THE INTERNATIONAL JOURNAL OF AVIATION PSYCHOLOGY, *13*(1), 1–22

FORMAL PAPERS

The Cockpit as Multiple Activity System: A Computational Model for Understanding Situated Team Performance

Brian Hazlehurst
Kaiser Permanente's Center for Health Research
Portland, Oregon

The complex activity that takes place in airline cockpits is productively understood as cognitive action distributed across agents and technology. This article focuses on how the activities of crew members in a B-727 cockpit collectively constitute a functional unit of cognition—a team of cognitive agents who must be coordinated with each other and their task environment. The article addresses actual crew performance executing the Before Takeoff Checklist and applies a model previously developed for characterizing individual cognitive properties underlying use of multiprocess systems. The analysis reveals mechanisms that are interruption-tolerant and enable robust preflight verification of aircraft configuration.

The cockpits of modern jet airliners are locations of complex cognitive activity. This activity is productively understood as cognitive action distributed across agents and technology, both within and beyond the cockpit, as a system. This article focuses on how the activities of crew members in a cockpit collectively constitute a functional unit of cognition—a team of cognitive agents who must be coordinated with each other and their task environment. The following synopsis of a simulation experiment by Lauber (1984) exemplifies the need for coordination in the cockpit under high workload conditions:

Requests for reprints should be sent to Brian Hazlehurst, Kaiser Permanente Center for Health Research, 3800 N. Interstate Ave., Portland OR 97227-1098. E-mail: brian.hazlehurst@kpchr.org

> After the captain decided that it would be necessary to dump fuel, the captain and first officer decided that 570,000 lbs was the correct landing weight.... The flight engineer then calculated a dump time of 4 minutes and 30 seconds and this was accepted by the captain without comment, although it was approximately one third the actual time required... instead of dumping for 12 minutes, the engineer terminated the dumping procedure after only three minutes.... Unsatisfied, he again started to recalculate, but was interrupted by the failure of the No. 3 hydraulic system (part of the scenario). During the next eight minutes, the flight engineer was subjected to high workload but then noticed that the gross weight was much too high and decided to reconfigure the fuel. During that time, he was subjected to further interruptions and did nothing more about the fuel until the captain noticed that the gross weight indicator read 647,000 lbs and decided to make an over gross weight landing. A minute and a half later, the flight engineer rechecked the fuel as part of the landing checklist and became concerned about the gross weight. He spent a minute and a half rechecking calculations and announced that the aircraft gross weight computer must be in error. Two minutes later, the simulator landed at 172 kts with only 25 degree flaps, 1000 ft/minute sink rate, and 77,000 lbs over correct weight. (pp. 20–21)

The causes of incidents like these are usually attributed to "human error." According to Foushee and Helmreich (1988), "Although estimates vary, even conservative figures attribute about 65% of all accidents worldwide to the human error category" (p. 192). A typical response to data such as those recounted in the simulated flight above is to envisage new technologies that automate tasks and thereby eliminate "human error." In fact, many of the tasks of the flight engineer *have been* relinquished to computer and automated control, often resulting in the elimination of one crew member from the cockpits of modern aircraft. While such automation (targeting tasks that humans are known to perform poorly) is an important component of improved design, we seldom understand the basis for crew performance—before or after the redesign!

The fuel-dumping example cited above entails many crewmember actions that are inappropriate. However, it is difficult to declare the system broken without a thorough characterization of normal performance. This article attempts to describe some of the properties of crew coordination in the cockpit of a three-person Boeing 727. I show that the flight engineer's role, as one member in a multiple-activity system, includes many computational properties that are not a priori obvious, yet which render the cockpit system robust under many conditions. In particular, we examine the nature of interruptions in the execution of the well-rehearsed team task of checklist execution and explore which computational properties are entailed by the flight engineer's performance from the standpoint of addressing the multiprocess nature of cockpit activity. This approach demonstrates that aspects of the flight engineer's activity support the multiple tasks that are simultaneously executed in the cockpit and are not simply those of a crew member dedicated to calculating raw data. This perspective on the flight engineer's role in the cockpit holds important implications for flight safety and cockpit automation design.

MULTIPLE ACTIVITIES IN THE COCKPIT

Each crewmember in the cockpit plays a role in the institutionalized effort of safe and efficient flight. These divisions of cognitive labor have evolved through a complex history of making flight safer, more efficient, more satisfactory to crew members and their unions, more appealing to customers, more profitable for corporations, more in line with scientific knowledge, and more accountable to investigation. It is nearly impossible to make general claims about *why* the divisions in cognitive labor within the cockpit have come to be structured exactly the way they are, although the historical development certainly yields valuable insight into aspects of modern crew culture that influence crew coordination (cf. Foushee, 1984).

What we *can* readily do is look at *how* the divisions of cognitive labor among crew members are instantiated in the *actual performance* of well-defined cockpit tasks, such as executing a checklist during normal operations or executing a problem-solving procedure during abnormal flight conditions. These tasks entail well-rehearsed, bounded, and well-documented protocols for executing multiparty (and therefore public) cognitive tasks. These facts make it possible to characterize the resources that crewmembers may be drawing on to inform their actions. That is, given sequences of states of the cockpit—including crewmembers' situations within it—we can infer the bases for actions that serve as causal connections between these states. This requires paying attention to the traditional concerns of cognitive science, such as individual attention and short-term memory. But our theory must be more general about the nature of cognitive actions and information processing in the cockpit and more willing to look beyond the immediate activity to the broader context that informs it. That is, we need to draw on a wide range of ethnographic data so as to give order to the multitude of resources available to, and employed by, crewmembers for performing in the cockpit.

This more general theory of cognition enables a description of cockpit coordination in terms of many more computational events than just those that take place in the brain of a single pilot. Rather, the theory must accommodate the simultaneous satisfaction of parallel but interacting task performances distributed across actors situated in a specific and culturally defined workplace. This article takes up the challenge of applying this perspective on cognition, called "distributed cognition" (Hazlehurst, 1994; Hutchins, 1996), to address the multiple activity system constituted by group performance in the cockpit of the Boeing 727 aircraft.

The 727 is flown by a three-person crew composed of the following members: a captain (Capt.), a first officer (F/O), and a second officer (S/O). The Capt. and F/O face forward with nearly identical instruments and controls in front of them. The engine throttles, flaps handle, and some other gear (collectively known as the "throttle quadrant") are within easy reach between the two pilots, allowing either to be in control of flying the aircraft. The Capt. and F/O rotate the duty of flying between them (here denoted as "pilot flying" [PF] and "pilot not flying" [PNF]), meaning that, although the Capt. retains the formal status of "pilot in control" at all times, he or she is responsible for the F/O's performance and therefore must attend to the F/O's actions, even when the latter is acting as PF. The S/O, sometimes referred to as the flight engineer, is seated directly behind the F/O, facing the right side of the aircraft, where a large panel of gauges and switches is located. This panel constitutes the aircraft's non-flight-critical control center. It includes such systems as fuel, electrical power, and air pressure. The S/O's main duties are related to administration and monitoring of these systems.

This article addresses the idea that while crewmembers have highly specialized roles constituting somewhat independent activities, their actions are highly interdependent and require and entail coordination to achieve adequate group performance. Some of this interaction is explicitly institutionalized in protocols designed to promote coordination among the crew, the aircraft, air traffic control (ATC), and the flight situation. For example, it is customary for a permission granted by ATC to be repeated by the communicating crewmember over the radio. This feature of the communication protocol entails an important set of resources for both the individual receiving the message (e.g., a chance for echoic memory to penetrate deeper into consciousness by enacting motor sequences which duplicate the message) and the cockpit system as a whole (e.g., multiple instances of the message focuses attention upon its meaning), ensuring shared understandings about the current situation.

But some implicit mechanisms for assisting crew coordination never make it into the manuals (e.g., knowledge shared among pilots of a company—or of a type of aircraft—that aids performance). Other behavioral patterns that facilitate coordinated crew performance never make it into the oral tradition of the culture (e.g., when observed by other crewmembers, these behaviors can yield expectations about subsequent events; cf. Hutchins & Palin, 1997; Segal, 1990). Seldom do any of these kinds of mechanisms make it into the specifications from which new technologies are built. And yet, these mechanisms provide resources for organizing behavior and therefore play an important role in the achievement of adequate performance.

This article presents a model of crew interaction as a multiple activity system in order to try to characterize the elusive properties of group performance. The general notion is that crew members have independent tasks to perform, yet accomplish those tasks with more and less degrees of sharing of knowledge, attention, and actions about those tasks. By examining the resources available to, and produced by, crewmembers for accomplishing these tasks, the mechanisms that accommodate group performance in the cockpit are made explicit and become a basis for evaluating proposed technological and social changes to the system. Such an approach creates an opportunity to guide the evolution of the system through design that enhances group and thus cockpit system performance.

Multiple Activities in Normal Flight: Executing a Checklist

Although the PF is in charge of flying the aircraft, the PNF is customarily responsible for flight planning and communications with ATC and the cabin crew, as well as two-man troubleshooting with the S/O, should the need arise. The PNF is also responsible for traffic lookout at all times, meaning his or her attention to the visual field outside the aircraft is required whenever possible. The S/O's tasks are largely related to acting as an interface between the aircraft and the two pilots, on the one hand, and between the aircraft and the company on the other. For example, in preparation for takeoff, the S/O computes engine thrust settings given current load, weather, and runway altitude variables. Later in the flight, the S/O will communicate with the company about aircraft maintenance, fuel accounting, and arrival schedules.

As part of the normal functioning of everyday work routines, the multiple activity system of normal flight is well-rehearsed. Crewmembers have well-developed expectations about states of the entire cockpit, including what the other crewmembers are doing and plan to do next. Often these routines are reinforced by institutional means that make available prescribed protocols for crewmembers' actions.

Checklists provide an important example of the institutionalization of crewmembers' interactions. A checklist is a sequence of prescribed states of the aircraft that must be verified by the crew.[1] Checklists are employed during critical situations—generally before takeoff and landing phases of flight—where an improper airplane configuration carries a high risk and high crew workloads increase the likelihood of such an event. For example, prior to takeoff the setting of flaps (giving desired lift), engine pressure ratios (giving desired thrust), and speed bugs (marking ground speeds on the takeoff roll) must be verified. To accomplish this task, a paper card listing the sequence of checks to make is part of each crewmember's paperwork.[2] A different card, with different action items on each, is used for each checklist, and each takes its name from the relevant stage

[1]Here I consider only the "verification" form of checklist use. Degani and Weiner (1990) identify two other forms of checklist use, the "do-list" and "combination" methods. The verification method accounted for 60% of checklist use among a sample of 20 airlines taken by Boeing in 1989 (Degani & Weiner, 1990, p. 19).

[2]Mechanical and electronic checklists also exist, but here we consider only the manual/paper checklist of the B-727.

of flight: Before Start, Before Takeoff, After Takeoff, Approach Descent, Final Descent, and Parking.[3]

Furthermore, norms exist (company policies set in accordance with FAA regulations[4]) for initiating and executing checklists. The Capt. or PF is generally responsible for initiating all checklists by calling for their execution. Each crewmember is then expected to pull out and attend to his or her own copy of the checklist. Checklists are executed using the "challenge–response format." The S/O (challenger) reads an item on the list and the appropriate responder(s) (Capt., F/O, or both) is/are supposed to check that the state(s) of the system referred to by the label is/are correct. The S/O then proceeds, upon hearing the response and verifying the state for him- or herself, to the next item on the list. The procedure ends with the explicit proclamation that the checklist has been completed.

Note that the state referred to by each item on the checklist should have already been set, a product of individual crewmembers having already executed their own independent procedures. The check thus acts, by way of collective ritual, to ensure the proper functioning of (previously enacted) individual performance, as well as to establish a situation where sharing the aircraft's state is possible. Degani and Weiner (1990, p. 18) claim that achieving this *parallel redundancy* in the system is the principle epistemic factor behind the philosophy of checklist use. The first kind of redundancy, *configuration redundancy*, stems from the "verification" nature of the checklist—it is a backup for another procedure, the initial setting of aircraft state. The second kind of redundancy, *mutual redundancy*, stems from the collective nature of the execution—two or three crewmembers are engaged in making the check.

Yet, despite Degani and Weiner's astute attention to the task and cultural world in which checklists are executed, the authors fail to address the actual properties of group performance. What do knowledge sharing, "keeping all crewmembers in the loop," facilitating "crew coordination," or "logical distribution of cockpit workload" actually mean for performance in the cockpit? How would one know if a new piece of technology enhanced or hindered these properties of performance? It would be desirable to know enough about the constraints on group performance to understand how the cockpit functions as a system.

The analysis in this article attempts to reveal some of the mechanisms by which the crew achieves coordination during collective work activities. The analysis suggests which roles might be played in the cockpit system by such concepts as shared knowledge, divisions of responsibilities, joint and individual attention to tasks, and

---

[3]Here, as throughout this article, we are dealing with one airline's procedures and policies. As Degani and Weiner's (1990) study shows, much variation exists across airlines in the actual policies, implementation, and use of checklists.

[4]As Degani and Weiner (1990, p. 23) note, FAA regulation of checklist policy and use is almost nonexistent. In effect, the FAA representative (Principal Operations Inspector, or POI) assigned to each airline has a lot of freedom to dictate (or negotiate) the policy use and enforcement.

the institutionalization of crew interactions. In this article, a computational model of group performance in multiple activities organizes these concepts. The analysis relies on examining detailed video footage of crew performance during the first segment of a flight undertaken by a practicing airline crew aboard the NASA–Ames B-727 flight simulator. The simulated flight departs from runway 16 Left at Sacramento (SMF) en route to Los Angeles (LAX).[5]

Before Takeoff Checklist

In this episode, a little longer than 2 min in duration, the crew must execute Before Takeoff procedures, taxi the aircraft into takeoff position at the end of the runway, communicate with Sacramento ground control (SMF-GND), and execute the Before Takeoff Checklist.[6] The transcript for the episode is shown in Table 1.[7] The episode begins with the aircraft taxiing out to the runway. SMF-GND has instructed the crew to follow the center taxiway at a heading of 315 to find runway "one six right." The Capt. is steering the plane using the tiller, which controls the front wheel of the aircraft. Only the Capt. has access to this device, therefore only the Capt. can steer the aircraft while taxiing.

Checklist initiation. At 4356, the F/O begins his private procedures to configure the aircraft for takeoff. The Capt. also begins this activity, apparently prompted by the F/O's actions. The S/O monitors this scene and (at 4414) prepares for his role as challenger on the Before Takeoff Checklist by placing his finger on the first item of the checklist card. Subsequently, 6 sec later (at 4420), the S/O briefly turns his attention to some other writing task, possibly recomputing trim settings (see 4523) and returning his attention to the checklist when the F/O speaks at 4421. Hearing the F/O and Capt. agree that they are "ready," the S/O then initiates the Before Takeoff checklist *with no explicit reference being made to the checklist.* The crew's behavior here indicates that the task of initiating the checklist is anticipated by all crew members, suggesting that this task is embedded within a larger one of "takeoff preparations."

[5]The segment is part of a complete simulated flight that lands on runway 24 Right at Los Angeles International Airport (LAX). On the approach into LAX, the number-one generator fails and the crew must reconfigure the electrical system to draw power from the remaining two generators, leaving the failing generator as a source of backup electrical power only. A few minutes later, the number-three engine reveals low oil pressure and is shut down, leaving two engines and one healthy generator for the landing. A longer version of this article addresses the crew's use of published Irregular Procedures to problem-solve during the landing segment of this flight.

[6]Space limitations prevent inclusion of an image of the actual checklist.

[7]The four-digit numbers given in the text (e.g., 4356) refer to the video time code of the cited event, found in the left-most column of the transcript.

TABLE 1
Execution of Before Takeoff Checklist

Time	*Actor*	*Verbal Transcript*	*Other Behavior Transcript*	*Interpretations*
4320	F/O	Well let's see I think the taxiway is off to the right		
4326	Capt.	I think it is too		
4328	F/O	Well		
4341	F/O	I think it's over there Come to		Capt. is steering plane with tiler down by left knee
4343		Three-one-five		This is the heading of the taxiway given to them by Ground Control at 4210
4345	Capt.	Yeah		
4348	F/O	Ok there's the end of the runway, there		
4350		Ok		This acknowledgment frees the F/O to begin his own procedures to configure the A/C
4356	F/O		Sets flaps to 15°	The takeoff data card, computed by S/O and handed to F/O earlier contains the desired takeoff flap setting
4357	Capt.			Appears to be cued by F/O setting flaps
4358	F/O		Leans over to read stab trim	
4358	Capt.		Touches throttle levers	
4359	F/O		Begin control check	
4401	Capt.		Begins check and test of main gear antiskid	
4406	Capt.		End check of main gear antiskid	
4407	F/O		End control check	
4414	S/O		Puts his finger on the first item in the checklist	
4420	S/O		Reads gauges (fuel?) removes pen from pocket begins some other computation	
4421	F/O	I'm ready whenever you are		F/O indicates he has completed his procedures

(*continued*)

TABLE 1 (*Continued*)

Time	*Actor*	*Verbal Transcript*	*Other Behavior Transcript*	*Interpretations*
				and is ready to run the Before Takeoff Checklist
	Capt.	Ok let's		
	S/O	\Start levers		S/O has anticipated the call for the checklist at least as far back as 4414. His initiation of the checklist does not entail any explicit reference to the checklist
4424	F/O	Three at idle	Touches the start levers	
	S/O	Flaps		
4426	F/O		Touches the takeoff data card	
4427	F/O		Points to flap position indicator	
4427	S/O		Looks away from checklist	
4428	F/O	The card says fifteen We have fifteen fifteen		
4429		Green light detent	Taps the flap handle	F/O anticipates the next item on the checklist
4430	F/O		Reaches speed brake handle and ensures it is down	
4431	S/O	Ok		
4438	Capt.		Leans toward F/O and raises right index finger as if to point out the window	Capt. is looking uncertain about his visual bearings on the runway location
4439	S/O	Speed brake		
4440	F/O	It's down		
4441	Capt.	See		
4442	Capt.	Is this the taxiway here		Capt. interrupts checklist execution
4442	S/O	Compass indicators		
	F/O	xxx Keep going down the line I think we'll see it		F/O responds to Capts. call for assistance to visually locate the runway
4447	F/O	Aligned on the left (right?)		F/O attempts to resume checklist execution by prompting Capt. to verify his own instrument setting
4447	SMFGND	Nasa nine hundred /…/ Come to the left		

(*continued*)

TABLE 1 (*Continued*)

Time	*Actor*	*Verbal Transcript*	*Other Behavior Transcript*	*Interpretations*
		about fifteen degrees for the center taxiway		
4452	F/O	Nasa nine hundred		
4453	Capt.	That's it		
4455	S/O	Compass		S/O attempts to resume checklist execution
4456	Capt.	\xxx Yah, there we got it		Capt. signals end of his problem
	S/O	Compass indicators		S/O resumes checklist execution
4457	F/O	Aligned on the right		
	Capt.	And the left		
4459	S/O	Flight and nav instruments		
	F/O	Checked and set		
	Capt.	Ok /…/ I'm checked and set on the left		
4506	S/O	Engine anti-ice		
	F/O	Close	Reaches engine anti-ice switch	
4510	S/O	\Pitot heat		
	F/O	It's on	Reaches pitot heat switch and changes its position	
4512	S/O	Antiskid		
	F/O	It's on	Reaches antiskid switch	
4514	S/O	Controls		
	F/O	Checked lights are out		
	Capt.	xxx On the bottom		
	F/O	xxx		
4516	S/O	\Hydraulics checked		
		Brake interconnect is closed[?]		
		Fuel is /…/ set		
		Apu is off		
4523		Manifest changes		
		xxx Make the trim five-point-five		
	F/O	Ok		
	S/O	Manifest changes are checked		
	F/O	Down to the line		

(*continued*)

TABLE 1 (*Continued*)

Time	*Actor*	*Verbal Transcript*	*Other Behavior Transcript*	*Interpretations*
	S/O	We are down to the line		F/O and S/O acknowledge completion of the first portion of the Before Takeoff Checklist

Note. Text marks indicate the following: /…/ indicates pause; \ indicates overlap between speakers; [] indicates ambiguity in interpretation of speech; {} used to augment transcript with description; and xxx indicates unintelligible utterance or speech unit.

Checklist execution. Checklist execution proceeds with the S/O challenging on each item and the F/O responding, while the Capt. steers the aircraft with the tiller. The F/O has a "data card" mounted above the throttle quadrant, which he received from the S/O earlier, listing precomputed information, including takeoff speeds, engine pressure ratios (EPRs), and flap settings. The F/O and Capt. both have readily visible copies of the checklist clipped to their respective yokes. Note *the F/O's anticipation of the S/O's* call to check the speed brake at 4430. It is likely that his sequence of motor actions here—data card, flap position indicator, flap handle, speed brake handle—constitute a "flow" of arm movements that are quite automatic for him (cf. Hutchins, 1986). He might also just be remembering two items from the checklist, consciously choosing to attend to the sequence of items as a chunk rather than returning his attention to the checklist between checks.

Checklist interruption. At 4438, the Capt. attempts to elicit help in establishing visual contact with the taxiway by gesturing toward the runway. He vocalizes this explicitly 4 sec later (at 4442), interrupting the crew's activity of executing the checklist. The F/O offers his help to the Capt. and then tries to immediately continue with the checklist (at 4447), but is interrupted by SMF-GND, which provides some guidance for taxiing.

Checklist resumption. The Capt. signals the resolution of his problem (at 4453). *The S/O interprets the Capt.'s actions to mean that the interruption to the checklist process is over* and he resumes the checklist process (at 4455 and 4456) by repeating his earlier challenge for the Compass Indicators check item.

Checklist completion. The S/O and F/O announce that the checklist is completed "down to the line" (at 4530), referring to the dashed line on the checklist that separates taxi items from items remaining to be checked immediately prior to takeoff.

At this point the entire *crew shares the knowledge* that the aircraft is configured for takeoff, at least "down to the line" of the Before Takeoff Checklist. Of course, many failures could occur that the configuration check cannot possibly discover. But the majority of configurable options of the aircraft—options that are necessary to provide safe flight under a wide range of conditions—have now been set to the desired state for this takeoff and checked that they are set. The crew's shared knowledge *could* be just a by-product of the public proclamation "we're down to the line." That is, the speech act "we're down to the line" is *both* a product of the checklist routine (i.e., it is the last item published by the challenger S/O) and a designator of a state of readiness of the aircraft (i.e., it stands for a point in a stream of activity that in most cases, culminates in safe and efficient takeoff).

These two properties of S/O's speech act are obviously related by the practice of executing the checklist; having performed the checklist in the culturally prescribed manner, each crewmember "knows" the state of the aircraft without needing to hold any particular representation of that state in his or her head. The "knowing" is with respect to a set of expected outcomes given a set of performed actions; having configured the aircraft by the prescribed wisdom, certain takeoff performance should be experienced. The proposition "The aircraft is safely configured for takeoff" is known both personally (through subjective experience of having piloted takeoffs before) and collectively (by the results of a cultural process that has created the aircraft, crew culture, regulatory institutions, accident investigations, etc.).

Along the way, the crew may encounter—and did encounter in this episode—a number of "distractions" that might have derailed the checklist procedure, creating a disparity between subjective "knowing" and objective fact about the readiness of the aircraft for takeoff. Since the crew's "knowing" is a product of their joint activity of executing the checklist, we need to characterize this interaction to understand its strengths and weaknesses and thus its role in the functioning of the cockpit as a system.

A DESIGN FOR A MULTIPLE ACTIVITY SYSTEM

Miyata and Norman (1986) draw on an analysis of the multiprocessing capabilities of computer users to suggest design principles for multiprocess computing machines. Here I employ several of Miyata and Norman's observations about human–computer interaction (paraphrased *in italics* below), extending them to characterize the interindividual nature of team performance in the cockpit.[8]

[8]Miyata and Norman (1986) use a theoretical framework based on mental "schemas" that structure the resources available to individuals for actions. Within the framework proposed here they have restricted

Conscious and Subconscious Control Systems

Two different systems in the individual form the basis for two different kinds of processing capacities. Conscious control seems to be limited to single-task processing and is well-suited to (a) critical or difficult tasks through focusing of attention that acts to improve performance by allocating sufficient resources for the job at hand, (b) situations that call for overriding automated procedures and resolving conflicts among competing schemas or tasks in favor of critical ones, and (c) tasks that are novel or ill learned. Subconscious control provides the ability to do multiple, independent tasks that are generally well learned or automatic but that often lack extensive error checking and redundancy due to specialization of resource utilization.

The group analog to these two modes of cognitive activity constitutes more of a gradient of attentional resource allocation—namely the gradient represents the extent to which shared attentional focus leads individuals in the group to mutually engage a shared task and thus converge upon a single shared perspective regarding the task situation. All three members of the crew collectively performing a task is similar to "conscious control" of the process, whereas completely individual and independent task performance is similar to "subconscious control" by the group. The gradient between these two extreme poles thus represents a distribution of awareness about the task situation, including group members' intentions and expectations.

By definition, complete crew involvement in every aspect of executing a checklist would entail mutually redundant attention to each single step involved—an example of conscious process control at the group level. While this is impossible in reality, it provides a convenient theoretical pole for our discussion. It is clear that executing a checklist in fact constitutes a critical task and that the multiple-participant design was meant to allocate a lot of cockpit resources to this task. The format seems to ensure not only that individuals consciously attend to the setting of the aircraft state but also that the *group consciously attends* to it (see following paragraph)—both result from the configuration of resources required for this activity by the prescribed challenge–response format.

---

their attention to individual-centered cognition mediated by mental constructs and their constituent constraints. Although I am not denying that this is both important and useful to describe, the claim being made here is that this is not a *sufficient* characterization of human cognition or activity. In particular, it constitutes only one level of cognition/computation that takes place in the cockpit. Furthermore, a schema-theoretic analysis of individual cognition is only *explanatory* to the extent that the latter operates independently of other levels in this cognitive system. Nevertheless the properties Miyata and Norman identify, in both individual and systemic (machine) terms, are suitable for characterizing many aspects of cognition in the cockpit. The goal here is to extend these notions about multiprocessing systems to the interindividual level of cognitive activity in the cockpit.

Executing a checklist entails not only jointly engaging in the activity itself, the challenge–response format forces this, but also intersubjective sharing about the state of the aircraft.[9] As a result, each member not only learns the state and what the other members know but also is assured that the others know about their knowledge of the state of the aircraft. These three kinds of knowing are critical for maximum coordination among the individual crewmembers in the performance of actions that require knowledge of the state of the aircraft. If you and I are crewmembers, my knowing of the state of the aircraft enables me to coordinate my activities with it; your knowing of my knowing enables you to interpret my actions and coordinate with those as well. Finally, my knowing that you know that I know the state of the aircraft means that I am more certain you can interpret my actions and therefore I can, reciprocally, more easily coordinate with your actions. This phenomenon entails what I am calling interindividual (or group) conscious control of processing, since the resultant processing converges on the extreme pole of individuals engaging the task with a single, shared perspective.

When individual crewmembers check aircraft state, they have to both visualize the setting and vocalize the state seen. This reduces the possibility of using visual or vocal systems for other tasks—that is, it requires conscious control, on the part of the individuals, to accomplish the verification task. Furthermore, institutionalized prescriptions for the enactment of each challenge and response can potentially constrain variability in individual performance by distributing the cognitive labor and encouraging social conformity. In other words, the challenge–response format implements group conscious control of checklist execution.

The tendency for individuals to automatize well-learned actions, thus transferring them to subconscious control, appears to be pervasive in human behavior. In the episode recounted above, we saw how the F/O's automatic motor sequences probably entailed a deviation from a strictly coordinated flow of crew interaction —the F/O attended to the speed brake handle before the S/O challenged for this check to be performed. It is possible that this deviation could subsequently hinder the S/O's ability to "verify" the state himself, something he will often accomplish by attending to the physical behaviors of the responder.[10] In this case, the F/O's response ("it's down") followed 10 sec after he actually attended to the speed brake (part of an automated manual sequence of his own making) and rendered the verification action invisible to the S/O (who, in general, is responsible for double-checking the state for himself). This discoordination in the normal sequencing of

[9]As D'Andrade (1987) has commented intersubjectivity seems to be a fundamental component of social life, leading to our highly proficient abilities for interpreting actions and sharing expectations about those actions.

[10]Although in principle S/O is supposed to verify the actual state of the configuration being checked, the actions taken by the responder to his challenge serve as an important indicator of that state. In fact, it's likely that S/O cannot even see some of the devices that he is supposed to visualize in order to verify their state.

the challenge and response, although minimal and probably inconsequential in this case, could increase errors in the verification process. The distribution of cognitive labor, entailing the passing of process control between crew members, may mitigate the tendency of individuals to resort to automatic behaviors under subconscious control,[11] and this may have performance consequences for the cockpit system.

Task-Driven and Interrupt-Driven Processing

Concomitant with the two modes of process control are two modes of processing. Task-driven processing characterizes the typical mode of processing under conscious control. Attentional focus and processing resources are largely allocated to one task, preventing both distraction from and concurrent processing of other tasks. Interrupt-driven processing is largely enabled by subconscious control. It requires specialization in resource utilization, reducing the redundancy in resources that makes possible accurate performance and error detection. On the other hand, it provides for simultaneous performance of tasks that are relatively independent—that is, tasks that do not rely on the same resources can be executed in parallel.

Task-driven processing at the interindividual level is characterized by group fixation upon singular tasks. This feature is desirable for those tasks deemed to be both critical and embedded in a predictably stable environment. Executing a Before Start checklist, where the environment seldom introduces unexpected time-critical new tasks, is a good example. However, this kind of processing can also be disastrous. The L-1011 jet, which crashed into the Florida Everglades in 1972, might be an extreme example of the unsuitability of task-driven processing by the crew.[12] In this tragic incident, a malfunctioning landing gear light captured the crew's attention. While collectively attending to the task of troubleshooting this malfunction, the aircraft slowly lost altitude and contacted terrain.

---

[11]Humans also have a capacity for clothing social ritual in the appropriate guise while drifting with respect to the meaning (and thus intended effects) of behavior. Thus even the elaborate checklist procedure can be executed without attention to the details of actions if crewmembers implicitly agree to be sloppy in this regard. It has been suggested among our research group that the airlines' policy of maintaining constant flux in crew membership may alleviate one tendency for subcultural drift that results from crewmembers working extensively together. On the other hand, it appears that an increase in communicatory behaviors (yielding improved crew performance) results from some increase in time spent working together as a crew (cf. Foushee & Helmreich, 1988). The prediction made here is that this coordination could proceed (over the long term) at the expense of both individual and group "conscious control" of the tasks so coordinated.

[12]This incident was perhaps extreme, but certainly not unique. Foushee and Helmreich (1988, p. 195) also mention a 1978 DC-8 crash—a few miles short of Portland airport—caused by the aircraft running out of fuel while the crew was debugging a minor problem. "Again, a burned out bulb denied the crew a green light on the landing gear, and again, the crew became preoccupied with trying to diagnose

However, crew activities seem to be more profitably, and realistically, characterized by degrees of interrupt-driven processing taking place throughout this distributed cognitive system. Interruptions, considered in the broad context of a multiple-agent computational system, are ubiquitous—almost every event will act to interrupt some computation or cognitive act that is in process. Interruptions then, like computational processes themselves, must be examined in relation to the activities and information transforming actions which are, in real time, affected by the new constraints introduced by interrupting events.

Demand for flexibility in cockpit activities, which stems from the inherently dynamic environment, yields much interrupt-driven processing. The PNF must tend to the following tasks: navigation, ATC communications, the PF's actions, visual traffic, and monitoring the S/O's actions. In the episode recounted above, we saw how crewmembers rely on each other to solve problems that arise unexpectedly in their individual tasks. The Capt., unable to make visual contact with the runway while steering the aircraft, enlisted the F/O's help in the middle of executing the checklist. The costs of this kind of interrupt-driven processing come about through added coordination overhead, as discussed later.

Current and Suspended Processes

Current processes can be either foregrounded (under conscious control) or backgrounded (under subconscious or an external agent's control). Suspended processes are those processes, recently current but now inactive, that await resumption as current processes.

Conscious control of task-driven processing, as we have seen, can be described by the method and extent of crew attentional focus upon a mutually shared task. To the extent that the task requires crew interaction (or, conversely, is performed solo), we might say it is foregrounded (or, backgrounded) with respect to group processing. Furthermore, we might say that *external backgrounding* includes the distribution of a computational process to an "outside agency," such as ATC (e.g., establishing a schedule for attaining a desired flight altitude) or the aircraft itself (e.g., recording an altitude clearance in the Flight Management System and being forewarned when it is approached). *Suspended* processes at the group level are those that were current but then, perhaps because of a computational resource conflict due to an interruption, are put on hold and must await explicit reexecution or resumption.

---

the problem and decide on a strategy." In 1981, a Lockheed Jetstar—a small corporate jet—crashed on approach into Westchester County Airport in White Plains, New York, in gusty winds and poor visibility. The NTSB (1981) report concluded that the two-man crew was distracted by a puzzling yet non-critical generator failure, causing an "undetected deviation of the aircraft's flight path into terrain."

Backgrounding capability primarily benefits the crew by releasing allocated resources. Of course this release often comes at the expense of additional overhead for coordinating disjoint or parallel processes (for example, mechanisms for mutual understanding of crew roles and responsibilities) or protocols for communication with ATC. Automation in the cockpit entails a similar taxation of resources to take advantage of backgrounding capabilities. Automatic devices, checklists, and procedures in the cockpit must be learned and coordinated—they present tasks themselves (Degani & Weiner, 1990; Norman, 1991). Of course, the other problems associated with relinquishing conscious control—such as reduced resource redundancy and error-checking capacity—also apply to backgrounding tasks.

Suspension of current processing seems like a critically important feature of any complex cognitive system. But it is only useful if the organization of processes suspended is appropriate and their reactivation timely. Miyata and Norman (1986) identify several principles of computer system design aimed at supporting these notions about the interruption and effective resumption of multiple activities. I cite them here to evaluate the effectiveness of the three-person 727 cockpit as a multiprocessing computational system. In other words, given Miyata and Norman's proposals for the design of a system that must support multiple activities while being sensitive to the individual cognitive properties of humans, how does an evolved multiple activity system composed of humans (the cockpit crew) implement these properties?

Design criteria for a multiple activity system includes:

1. *Easy interruption, nondisruptive to initiation of interrupting process.*
2. *Sufficient saving of context to enable resumption of suspended process.*
3. *Establishment of a reminding structure with following properties:*
   - *inform when conditions are appropriate for resumption*
   - *inform when urgency dictates immediate resumption*
   - *don't distract from current activity*
   - *organize multiple suspensions*
4. *Restore saved context upon resumption of suspended activity.*

The structure of crew interaction during the episode recounted above exhibits many of the principles of this design. In particular, while the crew is taxiing and executing a checklist, team performance allows for the following:

1. Easy interruption. When the Capt. needs assistance navigating the taxiway, or a radio response is required from the F/O, the checklist process is easily suspended. Similarly, the process would be easily suspended should the S/O need to attend to something more important or time critical. This is so because the S/O, as challenger and keeper of process state, is the (theoretical) central processor in

checklist execution and the F/O and Capt. have (in theory) no state that must be saved during an interruption to the process.

2. Sufficient saving of context. The S/O is a member of the cockpit who is dedicated to the task of invoking all items on the checklist in the order represented and culturally prescribed on the physical list in his or her hands. If this process is interrupted, as it often is, it is the S/O's responsibility to store the state of the task because it is the S/O who must produce completion of the process. The S/O stores process state through utilization of his or her own short-term memory to encode the last item executed or, more likely, keeps his or her finger on the required next item on the physical list itself. The other crew members also each have a copy of the checklist visually available, providing redundant access to information about where the task was interrupted and where it should resume. However, having access to the information is different from having the impetus to act on it. Some of this impetus is provided for the Capt. and the F/O by the checklist being clipped to the yoke in front of their faces and on top of other paperwork that needs attending to in order to plan for the takeoff.

3. Resumption of suspended process. It is the S/O's responsibility to drive checklist execution to completion by reaching the bottom of the list. Through monitoring and understanding the cockpit activity, the S/O is able to establish when resumption of the checklist activity should occur. In other words, the S/O watches and understands when the other crewmembers are free to resume the checklist task. At this point, the S/O resumes the suspended process, thus minimizing disruption to other critical tasks that might require crewmember attention. Although Capt. and the F/O have no formal role in saving and reinvoking the task, in practice they have the expectation that the checking process will resume and that their vocalization of that checking will resume, and this may lead them to prompt for the resumption of the process.

4. Restore saved context and resume task execution. If the S/O remembers the last item on the checklist that was verified before interruption, he or she can now consult the physical checklist in his or her hands to resume the checklist task in the appropriate place. Often, the S/O's finger is already positioned on this next item and it is invoked directly. In either case, the physical checklist greatly simplifies resuming the task because only a pointer to the list needs to be saved during interruption. Devices that mechanically implement this pointer without sacrificing the versatility of a paper checklist may provide great benefit to the configuration verification process.

CONCLUSION

John Lauber (former member of the National Transportation and Safety Board; 1989) has investigated several large commercial aviation accidents that involved crew failure to verify aircraft state before takeoff. Invoking a "schema activation" model of individual cognition, Lauber is led to the following conclusion regarding the execution of checklists in the cockpit.

> It must be recognized that any disruption or interruption of sequentially dependent tasks is associated with a high probability that some or all of the elements of these tasks may be missed entirely, especially if any significant amount of time passes during the period of interruption. Thus, operating procedures should explicitly state that any interruption to an ongoing sequence of activities, especially running checklists, will automatically trigger a restart of the process which was interrupted. (p. 10)

While properly motivated by real-world aviation incidents, Lauber's approach leaves out any analysis of how group interaction might structure the task of executing a checklist and ensure its completion. Lauber's suggestion for dealing with interruptions entails a brute-force "start over" prescription, which seems unrealistic and unworkable. This prescription does not help explain what constitutes an interruption and therefore what particular events count as ones that should "trigger a restart of the process which was interrupted."

The analysis given in this article makes explicit some of the cognitive properties of a three-person 727 crew that ensure adequate recovery from interruptions to checklist activities. It is clear that many of the crew members' actions implement desirable aspects of a multiple activity system. For instance, by knowing the meanings of subtle actions of the two pilots, the S/O is able to know not only the right time to resume a suspended process but also the right time to suspend a process in the first place. When the S/O's ability to perform this monitoring is taxed—for instance, when he or she has other duties to attend to or other tasks in process—degradation in this aspect of the system is noticed. This situation was evident later in this flight (although not examined in this article), when the S/O's attention to a data computation task associated with landing parameters hindered his ability to coordinate with the Capt. on a critical irregular procedure employed to troubleshoot a failed generator.

By sharing parts, but not all, of a task (that is, by way of distributing cognitive labor across time, space, and social organization), group performance can be made more robust to errors (cf. Hutchins, 1990). Not being able to see something for oneself (one example of lacking the resources necessary to represent directly the state of the environment) requires attending to the performance of *others* for interpreting objective states of affairs. This kind of "reading reality through attention to others" is pervasive in social life and it is important to understand its role in human activity systems. As Foushee and Helmreich (1988) and others have noted, this phenomenon can generate cultural norms prescribing ways of monitoring based on principles *other* than those most conducive to coordinated teamwork.[13] In this article, we have shown how the interactions among members of the team create robust mechanisms for ensuring safe configuration of the aircraft using the Before Takeoff checklist procedure.

Distributing a task across crewmembers enables access to a large body of resources and skills for their effective employment, but managing this organization takes effort. This effort is realized on both the micro timescale of actions and the macro timescale of practice evolution. Checklist execution is an instance of distributed tasks generating overhead (in the form of additional resources) for coordinating multiple processes. Executing a checklist exemplifies using a lot of resources in the cockpit to accomplish a small number of safety-critical things. We have seen how the activity of checklist execution organizes the crew into a mode exhibiting "conscious control" of action. Here the institutionalized challenge–response format implements a robust method for coordinating actions to ensure the desired outcome—a prescribed configuration of the aircraft and shared knowledge of that state.

What are the implications of this method of analyzing cockpit activity for the practical enhancement of aviation safety? It seems fair to say that designers, developers, and managers of technology generally tend to employ ever more modern technology to eliminate manual tasks; yet they do so with an incomplete understanding of the tasks in place. This tendency seems to direct evolution of the system, and the preponderance of seeing adverse events as incidents of "human error" only fuels this process. While the strategy of blanket removal of manual tasks has proven productive at times, this analysis makes it clear that the resources that structure cognitive activity in the cockpit are far-reaching and consideration of them will help the design process. The actual use of a checklist seems to entail a number of group-level computational properties that serve to ensure interruption recovery and provide the error checking and redundancy of resources required to carry out appropriate actions. Cockpit designers need to consider what the effects of new technology will be for these computational properties. This can be accomplished only through a more complete understanding of these properties.

Palmer and Degani (1991) conducted an experimental study of the use of automated and manual checklists in the cockpit. The authors' general finding was that automation of the checklist process (including automated sensing of aircraft state) led to reliance upon the technology, effectively subverting the verification function expected of the crew. The study found that paper checklist technology yielded better results than the automated checklists when the process was performed as a private, individual task. The inference here is that crewmembers put too much trust in

[13]Attention to these issues have led to attempts to modify cockpit culture based on studies that try to correlate modes of communication and team performance (Foushee & Manos, 1981; Foushee, 1984; Foushee & Helmreich, 1988; Kanki, Lozito, & Foushee, 1989; Orasanu & Fischer, 1991).

the automatic sensing device and failed to verify the aircraft state for themselves. However, the challenge–response format acted to override this difference in performance between checklist technologies. In the terms used here, *group conscious attention to the task (including the passing of process control between crew members) reinstalls individual conscious attention to verifying aircraft state.*

Palmer and Degani's general finding was that the automation of the verification process led pilots to accept what the computer indicated without actually performing the check. In the context of a culture that drives automation forward in design of the cockpit (and other work environments), perhaps this trusting behavior by pilots should not come as a surprise. These experimental findings are consistent with what has been argued here. The use of a checklist in the traditional challenge–response format seems to entail a number of group-level computational properties that serve to ensure interruption recovery and provide the error-checking and redundancy of resources necessary to implement a robust verification process.

The analysis of this article makes clear that checklist execution in the three-person B-727 cockpit implements processes bearing on the effectiveness of the verification task that go well beyond the types of computations that are typically off-loaded to computer technology. The main argument of this article is *not* that we should retain the flight engineer in the cockpit (this is a moot point since, for the most part, the flight engineer has already been automated away). Rather, the argument is that we need to understand how the cockpit works as a system before (or, at the very least, at the same time as) we submit it to surgery for repair or technological enhancement. Our science of situated human cognition lags far behind our ability to engineer powerful technologies that can be inserted into the cockpit. This article advocates for a "Hippocratic design principle," namely "in the production of enhancements and repairs to the system we should do no harm." To accomplish this in practice, we need a better science for characterizing how the cockpit works as an implementation of a multiple activity system. This article provides some ideas for how this science might proceed through appealing to the perspective of distributed cognition.

ACKNOWLEDGMENTS

The research reported in this article was conducted while the author was a graduate student and postdoctoral fellow at the Departments of Anthropology and Cognitive Science at University of California, San Diego. The research was supported by a grant to Edwin Hutchins and Donald Norman from NASA–Ames Research Center. The article has benefited from extensive comments by Ed Hutchins, Nick Flor, Christina Halverson, Barbara Holder, and other members of the Distributed Cognition Laboratory at UCSD. The author also thanks the reviewers at IJAP for their thoughtful remarks and Jen Coury and Vicki Hopkins for their editorial support.

REFERENCES

D'Andrade, R. (1987). A folk model of the mind. In D. Holland & N. Quinn (Eds.), *Cultural Models in Language and Thought.* Cambridge, UK: Cambridge University Press.

Degani, A., & Wiener, E. (1990). Human factors of flight-deck checklists: The normal checklists (NASA Contractor Report 177549). Moffet Field, CA: National Aeronautics and Space Administration.

Foushee, H. C. (1984). Dyads and triads at 35,000 feet: Factors affecting group process and aircrew performance. *American Psychologist, 39*(8), 885–893.

Foushee, H. C., & Helmreich, R. (1988). Group interaction and flight crew performance. In E. Weiner & D. Nagel (Eds.), *Human Factors in Aviation.* San Diego: Academic Press.

Foushee, H. C., & Manos, K. L. (1981). Information transfer within the cockpit: Problems in intra-cockpit communications. In C. Billings & E. Cheaney (Eds.), *Information transfer problems in the aviation system* (NASA TP 1875). Moffet Field, CA: NASA-Ames Research Center.

Hazlehurst, B. (1994). *Fishing for cognition: An ethnography of fishing practice in a community on the west coast of Sweden.* Unpublished doctoral dissertation, University of California, San Diego [on-line]. Available at: http://www.dcog.net.

Hutchins, E. (1986). Mediation and automatization. *Quarterly Newsletter of the Laboratory for Comparative Human Cognition, 8*(2), 47–58.

Hutchins, E. (1990). The technology of team navigation. In J. Galegher, R. Kraut, & C. Egido (Eds.), *Intellectual Teamwork: Social and Technical Bases of Cooperative Work* (pp. 191–219). Hillsdale, NJ: Lawrence Erlbaum Associates.

Hutchins, E. (1996). *Cognition in the Wild.* Cambridge, MA: MIT Press.

Hutchins, E., & Palen, L. (1997). Constructing meaning from space, gesture, and speech. In L. B. Resnick, R. Saljo, C. Pontecorvo, & B. Burge (Eds.), *Discourse, Tools, and Reasoning: Situated Cognition and Technologically Supported Environments.* Heidelberg, Germany: Springer-Verlag.

Kanki, B. G., Lozito, S. C., & Foushee, H. C. (1989). Communication indices of crew coordination. *Aviation, Space and Environmental Medicine, 60*(1), 56–60.

Lauber, J. (1984). Cockpit resource management in the cockpit. *Air Line Pilot, 53,* 20–23.

Lauber, J. (1989, April 19). *Remarks of John K. Lauber, Member of the National Transportation Safety Board, before the Flight Safety Foundations 34th Annual Corporate Aviation Safety Seminar* [Manuscript]. Washington, DC: National Transportation Safety Board.

Miyata, Y., & Norman, D. (1986). Psychological issues in support of multiple activities. In D. Norman & S. Draper (Eds.), *User Centered System Design.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Norman, D. (1991). Cognitive science in the cockpit. *CSERIAC Gateway, 2,* 2.

Orasanu, J., & Fischer, U. (1991). Information transfer and shared mental models for decision making. In *Proceedings of the 6th International Symposium on Aviation Psychology*. Columbus: Ohio State University.

Palmer, E., & Degani, A (1991). *Electronic Checklists: Evaluation of two levels of automation.* Paper presented at The Sixth Symposium on Aviation Psychology, April 29–May 2, 1991, Columbus, OH.

Segal, L. (1990). Effects of aircraft cockpit design on crew communication. In E. Lovesey (Ed.), *Contemporary Ergonomics.* London: Taylor & Francis.

Manuscript first received September 2001