Simulating Organizations

Computational Models of Institutions and Groups

Foreword by Michael D. Cohen

Edited by Michael J. Prietula, Kathleen M. Carley, and Les Gasser

1998 AAAI Press / The MIT Press
Menlo Park, California, Cambridge, Massachusetts, London, England

9

An Approach to Modeling Communication and Information Technology in Organizations

David J. Kaplan and Kathleen M. Carley

Imagine a manager trying to make technology purchasing or scheduling decisions where a wide variety of technology and work arrangements are possible. Should the manager select a collaborative setting? Perhaps an individual with many years of experience would be best. Maybe the presence of video in a collaborative setting would be helpful; or, perhaps electronic manuals should be the single most important item. Is the manager considering quality or time as the decision criteria, or is the manager interested in minimizing the physical handling of delicate machinery? Should the company invest in wearable technology or desktop technology?

To answer such questions, managers can consult many practical and academic articles that discuss the relative impacts or costs and benefits of various telecommunication devices, software, computers, and so forth. Indeed many excellent scientific articles using either experimental or field approaches discuss the relative social and information processing impacts of various information technologies[1] (Fish, et al. 1993; McLeod 1992; Sproull and Kiesler 1991; Kraut, Galegher and Egido 1990; Carley and Wendt 1991; Eveland and Bikson 1988). While such research is valuable, it may not meet a specific manager's needs. Experimental research is limited by the number of technologies that can be simultaneously compared and the complexity of the organizational setting and task that can be reasonably examined. Field research is limited in that researchers have little control over the relative impact of a wide variety of exogenous factors on their study's results. These problems are perhaps even more acute—and certainly added complications arise—when one is dealing with "futuristic" technology that is still under development. One example of futuristic technology is wearable technology.

Wearable technology is a general term for a collection of devices that incorpo-

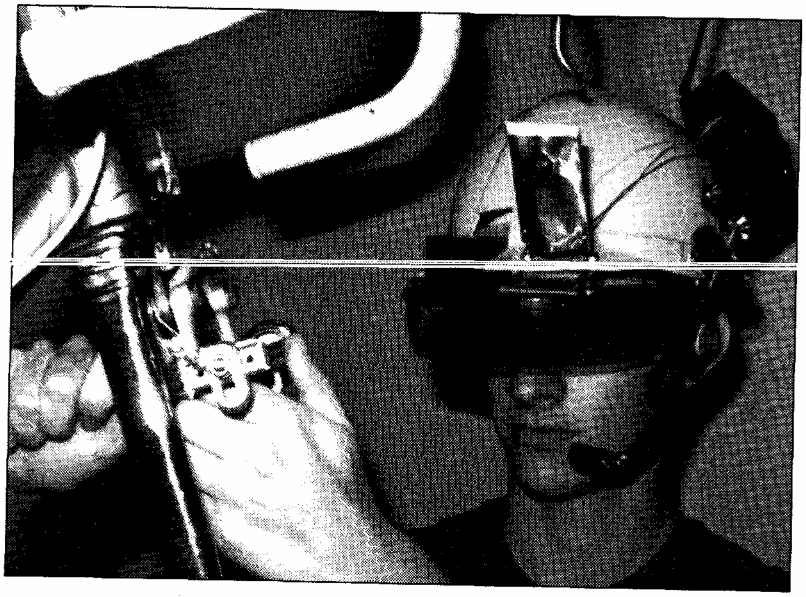

Figure 1. Example wearable system.

rate into wearable units one or more information technologies, such as a computer, telephone headset, radio transmitter, or video camera. Wearable technology should increase worker mobility and admit more rapid access to a remote expert or on-line manuals. For example, a wearable telephone headset with radio transmitter allows hands-free operation as well as increased user mobility. Heavy, difficult-to-manage paper documents can be replaced with wearable computer-based manuals whose output is displayed on a small, yet high resolution, head-mounted display, where manipulations are conducted with voice commands or simple control devices. An example of a wearable device is shown in figure 1. With such wearable technology, maintenance personnel, for example, do not have to carry manuals, have hands free for repair, and can contact a remote expert with questions when discrepancies occur between their observations and the on-line manual.

The use of computational techniques comprise an alternate approach to experiments and field studies for examining the potential impact of information technologies on organizations—particularly futuristic technologies. Questions about the relative impact of different and futuristic information technologies on organizational performance could be addressed if we had a computational framework where the organization, the individuals, the task, and the information technologies were simultaneously modeled. Such a model would be advantageous because it would allow us to examine the results of complex interactions among adaptive individuals, the organizational environment, and the information technology being employed. By using such a computational model, we could advance theories of how changes in any one of these complex factors might affect the others and overall organizational performance. In this chapter, we propose such a model, describe its necessary features, and illustrate its possible results. We are not arguing against the use of experiments and field studies for examining the impact of information technologies in organizations. On the contrary, we are arguing that a computational approach can, through theory development and initial exploration, help researchers determine which empirical investigations are most important.

The communicating and information technology model (COMIT) is such a computational model. COMIT is a prototype. It comprises a first step toward integrating, into a single framework, a model of the organization as an information processing agency, a model of individual action, a model of task, and a model of information technology. In describing this model, we point to critical issues in the interrelationship among organization, agents, tasks, and technology that are not well understood. We illustrate the type of results that are possible from such models by contrasting organizational performance under four different information technology environments — the individual acting alone with only a hard copy manual, the individual acting alone with an electronic manual, the individual interacting with a remote helper with an audio and hard copy manual, and the individual interacting with a remote helper with an audio, video, and electronic manual (the wearable technology condition). These conditions are interesting, in and of themselves, as they represent extremes in information technology possible with current technology. The value of this work includes a characterization of the complexity of such models, an extended checklist of features that these models should have, and an illustration of initial results.

COMIT seeks to simulate both macro and microlevel views of the organization with an emphasis on cooperative work. Cooperative and collaborative work bring together issues of agency, technology, task, and the impact of the organizational environment (Malone 1987; Kraut and Streeter 1995). Cooperative work can be supported by a variety of information technologies, such as telephone, video conferencing, or shared electronic manuals (Clark and Brennen 1991). These technologies allow groups of organizational actors freedom from copresence, admit access to remote experts, and alter the rate at which information can be accessed. COMIT must be flexible enough to represent a variety of organizational designs and tasks (such as those discussed by Levitt et al. [1994]; Decker and Lesser [1993]; Carley [1992]; Carley, et al. [1992]; and Cohen [1992]).

Our ultimate goal is to create a computational framework that integrates our understanding of individual decision making, organizations, and information technology and that can be used to predict some of the information processing effects of new information technologies and/or alternate organizational designs. Models that include aspects of organizational performance, individual decision making, and information technology exist (Masuch and LaPotin 1989; Carley 1992; Kaufer and Carley 1993; Decker and Lesser 1993). However, little work has been done to integrate these three aspects into a single composite model (Carley and Prietula 1994). An important exception here is the virtual design team (VDT) (Cohen 1992; Levit et al. 1994), where organizations are modeled as networks and information technologies affecting the rate and type of information that is transmitted. In the VDT model, individuals cannot learn; however, their experiences may vary. Further, within VDT order of information processing is the primary task. Essentially, VDT combines PERT chart information with information on the organizational hierarchy. COMIT is similar to VDT in its representation of task and experience and in its assignment of needed actions to steps. However, COMIT differs from VDT in a number of ways: there are only one or two individuals; the task is composed of a set of sequenced steps; each step has a set of subtasks; steps or substeps can be skipped; and information technologies alter the rate at which individuals can acquire information, whether individuals can access other individuals, and the communicative features of that interaction.

The COMIT Model

In this section we will describe the model. We begin with a brief discussion of the simulation's purpose, and then describe its process, paying special attention to the way actions are selected and performed. Next, parameters of the model are defined, followed by a discussion of each type of parameter. This section concludes with a brief example of a fairly skilled worker performing a diagnosis and repair task, aided by a reference manual.

Purpose

COMIT allows researchers to examine the performance of a small organization (one or two actors) doing a hierarchically decomposable task using various information technologies. The top-level process and parameter categories for COMIT are shown in figure 2. Illustrative information on each category is provided in the figure. To use COMIT, users adjust a set of parameters. These parameters describe the task representation, the agent or agents working on the task, the actions capable of being performed by the agents, the information technology used by the agents, and the type of descriptive output measures desired for analysis. After the parameters are adjusted, COMIT performs a virtual experiment with statistics about performance reported upon completion.[2]

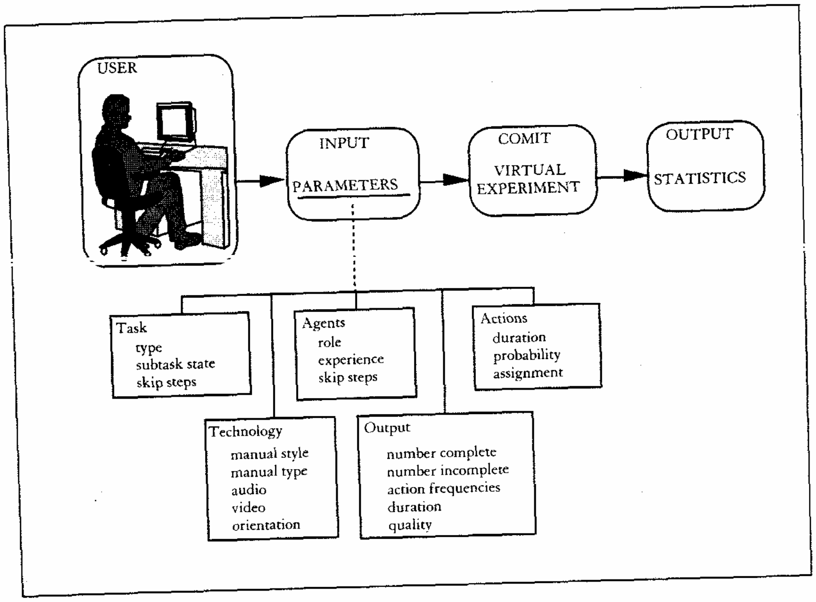

Figure 2. Overview of COMIT.

The COMIT Process

The process performed by COMIT is illustrated in figure 3. First, COMIT reads the set-up files provided by the user. These files specify task, agent, action, information technology (including manual), and output parameters. After set-up is complete, COMIT selects subtasks to perform based on the specified task-sequencing of subtasks, along with the worker agent's desire to continue or give up. Each subtask selected is performed until the worker agent gives up or completes the subtask (done status).

Each iteration of the action selection and performance section includes the selection of an action, performance of that action, performance of parallel actions or interrupts, and occasionally continuance of the performance of the selected action. The actions, parallel actions, and interrupts selected are those currently available. Their likelihood of selection is based on their probability of occurrence. Because some actions can be interrupted by a parallel action, such actions may continue after the interrupt.

After all subtasks have been performed or the worker agent has given up, statistics are reported to the user. These statistics include the duration and frequency of all actions reported by subtask or aggregated across the entire task. In typical Monte-Carlo fashion statistics aggregated across the simulations can be

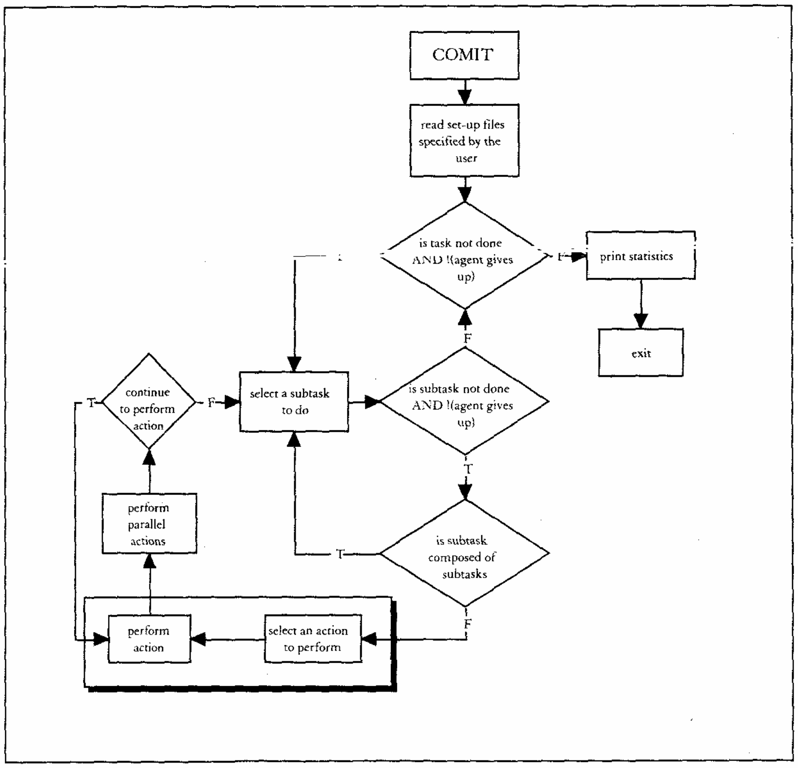

Figure 3. Top-level system diagram of COMIT.

reported for numerous simulations of the same organization. When many simulations are run, reporting is broken out by those simulations that have a "done" status for all subtasks and those simulations containing one or more subtasks with a status of "partially done" or "not done." A subtask—not composed of other subtasks—is "done" if the worker reaches a certain step, such as the eighth of ten steps (note that steps can be skipped along the way). A subtask is "partially done" if the worker reaches an earlier predefined step, such as the forth of ten steps. Otherwise a subtask is "not done." Subtasks composed of other subtasks have their status set to the maximum status of their subtasks.

Parameters

To discuss parameters concisely, we must first describe the topology developed that allows us differing degrees of abstraction from modeled items. Many mod-

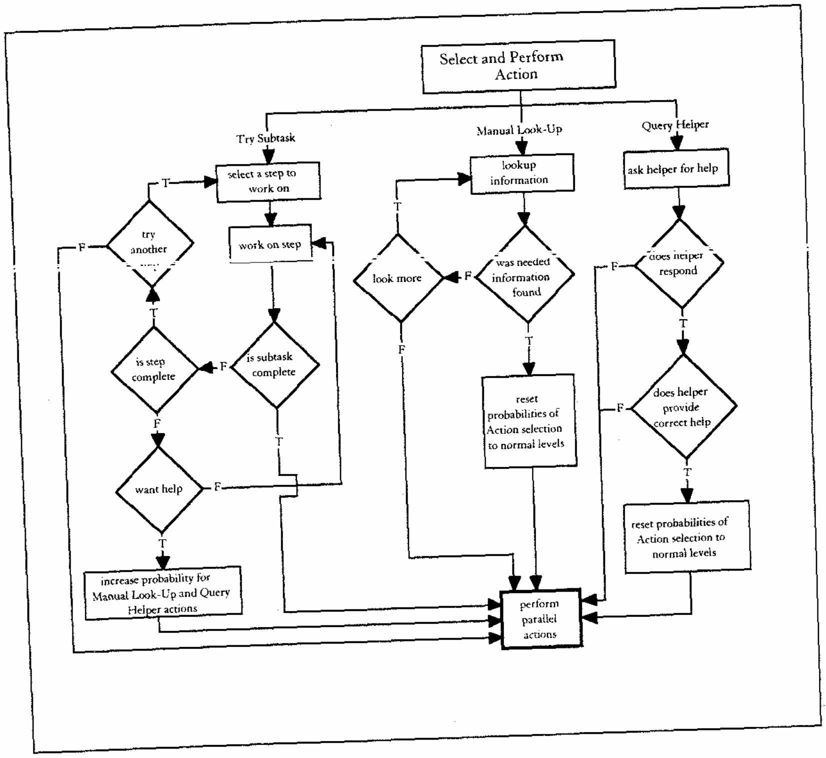

Figure 4. Detailed system diagram of action selection and performance.

eled items, such as a manual, require numerous variables to be adequately modeled. Other items, such as the probability of an action selection by an agent, require simply the specification of a value chosen from a range (for example, the selection of .5 from a range of 0 through 1). This topology was developed to enable users to interact with COMIT without a thorough knowledge of all its intricacies.

Much of what has been modeled requires a group of parameters rather than simply a limited set or range of values. For example, manuals are characterized by two metaparameters—manual type and manual style. For most users, simply noting that the manual has type "electronic" and that style is "reference" is sufficient. However, electronic manuals can also be characterized by setting the underlying parameters, such as speed of access or time required to read a page. These more detailed parameters allow researchers to model items at a more complex level, while the metaparameters provide simple access for users. Table 1 presents a subset of the parameters that can be manipulated by the user. Those in bold type are manipulated for the results presented in the Illustrative Results section.

Parameter	*Topological Type*	*Values*	*Default*
TASK			
Task representation	MP	Hartzell Task	Hartzell Task
Subtask state	LS	Done, Partially Done, Not Done	Not Done, each subtask
Skip steps	LS	Yes, No	Yes
AGENTS			
Role	**LS**	**worker, helper**	**worker, helper**
Worker experience	**M**	**0-100%**	**50%, each cell**
Worker probability skip steps	BCR	0..1	0.5
Presence of helper	**LS**	**Yes, No**	**No**
Helper experience	M	0-100%	100%, each cell
ACTIONS			
Duration of worker action	BCR	upper and lower bounds	action dependent
Duration of helper action	BCR	upper and lower bounds	action dependent
Probability selected by worker	BCR	0..1	action dependent
Probability selected by helper	BCR	0..1	action dependent
Assign actions to subtasks	M	True, False	All cells True
TECHNOLOGY			
Manual style	MP	reference, step-by-step	reference
Manual type	MP	physical, electronic	physical
Audio	LS	not present, present	not present
Video	LS	not present, present	not present
Orientation	LS	remote, face-to-face	remote
OUTPUT			
Number complete runs	BCR	0..1000	200
Number incomplete runs	BCR	0..1000	200
Report action frequencies	MP	Yes, No	Yes
Report task duration	MP	Yes, No	Yes
Report task quality	MP	Yes, No	Yes

Table 1. Illustrative parameters.

The topology consists of four types of parameters: bounded continuous range, limited set, matrix, and aggregate representation. Bounded continuous range (BCR) consist of a range of values bounded by an upper and a lower bound. For example, the probability of an action's selection by the helper may range from 0 to 1. Limited set (LS) consists of a set of values. For example, the status of a subtask is represented as one of three values: "done," "partially done," or "not done." Matrix (M) is a set of values in a two or more dimensional matrix. For example, the assignment of allowable actions for each subtask is represented as a two-dimensional matrix with true and false values in each cell. Each action specified by the user must be assigned to subtasks. While the default allows for all actions to be performed during any subtask, it is not always required. For example, it may be necessary for the worker to report only during a particular subtask. To represent this, a matrix of subtasks by actions is necessary. Finally, metaparameter (MP) is a parameter composed of multiple other parameters, each of which may vary in type.

Task

Users can manipulate three basic parameters in COMIT. The first, task representation, allows them to select a prespecified task. The second, subtask state, allows users to specify the beginning state of each subtask (i.e., whether it is "done," "partially done," or "not done"). For example, work in progress can be represented by specifying the beginning state of a subtask. Thus, a user could explore potential bottlenecks in the task. Finally, skip steps allows users to specify if the artificial agent can skip steps.

Only one task is currently prespecified within COMIT—the Hartzell task (figure 5). The Hartzell task is typical of a relatively simple sequential task that requires specialized knowledge to perform it well. This task involves the maintenance of a Hartzell constant speed propeller assembly. It consists of thirteen sequential subtasks. The measurement subtask (which is currently experimental work) is composed of a set of nonsequential subtasks (four measurements to be taken). In the future, we intend to calibrate COMIT by comparing its output with that from these experiments. A preliminary report on our experimental results appears in Siegel et al. (1995).

If the default task is not desired, users can specify an alternate task by listing all subtasks, their substructures, and the sequentiality among subtasks. COMIT allows any hierarchically decomposable task to be modeled as a set of subtasks.[3] Each subtask may have any number of subtasks. Each subtask, in turn, may also have subtasks. A subtask that does not contain subtasks will have, instead, a series of steps that the agent must take to complete the task.[4] Some subtasks are conditional on others. The sequencing of subtasks is specified by setting a matrix of linkages among subtasks.

Regardless of the task, each subtask and step can have, from the agent's perspective, one of three completion states: "done," "partially done," and "not done." By default, all subtasks and steps are initially in the "not done" state. As the worker agent performs the steps and subtasks, the completion state changes. To advance from one step or subtask to the next, the worker agent must achieve at least the prespecified required level of completion for this step or subtask. This level of required completion is specified by the researcher and may be "done," "partially done," or "not done." For example, in the Hartzell task, the worker agent cannot advance from Disassembly I to Disassembly II without first completing Disassembly I (status "done").

The worker agent can skip steps within a subtask, although quality will be diminished. Nevertheless, the subtask can still achieve a status of "done." Any number of steps within a subtask can also be skipped. Such steps will be left with a status of "not done." Similarly, subtasks can be skipped, but only according to the prespecified subtask ordering. If subtasks are skipped, the effect on

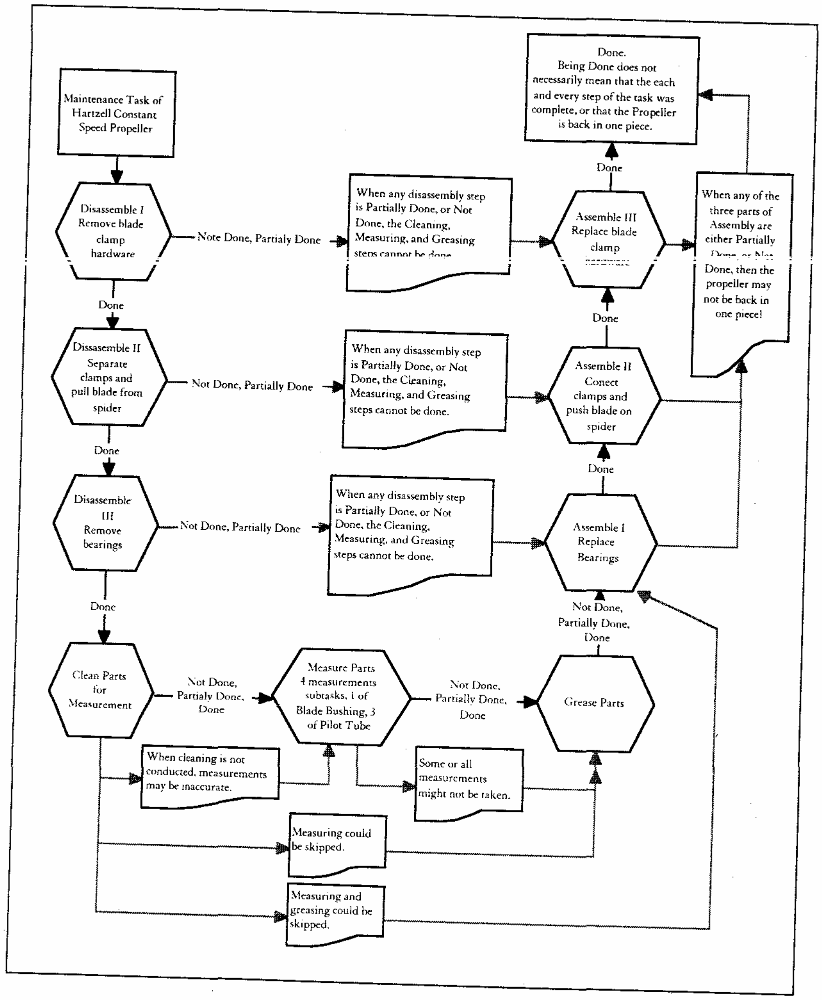

Figure 5. Hartzell task.

task completion will differ depending on the subtask that has been skipped. For example, in the Hartzell task, if the worker agent has "not done" or "partially done" Disassembly I, and decides not to try the subtask any longer, the only subtask that can be performed next is Assembly III. However, the critical measurement step cannot be performed because earlier subtasks were not completed. On the other hand, if the worker had decided to stop working on the clean subtask, leaving that subtask with a "not done" or "partially done," status the measurement subtask could still be performed.

Agents

Agents are characterized by their role (worker, helper), experience (task knowledge), and (if it is a worker) the likelihood that they will skip steps. COMIT can simulate organizations with a maximum of two agents, a worker and a helper. When both a worker and a helper are present, they work cooperatively to complete the specified task using various information technologies. Every simulated organization must contain a worker because only the worker agent performs tasks. The helper is optional. If a helper is present, the information technology chosen determines whether the helper is physically copresent (face-to-face condition) with the worker, or remote (communicating by means of a technology). Both workers and helpers vary in how much experience they have. Experience parameters apply to all subtasks and steps of the task, and have an initial experience level. The worker can learn from experience (by performing the task), by looking up information, or from helper communications. Learning raises the worker's experience level. The helper may initiate a communication with the worker, which may cause the worker to learn. Both workers and helpers can learn, although there is a cap at 100 percent experience. In addition, workers may vary in their likelihood to skip steps or subtasks. The worker probability skip steps parameter allows the user to specify the extent to which the worker is allowed to skip steps. The parameter of presence or absence of helper simply allows the user to specify whether or not the helper is present.

Learning is accomplished by either reading a manual or performing the work. If the agent looks up information and finds it, then it learns that information; otherwise it does not learn (i.e., it does not learn information it finds that is not relevant to the information it seeks). Agents learn about steps, but not subtasks. There is one piece of information for each step. Both workers and helpers are capable of learning by looking up information.

When a worker agent physically tries a subtask, it may learn, despite giving up. Thus, if the agent completes a subtask (status of "done"), then it gains complete experience. If the agent only partially completes a subtask, then it gains only partial experience. If the agent does not complete the subtask, then it learns nothing. However, if a worker has learned more (through lookups or helper agent provided information) than what it might get through experiential learning, then it retains its old experience (i.e., it never forgets, and knowledge is not additive; it is absolute each time). This learning mechanism only applies to the worker.

When a helper is present, the worker can learn by being told, as when the helper provides either solicited or unsolicited help. When help is provided, the worker learns an amount equal to the helper's knowledge on the step in question. Thus, a worker will learn under three different conditions. First, it will learn if at least one step was physically tried and a "partially done" status was achieved. In this case, only the worker learns. Second, it will learn when at least one lookup was made successfully. In this case, both worker and helper gain complete experience. Finally, the worker learns when the helper provides help to the worker at least once. In this case, only the worker learns.

If none of these conditions are met, no learning occurs. Thus, learning (or the lack thereof) does not necessarily rely on whether or not the worker gives up on the task.

Actions

Agents have a discrete set of actions they may perform, as described in table 2. Each action is listed along with the type of actor that can take the action. Each action has an associated probability of selection by the agent selecting the action. Many of these probabilities can change with the task state and/or the information technology. For example, if a worker realizes it needs information in order to complete a step, the probability of selecting actions such as look up or query helper will increase. This feature makes it possible for the researcher to implement assumptions such as individuals prefer to learn by being told rather than through trial and error. Increasing the probability of looking up information does not guarantee that the worker will look up information, nor that the agent will find the requisite information if it does look it up. Additionally, each action has some duration which is uniformly distributed over a range as specified by the user and which depends on the type of agent that can take that action. Finally, the user can specify during which subtasks/steps an action can be selected.

The worker can choose from an array of actions, provide unrequested updates to the helper, learn about the current subtask or step, give up on a subtask or step, or retry it upon failure. In addition, the worker knows the exact ordering of the subtasks and can recognize whether or not information is relevant to the current subtask or step, whereas the helper can respond to requests, view video, request situation updates, provide unsolicited help, ignore the worker, provide incorrect help if the worker and helper are not synchronized to the same step, and so forth. Both workers and helpers know the correct ordering of the subtasks to be performed, but each may have different experience for each step of a subtask.

There are both physical and cognitive actions. Cognitive actions occur in parallel with other actions and take no additional time. Physical parallel actions do take time. The minimum and maximum amount of time taken by each action is defined by the user and may be affected by the choice of information technology and agent experience.

Actions may not have the desired consequences. For example, the worker might ask for help but might not receive it or might receive inaccurate information. Further, the likelihood of some actions can change as a function of other actions. For example, although the worker can provide a situation description to the helper, the worker's desire to do so can change over time with respect to the last time a situation description was provided or as the helper reminds the worker to provide situation descriptions.

Action	*Definition*	*Actor*
Try Subtask	agent physically attempting a step	worker
Query Helper	agent requests step-specific information	worker
Manual Lookup	agent looks up information in a manual for a particular step	worker and helper
Report	agent makes a report	worker
Nontask Communication	communication between worker and helper not related to the task being performed	worker and helper
Waste Time	time spent not performing the task and not communicating	worker and helper
Parallel Cognitive Actions		
Update Desire to Provide Situation Description	adjusts agent's desire to provide a description of the step the agent is doing	worker
Update Desire to Provide Unsolicited Help	adjusts agent's desire to provide unsolicited help	helper
Update Experience	adjusts agent's desire to provide a description of which step the agent is doing	worker and helper
Parallel Physical Actions		
View Camera	agent can view the monitor and learn which step other agent is currently on	helper
Provide Situation Description	agent provides a description of which step the agent is working on	worker
Unsolicited Help	provide unsolicited help to the other agent	helper

Table 2. Available actions.

Technology

Information technology is characterized in terms of the way in which stored information is presented (manual style and manual type) and the way in which information is transmitted (audio, video, or orientation). With the parameters manual style and manual technology, a range of interesting mechanisms for retrieving information can be simulated. For example, a standard paper manual is physical in type and reference in style, whereas the same information presented in an on-line system would be electronic in type and possibly step-by-step in style. If an electronic manual is chosen, the user may allow the worker to use a check-list that automatically advances the manual's page to the appropriate next step. With the parameters (constraints) audio, video, and orientation, it is possible to characterize many different communication technologies. For example, discussions with a remote expert over the telephone has an audio constraint, no video, and a remote orientation.

Under a face-to-face orientation, the worker and helper agents are collocated. When worker and helper are not collocated, an audio channel exists between them. The audio channel in both the remote condition and the face-to-face condition is similarly modeled. The audio channel allows either agent to initiate a communication with the other agent and to respond to a communication if desired. No interrupts are possible, unlike a real telephone; however there is no coordination time required between communication turns, unlike a walkietalkie. Video allows the helper to view the worker's workspace from the perspective of the worker. This can be accomplished, for example, by using a lipstick camera mounted on a visor or helmet of a worker. The quality is akin to that of NTSC video (30 frames per second), although the helper might not always choose to view the video signal.

Workers can look up information in both physical and electronic manuals, which differ in the time it takes to access information (search time).[5] Furthermore, information in the manual can be arranged either step-by-step or in a more standard reference format. This choice also affects the search time. A paper reference manual can be described as a random-access manual where access is accomplished through the use of an index. Thus, search time for such a manual should be fairly constant and the probability of finding the information with the use of an index should be uniform across the task. In contrast, a step-by-step manual is sequential. Thus, search time will vary depending on which section of the manual was last used relative to which section is needed. Thus, when using a step-by-step manual, an agent will have a high probability of finding information on the next step and a lower probability of finding information on preceding steps or subsequent steps.

Output

COMIT will generate a series of results for the specified organization. Since the program is not deterministic, a Monte Carlo approach should be taken. To specify the number of runs, the user can specify either the maximum number of simulations or the maximum number of simulations in which the worker completes the task with a "done" status. Depending on the task and the agents, there may be a large difference between these two. For example, in the Hartzell task it might take 1,100 to 1,700 runs (depending on the experience of the worker) to generate 200 runs where the worker actually completes the task.[6] This condition reflects the agent's relatively high likelihood of giving up on the task rather than finishing it. In reality, there may be many incomplete simulations performed before 200 complete simulations are found. COMIT continues virtual experiments until the requested number of simulations are found, or until the computer runs out of memory.

To a certain extent users can tailor the output. That is, users can specify whether they would like to have reported the frequencies of the various actions, the length of time it took to do the task or the various subtasks, or the quality with which the task is done.

Example

Imagine that you are trying to use COMIT to design a virtual experiment. You are interested in how long it will take an agent to perform the Hartzell task. The scenario you have in mind involves a fairly skilled worker performing the

NOT COMPLETED SIMULATIONS										
Simulations			Agents	Output Measures						
Requested Not Complete	Actual Not Complete	Total Run	Experience Worker	Time Taken	Status	Quality	Manual Look ups	Physical Task Tries	Reports	Skipped Steps
200	1065	1265	30%	4260.88	2.02	0.29	53.39	133.58	5.75	56.93
200	1174	1374	40%	4585.45	2.01	0.29	55.88	131.68	5.76	56.40
COMPLETED SIMULATIONS										
Simulations			Agents	Output Measures						
Requested Complete	Actual Complete	Total Run	Experience Worker	Time Taken	Status	Quality	Manual Look ups	Physical Task Tries	Reports	Skipped Steps
200	200	1265	30%	5771.91	1.00	0.55	89.64	252.34	13.03	21.05
200	200	1374	40%	5665.55	1.00	0.55	89.29	245.27	12.71	20.99

Table 3. Sample output.

diagnosis and repair task with only the aid of a reference-style manual and his or her own experience.

First you select a task—the Hartzell task. This selection sets the task representation parameter. Then you decide that the defaults for the next two task parameters, subtask state and skip steps, are appropriate for this experiment.

Next, you specify the information about agents. You decide not to have a helper; this decision requires no parameter changes, since helper present is "no" by default. The only agent parameter requiring change is worker experience. You set the worker's experience to 30 percent to reflect an "unskilled worker."

Next you adjust the action parameters. You choose not to change the assignment of actions to subtasks, but you do want to change two of the action selection probabilities (probability of selection by worker). For example, you might want to model the worker as someone who dislikes looking up information in a manual, but enjoys filing reports frequently. Thus, you reduce the probability of looking up information and increase the probability of filing reports from the respective default values.

The technology parameter defaults are all satisfactory to you. The worker has access to a physical reference manual. Without an electronic manual, the agent cannot have a checklist because that is electronic based. The remaining technology parameters do not apply to the solo condition.

Finally, you set the output parameters. As in any experiment, you must specify a sample size. In COMIT, you specify the number of complete and incomplete simulations to run. Because you are working in an exploratory mode, you request timing, quality, and frequencies of all actions. You decide on a sample size of 200 complete simulations. You are not interested in frequencies or quality; therefore, you set the frequencies of actions and report task quality to "no." The Duration's default is "yes," so you do not change this parameter.

Having set these parameters, you are now ready to run the virtual experiment. COMIT runs in batch mode and upon completion of the experiment reports both

	Low Technology			High Technology		
	Parameter Type	*Parameter*	*Value(s)*	*Parameter Type*	*Parameter*	*Value(s)*
Solo	AGENT	Helper Present	No	AGENT	Helper Present	No
	AGENT	Worker Experience	30-70%	AGENT	Worker Experience	30-70%
	AGENT	Helper Experience	N/A	AGENT	Helper Experience	N/A
	TECHNOLOGY	Manual Style	Reference	TECHNOLOGY	Manual Style	Step-By-Step
	TECHNOLOGY	Manual Type	Physical	TECHNOLOGY	Manual Type	Electronic
	TECHNOLOGY	Check-List	N/A	TECHNOLOGY	Check-List	Yes
	TECHNOLOGY	Audio	N/A	TECHNOLOGY	Audio	N/A
	TECHNOLOGY	Video	N/A	TECHNOLOGY	Video	N/A
Collaborative	AGENT	Worker Experience	30-70%	AGENT	Worker Experience	30-70%
	AGENT	Helper Present	Yes	AGENT	Helper Present	Yes
	AGENT	Helper Experience	100%	AGENT	Helper Experience	100%
	TECHNOLOGY	Manual Style	Reference	TECHNOLOGY	Manual Style	Step-By-Step
	TECHNOLOGY	Manual Type	Physical	TECHNOLOGY	Manual Type	Electronic
	TECHNOLOGY	Check-List	N/A	TECHNOLOGY	Check-List	Yes
	TECHNOLOGY	Audio	Present	TECHNOLOGY	Audio	Present
	TECHNOLOGY	Video	Not Present	TECHNOLOGY	Video	Present

Table 4. Experimental design.

the number of complete simulations and the average duration of all complete simulations. The output from this experiment is shown in table 3, along with the results of an identical experiment with worker experience of 40 percent.

Illustrative Results

Let us return to our hypothetical manager who is trying to make technology purchasing or scheduling decisions where a wide variety of technology and work arrangements are possible. By modeling and simulating the situation, the manager can make educated decisions about this rather complex scenario. To examine this scenario, we used COMIT to run virtual experiments with four conditions (see table 4). We examine extreme cases, from an environment that exists now—individuals working alone with hard copy manuals—to a futuristic view employing wearable technology—a worker and helper with lots of technology (audio, video, and an electronic manual). The experimental design is illustrated in table 4.

	Technology	
	Low	*High*
Solo	5608.6	5513.5
Collaborative	13266.5	11693.8

Table 5. Task completion time.

	Technology	
	Low	*High*
Solo	241.1	238.9
Collaborative	251.1	239.6

Table 6. Number of physical manipulations.

Let's imagine that the manager argues that some training has been provided and that all workers have some experience (say 50 percent) and all helpers are experts with complete experience (100 percent). If the manager was interested in making a decision based solely on reducing time, then he or she would select an individual worker with little technology (table 5). However, if the manager was more concerned with minimizing the physical manipulations of the machinery, then he or she would elect a collaborative condition loaded with technology (see table 6). Note that it appears that collaborative groups are more affected by technology than individual workers. Further analysis in which the degree of technology is varied in the collaborative setting might provide us with the optimal amount of technology for this particular task.

Now imagine that the manager has decided not to train the workers, a priori. Instead, the manager has a variety of employees to assign as workers to the task, but they vary in their experience. The manager still has highly skilled employees (100 percent experience) available to act as helpers. Now the decision space becomes a bit more complicated. Previously, we considered the case where the workers were all equally and partially skilled (50 percent experience). Now let us consider five levels of worker experience—30 percent, 40 percent, 50 percent, 60 percent, and 70 percent. As can be seen in tables 7 and 8, the relation between technology and the individual versus the collaborative condition depends on the experience of the worker.

Basing a scheduling decision on completion time had the manager schedule an individual worker with 50 percent experience in a low technology arrangement. The manager would make the same judgment for all workers with less

		Technology	
	Worker Experience	*Low*	*High*
Solo	30%	5771.9	5785.4
	40%	5665.5	5611.6
	50%	5608.6	5513.5
	60%	5393.2	5346.3
	70%	5233.2	5178.8
Collaborative	30%	13903.2	12040.4
	40%	13502.1	11847.1
	50%	13266.5	11693.8
	60%	12539.0	11296.2
	70%	11980.5	10867.3

Table 7. Task completion time as worker experience varies.

		Technology	
	Worker Experience	*Low*	*High*
Solo	30%	252.3	253.0
	40%	245.3	245.5
	50%	241.1	238.9
	60%	230.3	230.7
	70%	222.7	221.5
Collaborative	30%	267.2	249.0
	40%	256.9	244.1
	50%	251.1	239.6
	60%	237.7	228.9
	70%	227.9	220.6

Table 8. Number of physical manipulations as worker experience varies.

than 70 percent experience. Highly experienced workers complete the task more quickly with higher technology.

In the case of physical manipulations of the machinery, the manager is faced with an even more complex decision. If workers are working individually, technology has little impact on performance. However, under collaborative conditions, technology actually decreases the number of manipulations, even below what it would be if there were no helper.

This virtual experiment suggests some of the complexities in making decisions. Simple rules of thumb may not capture the dependencies between technologies, work conditions, and experience. By using COMIT, the manager can adjust parameters, do a series of what-if analyses, and locate more desirable conditions.

Discussion

With today's telecommuting technology and tomorrow's wearable technology, managers must begin to learn to evaluate the worth of these technologies in a variety of situations. COMIT is a prototype of the type of computational framework that is needed to examine the impact of technology, training, task, and organizational design on performance, given adaptive agents. The ultimate goal is to develop a tool like COMIT that could be used by researchers and managers as a decision aid to understand the effects such technologies and their application environments have on both cooperative and individual work in organizations.

COMIT allows users to model problems, such as diagnosis and repair, as a series of interlocked stages. In principle, it is therefore possible to use COMIT to examine either a user's specific needs or more generic tasks. Similarly, it is also possible to model a variety of technologies. These technologies include those that are widely available today (such as the telephone) as well as those that may be widely available in the near future (such as wearable computing). Further, COMIT has the flexibility to allow a user to describe both general classes of technologies (such as manuals) and specific subclasses (such as physical reference manuals or electronic step-by-step manuals). Finally, COMIT is flexible enough to allow users to model agents with varying capabilities. By varying an agent's task experience or willingness to provide feedback, users may be able to better understand the relationship between agents and differing technologies.

COMIT provides users with the ability to model task, technology, and agent with ease while remaining applicable to the user's domain. What COMIT lacks (aside from a user-friendly interface), are facilities for easily modeling new technologies, tasks involving more than two people, and complex organizational designs. The task of modeling new technologies is complex. Although some technologies have models associated with them (Clark and Brennen 1991; McCarthy and Monk 1994), new classes of technologies often emerge before a theoretical model is accepted. Researchers interested in building or maintaining systems such as COMIT should either select models that appear applicable or design new models. In addition, each technology may require the implementation of new source code.

COMIT's organizational component is composed of, at most, two people working in a predefined fashion. Consequently, COMIT is a microlevel simulation not currently capable of simulating the complexities of large organizational projects. While other simulations exist to model the macroworld (see Levitt, et al. 1994), they do not model the microlevel interactions COMIT models between individuals and technology. For COMIT to evolve into a system useful for examining larger projects and more complex organizational designs, care must be taken to ensure that the spirit of COMIT continues; namely that the computational model is flexible enough to apply to a wide variety of situations. In particular, COMIT needs to be expanded to include general features of information flow and the constraints of organizational design on performance.

We applied COMIT to a simple issue—should management invest in more technology for solo or collaborative repair work. Our findings suggest that a more experienced agent performs the task faster with more advanced technology, but hinders the less experienced. This simple experiment demonstrates that technology, training, and organizational design interact in complex ways to influence performance. Virtual experiments such as these are important for developing a theory of embedded information technology. The value of these experiments, however, will increase once models like COMIT are calibrated.

Specifically, the need to model environment, agent, task, and technology required us to make a number of assumptions not explicitly discussed in this chapter. Many of these assumptions, usually simplifying assumptions about human behavior, have their basis in experimental results, while others are based on experience and intuition. Those not yet based on experimental results should be verified to validate COMIT. Validation is a two-step process involving calibration and prediction.

The calibration of a computational model is an important step in making its results useful. Calibration of COMIT will require matching COMIT's predictions with the experimental results from human experiments. For example, we must create an experiment varying the technology and agent parameters. To test the quality or time issue as it relates to computer-based manuals, we can design a human experiment varying the type of manual used. To test telecommunication devices, we can vary the presence or absence of a helper with such devices. We intend to calibrate COMIT using data from a human experiment using the Hartzell task. Calibration of the simulation will be conducted by "training" COMIT using the results of 20 percent of the human subjects. This training will take place through a process of setting parameters in COMIT based on this 20 percent sample, and then predicting performance results for this sample.

Once the performance of the COMIT agents and the humans is matched in this small sample within some tolerance, COMIT should be able to predict the performance of the remaining 80 percent of the subjects. If the prediction matches the experimental results, the model has been validated.

Conclusion

As information technologies come to play an increasingly important role in organizations, the need for tools that allow users to preview the potential impacts of these technologies prior to their being established in the organization will grow. COMIT is a prototype system that illustrates some of the features of such a system. It is useful for building theory about the interaction between technology, task, and organization when agents are adaptive.

Acknowledgments

This work was supported in part by the ARPA SSTO HCI program under grant DACA88-94-C-0014 and the Human Computer Interaction Institute at Carnegie Mellon University.

Notes

1. We use the term information technology in its broadest sense to include tele and traditional communication technologies, as well as electronic and physical information systems.

2. Additionally, a trace can be printed as the simulation proceeds.

3. The number of subtasks is bounded only by the amount of memory on the researcher's computer.

4. All subtasks have ten steps.

5. Additional parameters can also be specified that determine whether the manual is random access or sequential access, the time to find information dependent on the current open manual page, and the probability of finding the information needed. This enables the creation of forms other than those described here.

6. This is based on an inexperienced worker working alone, and using only a hard copy manual.