaction was impossible - the human partners were simply not around when they were needed to help the computer in its tasks. However, the pervasiveness of the belief among designers of computer systems that the procedures in offices were "routine" obscured the need for truly cooperative interaction. A result of this belief, and of the introduction and widespread use of batch processing systems in the business environment, has been the establishment and buttressing of the now-accepted notion that there is something fundamental about partitioning the world into *routine* cases and *exceptions*.

We view this distinction between routine and exception as quite artificial: routine cases are those cases where no action is required of the human partner in the symbiosis; exceptions are everything else. And, in fact, it is even worse than that: the distinction breeds viewing the world this way, which in turn enhances the distinction. To get the proportion of routine cases up to the point where batch systems could be justified, many cases which require some small amount of human processing are handled by forcing the human processing to be done before the cases are "entered into" the system. This is often done at the expense of some capacity of the system (as a whole) to handle not-quite-standard cases: these cases are wedged into the mold of the "routine". And when exceptions are not even permitted - when everything has to go through the system as a routine case, the mold can well become a straight-jacket.

We propose that office systems need not make this distinction between the routine and the exceptional. Instead, it should be possible, and is desirable, to return to the "good old days" when all cases were processed in the same way, some with more effort than others. We believe that, armed with the understanding of offices presented here, and supported by studies in the AI fields of automatic planning and knowledge representation, a modern version of the "good old world" can be achieved through systems built around the notion that all cases are handled by the powerful symbiosis of humans and machines.

Acknowledgements

We would like to thank Lucy Suchman and Eleanor Wynn for early contributions to the ideas in this paper. In particular, they made available to us transcripts of their interviews with office workers and observations of office work, and helped open our eyes to what those interviews and observations had to say about office work. We would also like to thank Lucy for her insightful participation in the continuing research that has led to the observations in this paper.

References

Brachman, R. et al. *KLONE Reference Manual*, BBN Report No. 3848, July 1978.

Fikes, R. E., Hart, P. E., and Nilsson, N. J. "Learning and Executing Generalized Robot Plans". *Artificial Intelligence*, 3(4), winter 1972, pp 251-288.

Fikes, R. E. "Odyssey: A Knowledge-Based Personal Assistant". To appear in *Artificial Intelligence.*

Flores, F. Univ. of California at Berkeley, personal communication.

Sacerdoti, E. D. *A Structure for Plans and Behavior*, New York: American Elsevier, 1977.

Suchman, L. A. "Office Procedures as Practical Action: A Case Study". SSL Internal Report, Xerox Palo Alto Research Center, Sept. 1979.

Wynn, Eleanor Herasimchuk "Office Conversation as an Information Medium". Department of Anthropology, University of California at Berkeley, May 1979.

Zimmerman, D. H. "The Practicalities of Rule Use". in J. D. Douglas (Ed.), *Understanding Everyday Life*, Aldine Publishing Company, Chicago, pp 221-237.