Sequences of Actions for Individual and Teams of Air Traffic Controllers

O. U. Vortac and Mark B. Edwards
University of Oklahoma

Carol A. Manning
Civil Aeromedical Institute
Federal Aviation Administration

ABSTRACT

Air traffic controllers participated in high-fidelity simulations of en route air traffic, either singly or with a second team member. The observed stream of time-stamped behaviors and communication events was analyzed using the Pathfinder scaling algorithm, which provides a directional graph of the latent structure in the data. The graphs were found to be similar across levels of traffic complexity, and the triggers for frequently co-occurring activities were equivalent for the individuals and the teams. This suggests that numerous aspects of air traffic control performance are robust and transcend some powerful situational variables. The implications for interface design and automation are discussed.
1. INTRODUCTION

Air traffic control (ATC) represents a highly complex and cognitively demanding task. The steady increase in air traffic during the past decade has raised controller workload to the point where additional traffic cannot be accommodated without automation (cf. D. Hughes, 1993). Ideally, ATC automation would relieve controllers of tedious routine activities without adversely affecting the cognitive skills that continue to be essential for decision making. By implication, the current ATC task ought to be analyzed to permit identification of such routine activities and allow their differentiation from cognitively relevant task components. In this article, we present an exploratory sequential data analysis (ESDA) approach that appears suited for this purpose. We report the results of an observational study involving air traffic controllers and tentatively identify several activities that represent candidates for future automation.

Previous research on ATC has illuminated numerous aspects of controller performance but has not described the temporal structure and sequential interrelation of the various subtasks we claim are important for the successful design of future automated interfaces. Our efforts can be pre-
presented in the context of the introductory article to this special issue on ESDA (Sanderson & Fisher, 1994). We began with an initial ethnographic stage, during which preliminary observations were made to set the stage for later data collection. Next, we borrowed from the behavioral-statistical tradition by choosing to classify observable behaviors and communication events. The transitions among these activities were analyzed using a scaling algorithm, which yielded a preliminary behavioral-statistical model of the controller’s task (Vortac, Edwards, Jones, Manning, & Rotter, 1993). Our interpretation of that descriptive model resulted in a preliminary, and cognitively inspired, hypothesis regarding automation and interface design (Vortac, 1993)—a hypothesis that is explored with further data in this article.

1.1. Automation of En Route Control

In the United States during the latter half of the 1990s, current ATC equipment will be replaced by a new computer interface (Ammerman & Jones, 1988). This change will be particularly visible in en route control, which handles the airspace between major-departure-and-arrival airports and corresponds approximately to the high-speed and high-altitude cruise between takeoff and landing. The current principal tools of the en route controller are radar and a paper flight progress strip (FPS) that records auxiliary flight data. Under the new system, the radar display will be considerably refined and updated, although its functionality will not be significantly altered. The paper FPS, on the other hand, will disappear and will be replaced by an electronic display.

The FPS includes 31 fields of information about a flight—for example, route, type of aircraft, estimated time over a fix. FPSs are mounted in plastic holders stacked in a posting board (or “bay”) next to the radar display.

FPSs are based on the flight plan submitted before takeoff and are printed by computer before an aircraft enters a controller’s sector. An FPS is thus primarily based on projected altitudes, speeds, routes, and arrival times. After a flight enters the controller’s sector, and after the FPS has been moved from a “suspense” to an “active” bay, it must be manually updated to reflect control instructions. While a flight is active, the control-
ler frequently interacts with the corresponding FPS. At least part of this interaction is mandated by the FPS's role as the legal record of the flight.

During the latter half of the 1990s, FPS functions will be transferred to a largely automated electronic flight data display (FDD). The FDD is a 20-in. x 20-in. color monitor that will display flight data as flight data entries (FDEs), the electronic equivalent of FPSs. Like FPSs, FDEs can be sequenced, updated, and changed by controllers. On the other hand, FDEs will differ from FPSs in numerous important ways.

It has been suggested that the conversion from paper FPSs to electronic FDEs is nontrivial, owing to the unique cognitive and behavioral functionality of FPSs. In particular, FPSs have been said to embody important communicative functions by permitting nonverbal exchange of information among controllers (Shapiro, J. A. Hughes, Randall, & Harper, 1991). In addition, it has been claimed that FPSs serve as a memory aid (e.g., Hopkin, 1988, 1991; Jackson, 1989). In support, interviews with controllers have identified FPS management, physical offsetting of FPSs within the bay, and FPS marking as the most frequently used memory aids (Gromelski, Davidson, & Stein, 1992). Similarly, Means et al. (1988) found that controllers relying only on FPSs looked further ahead in terms of conflict detection than controllers who also had access to radar.

Why does the FPS play such a potentially important role in ATC performance? Consider the routine activity of accepting a handoff from an adjacent sector, which involves manual placement of an FPS in the active bay when a flight enters the controller's sector. This seemingly trivial activity requires that the contents of the FPS be encoded and then be compared to that of other FPSs already in the bay. Thus, the simple act of placing the FPS into the active bay forces the controller to integrate new data with existing data, presumably consolidating "the picture" of the air traffic.

Several additional ways have been cited in which FPSs may support the controller's cognitive processes (for a review, see Vortac & Gettys, 1990). Of particular importance is the notion, derived from an earlier examination of the behavior of individual air traffic controllers (Vortac, Edwards, Jones, et al., 1993), that complex behavioral sequences reflect underlying behavioral or cognitive modules.

## 1.2. Modularity

Hayes-Roth (1977) proposed that cognitive processes that repeatedly co-occur will eventually become unitized, forming an independent cognitive structure—a cogit, as Hayes-Roth called it. Processing is most efficient when a cogit exists for the task at hand. If, instead, cogs must be combined, then additional time and resources are required. Similarly, and more important in the current context, if a task requires that a cogit be
Sequences of Actions in ATC

Broken apart—fractionated, according to Hayes-Roth—then, again, additional time and resources would be required.

Modular Automation. Expanding on Hayes-Roth's (1977) notions, we have argued that modularity is an important factor when transferring skilled operators to a new automated system (e.g., Vortac, 1993). In particular, Vortac (1993) argued that uninformed attempts to automate a complex system could fractionate an existing cognitive module (by automating only a part of a unitized sequence), in which case transfer to the new environment may be difficult. In the extreme case, the additional cognitive demands engendered by fractionation of a module could counter any advantages otherwise provided by the automation. In addition to the work reported by Hayes-Roth (1977), ongoing work in our laboratory suggests that fractionating a module can significantly delay mastery of the automated system. This calls for development of a quantitative technique that allows a cognitive engineer to enter an environment, observe behavioral sequences, discern modules, and prevent their fractionation during redesign of the interface.

To illustrate, suppose an analysis of a complex task with subcomponents A, B, C, ... Z identifies the modules ABC and DEF. If the operator is transferred to a system in which the tasks B, D, and F (but not A, C, and E) are automated, we would expect negative transfer because any potential gain associated with a reduction in the total number of tasks would be, at least temporarily, offset by the fractionation and the mental resources required to overcome it. On the other hand, if the operator is transferred to a system in which DEF is effectively and completely automated, the effects of automation could be more benign, although additional constraints may enter into determination of the extent of the benefit.

Specifically, the benefits of automation may depend on the role played by a module. For example, a module may represent operator habits that have been developed, through practice, to optimize performance on a set of peripheral subtasks. Alternatively, modules may represent coping strategies that reflect design faults in the interface (akin to Siochi & Ehrich's, 1991, argument that repetitions of action patterns usually indicate user-interface problems). In those two cases, effective and complete automation of a module would likely result in significant overall performance benefits. On the other hand, if a behavioral module reflects an invariance of behaviors associated with the fundamental nature of the overall task, then the module may be better left unautomated, thus leaving the essential element of the task under human control. Regardless, to avoid fractionation, the entire module should either be automated or preserved.

Limitations of the modularity view derive from the personnel situation: Clearly, if new workers are trained, there would be no old modules to
fractionate. Similarly, if skilled workers are placed in a new environment that does not evoke the old modules, then, again, fractionation would not occur. However, if trained personnel are to be transferred to a system similar to the present one, then fractionation would slow down acquisition of new skills. ATC automation clearly takes this evolutionary approach.

**Modularity Criteria.** The modular automation view requires that independent criteria be defined for identification of modules. We posit that a module is suggested by the analysis of transitions between behaviors if:

1. The relevant transition(s) occur relatively frequently and in close temporal proximity.
2. The relevant transition(s) occur in several situations.
3. The module has clearly identifiable and reliable triggers.
4. Activities forming a module do not distribute across individuals if more than one person is involved in the task. That is, a team member would assume responsibility for an intact module; teams of experts would not fractionate modules.

**Modularity in ATC.** We examined the modularity notion in the context of en route control. Earlier analyses by Edwards, Fuller, Vortac, and Manning (in press) and Vortac, Edwards, Jones, et al. (1993) suggested the possibility that FPS activities might form a behavioral module; here, we present additional data to explore and verify our criteria for modularity.

2. **PREPARATORY DECISIONS**

Preparatory decisions were made concerning type of study, choice of behaviors to observe, and type of measurement.

2.1. **Type of Study**

Our goals mandated a study with the highest possible external validity, preferably a nonintrusive observational study using experienced controllers. The study was therefore conducted at the Federal Aviation Administration (FAA) Academy in Oklahoma City using a radar-simulation facility that provided a close replica of the equipment and conditions in the field. All subjects were full-performance-level controllers serving as Academy instructors.

Our first goal was to gather an overall understanding of the ATC system. To this end, we attended controller training classes, visited the Dallas–Ft. Worth en route center, observed students and instructors at the Academy controlling simulated traffic, and interacted extensively with Academy instructors to explore their knowledge of the task. This phase of
the research lasted 12 to 15 months and resulted in a theoretical analysis of the likely cognitive implications of FPS automation (Vortac & Gettys, 1990).

A conceptual overview of the ATC system obtained during this ethnographic stage is reproduced in Figure 1. According to our analysis, the current en route system consists of seven conceptual units that share and exchange information. The extent and direction of information flow is represented by thickness and direction, respectively, of the arrows between concepts. Two controller positions ("R-side" and "D-side") represent the typical staffing level in the field. The R-side primarily monitors the radar display, whereas the D-side mainly interacts with FPSs and communicates with other ATC facilities. Both controllers use a keyboard to interact with the computer that manages flight data. Under low-traffic loads, R-functions and D-functions are often handled by one controller.

2.2. Choice of Behavioral Categories

The context and purpose of this study dictated that FPS interactions be included in the behavioral categories. Our experience also pointed to the
importance of recording communications between team members (R-side and D-side) as well as between controllers and pilots. Means et al. (1988) showed that two actions were performed simultaneously 42% of the time and that a large proportion of these simultaneous activities involved verbal communication while recording data on an FPS (or into the computer). This suggests that communication events and FPS activities may be involved in behavioral modules because they could pass the temporal-contiguity criterion stated earlier.

Last, a decision had to be made concerning the role of radar: On the one hand, radar monitoring clearly represents the most important controller activity; on the other hand, for that very reason it is also least diagnostic. We therefore decided not to record the controller's radar monitoring.¹

The final choice of behavioral categories had to satisfy several constraints. First, the total number of categories had to be sufficient to describe all relevant aspects of controller activities, without becoming unmanageably large. To achieve these potentially contradictory goals, behaviors were classified at two levels of resolution: Level 1 corresponded to broad classifications (e.g., “issue a command”) that were further subdivided, at Level 2, into more specific categories (e.g., “change speed”). Second, categories had to be mutually exclusive and exhaustive. These considerations suggested the set of behaviors shown in Figure 2. The three columns show, respectively, the Level 1 classifications, the acronyms used in the remainder of this article, and brief explanations, including an enumeration of the corresponding Level 2 behaviors. Discussions with experienced controllers and pretesting during several observational sessions before the experiment proper confirmed that these behaviors formed the desired set.

One potential difficulty with the behaviors in Figure 2 is that only a few (PREQ, SECTOR) represent events in the environment. Hence, analysis would have some difficulty differentiating between reactions to situational cues (e.g., conflict between aircraft) and endogenously generated actions. How, then, can analysis of the present set of behaviors contribute to the design of a future interface?

We propose that decisions about design and automation must be informed not only by how operators react to specific cues, but also by the invariance in behaviors that transcend specific situational cues. These invariances can be established without knowledge of environmental cues if performance is observed under a variety of different circumstances, and analysis focuses on the common subset of behaviors. This subset would primarily include actions without clear situational causes, but their se-

---

¹ We did not record controllers’ rare adjustments of the radar display parameters (brightness, contrast, etc.).
**Figure 2.** Behavioral categories used to classify controller activities.

<table>
<thead>
<tr>
<th>Category</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller command</td>
<td>CCOM</td>
<td>Instruct aircraft to change (1) route, (2) speed, or (3) altitude; provide (4) information</td>
</tr>
<tr>
<td>Controller query</td>
<td>CQUERY</td>
<td>Controller requests information from pilot about aircraft: (1) speed, (2) altitude, (3) route, or (4) other</td>
</tr>
<tr>
<td>Pilot request</td>
<td>BREQ</td>
<td>Pilot requests (1) route, (2) speed, (3) altitude, or (4) other change</td>
</tr>
<tr>
<td>Sector transitions</td>
<td>SECTOR</td>
<td>(1) Turnover of control to adjacent sector, (2) departure clearance, (3) initial contact with sector</td>
</tr>
<tr>
<td>Team communication</td>
<td>TEAM</td>
<td>Verbal or nonverbal communication between controllers (for teams only)</td>
</tr>
<tr>
<td>Look</td>
<td>LOOK</td>
<td>For R-side: Look at (1) suspense bay or (2) active bay*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For D-side: Look at radar display</td>
</tr>
<tr>
<td>Write</td>
<td>WRITE</td>
<td>(1) Verification of FPS information, (2) change of FPS information</td>
</tr>
<tr>
<td>Manipulate</td>
<td>MANIP</td>
<td>(1) Move FPS between bays, (2) sequence FPS within bay, (3) offset FPS, (4) flatten (undo offset) FPS, (5) remove FPS from bay</td>
</tr>
<tr>
<td>Update computer</td>
<td>UPDATE</td>
<td>Update computer database</td>
</tr>
<tr>
<td>Obtain computer information</td>
<td>INFO</td>
<td>Examine an aircraft’s flight plan, display an aircraft’s projected route on the radar</td>
</tr>
<tr>
<td>Conflict detection</td>
<td>CONFLICT</td>
<td>Display a circle with a radius of 5 mi of airspace around a specified aircraft</td>
</tr>
<tr>
<td>Computer handoff</td>
<td>HOFF</td>
<td>Accept control of aircraft from adjacent sector or facility</td>
</tr>
</tbody>
</table>

*Because a look obviously precedes other FPS activities, this category included only those looks at the bays that were not immediately followed by writing or manipulating. In addition, multiple LOOKS imply that the controller looked away (presumably at the radar) and returned to look at the strips. A long, single look at the bays, or a search of the bays, was thus coded as a single LOOK.

SEQUENCES OF ACTIONS IN ATC

Because a look obviously precedes other FPS activities, this category included only those looks at the bays that were not immediately followed by writing or manipulating. In addition, multiple LOOKS imply that the controller looked away (presumably at the radar) and returned to look at the strips. A long, single look at the bays, or a search of the bays, was thus coded as a single LOOK.

2.3. Type of Measurement

In order to do justice to the dynamic nature of ATC, and given that we were particularly interested in preserving the temporal context and sequential coherence of subtasks, the onset of each behavior was recorded
together with a time index. Activity durations were not recorded because all behaviors tended to be uniformly brief, typically not exceeding 2 sec. These data can provide information about three performance components: absolute frequency of occurrence of each behavior, temporal proximity between controller activities, and—given the proper statistical model—metric properties between pairs of behaviors.

3. METHOD

3.1. Subjects and Design

Ten subjects, all full-performance-level controllers who last controlled traffic, on average, 7.4 months earlier, participated in two sessions. Subjects had, on average, served 9.3 years with the FAA and had been full-performance-level controllers for 5.7 years. Subjects were familiar with the airspace but were naive to the scenarios selected.

During the first session, individual controllers were observed controlling simulated air traffic in two scenarios—low and high traffic complexity. During the second session, teams of two controllers were observed using a different set of low- and high-complexity scenarios. The experimental design, therefore, involved two fully crossed variables, each with two levels: complexity (low vs. high) and staffing (individual vs. team). For the individuals, activities were classified into 11 categories (those in Figure 2 minus TEAM). For the teams, the 11 activities were recorded separately for the two members, yielding 22 activities, plus the TEAM category, which resulted in 23.

3.2. Scenarios

Scenarios were chosen from the library used by the FAA for controller training. Low-complexity scenarios included an average of 2.8 departures, 3.8 arrivals, and 4.0 overflights and lasted 30 min; in the field, comparable traffic densities typically would not require the presence of a second controller. High-complexity scenarios included an average of 4.2 departures, 8.6 arrivals, and 28.6 overflights and lasted 60 min; in the field, comparable traffic densities would have required the presence of a D-side controller.

3.3. Observers

Four individuals took turns serving as observers. All four participated in the initial exploratory phase and had acquired considerable skill in observing air traffic controllers. Two of the observers had taken part in the FAA training program. Interrater reliability, when observing individual
controllers, averaged about 83%. No reliability analyses were available for the team situation, in which each observer was assigned to a different controller.

3.4. Procedure

Before each scenario, subjects were given the opportunity to organize all FPSs into the preferred sequence. All controllers used a single FPS bay located to the right of the radar screen, with one FPS per flight.

Two observers sat behind the controller—one to the left, one to the right. On each observer’s lap was a time-synchronized notebook computer for on-line recording of the controller’s behaviors. The computers were synchronized with the radar clock, and the observers classified the behaviors, using keyboard function keys, into the categories shown in Figure 2. Level 1 responses for the communication events (CCOM, CQUERY, PREQ, SECTOR) were recorded using the left-hand function keys F1 to F4, and FPS activities (LOOK, WRITE, MANIP) were recorded using the right-hand function keys F6 to F8. For the team conditions, TEAM was recorded using F5. A Level 2 response was recorded repressing a function key with the same hand that was used to record the corresponding Level 1 response. The remaining behavioral events involving the en route computer (UPDATE, INFO, CONFLICT, HOFF) were obtained from an on-line record and were later integrated into the data stream.

Radio and audio communications were recorded using a multitrack cassette recorder. Each observer wore a lapel microphone and was recorded on a separate input channel to allow oral annotation of event recording. In addition, a video record was obtained of all FPS interactions. The audio and video records were used later to correct the observers’ event recordings, where necessary. Figure 3 summarizes the experimental procedure and shows the flow of information from all sources. In Sanderson and Fisher’s (1994) terms, reliance on real-time, on-line classification resulted in a manageably low ratio of analysis time to sequence time.

Subjects were not informed about the emphasis placed on FPSs and were told to control traffic as they normally would in the field. Each experimental session lasted about 3 hr. Breaks between scenarios were about 20 min each.

4. RESULTS

4.1. Transition Frequency

The data resulting from our procedure formed long, complex sequences, which we summarized in transition matrices. One transition matrix was obtained for each level of traffic complexity for each subject.
3. Overview of sources of information used in the study.

translated data (visual to code) –

voice data , , , visual data

(dimensions = 11 x 11) or team (dimensions = 23 x 23). A particular coded event represented a particular row and column of the matrix. Hence, the datum within cell \( ij \) represented the number of times throughout the scenario that event \( i \) was followed by event \( j \), divided by the total number of transitions.

4.2. Temporal Proximity

The simple transition matrix may be appropriate for situations in which time is either invariant or irrelevant. However, that type of transition analysis does not apply to ATC, where time between events is of critical importance. For example, two commands separated by an extended period of radar monitoring seem qualitatively different from the transition between two commands that immediately follow each other and that may be issued to two conflicting aircraft.
SEQUENCES OF ACTIONS IN ATC

Many psychological models that describe performance as a function of time assume an exponential relation of the general form:

\[ P = e^{-\lambda t} \]  

(1)

where \( P \) refers to some performance measure, \( t \) to elapsed time, and \( \lambda \) is a decay parameter (large values of \( \lambda \) correspond to more rapid decay). Forgetting functions in human memory can often be described by this type of exponential process (e.g., Atkinson & Shiffrin, 1968; Loftus, 1985). An exponential function has also been found to describe the effects of practice in a visual search task (Heathcote & Mewhort, 1993). Given the apparent pervasive applicability of the exponential function, we decided to weight transitions by the time that had elapsed between the pair of events, using Equation 1, with \( P \) taken to correspond directly to the increment of the frequency count; \( t \) was measured in seconds.

The parameter \( \lambda \) was arbitrarily set to 0.1 for all analyses. Thus, two events that occurred simultaneously \((t = 0)\) would produce an increment of unity \((P = 1)\), two that were separated by 1 sec would be given an increment of .90, and so on. Sequential events separated by 20 sec (or more) contributed little to the transition count \((P = .13)\).

The decision to time-weight the transitions was made in order to do justice to the inherently temporal nature of the domain under study. Hence, although we compared the time-weighted matrices to their unweighted counterparts and found differences between the two sets, this article reports only the analyses based on the theoretically motivated time-weighting.

The time-weighted transition matrix is no longer interpretable in terms of absolute frequencies or probabilities, but, unlike the raw data, it represents a preliminary statistical model that takes elapsed time into account. Nevertheless, this preliminary model falls short for two principal reasons. First, the matrices are too complex, with a total of 121 \((11 \times 11)\) transitions for individuals and 529 \((23 \times 23)\) for teams, to be amenable to direct interpretation. Second, and more important, the matrices do not distinguish between those transitions that reflect the latent structure of ATC performance and those present because of noise. We thus extended the preliminary model by recovering the latent structure of the data.

Several psychological scaling procedures have been developed to reveal the latent structure underlying sets of empirical data. Some assume that the underlying structure can be represented by a multidimensional space (e.g., Shepard, 1962), by hierarchical clusters (e.g., Johnson, 1967), or by graphs and networks (e.g., Durso & Coggins, 1990; Schvaneveldt, Durso, & Dearholt, 1989). The scaling procedures are similar in assuming that the observed data reflect the latent ("true") structure and statistical noise and that the two can be differentiated by mathematical means. The assumption that observed data comprises both meaningful structure and
statistical noise is fundamental to measurement. The problem is, of course, separating the wheat from the chaff. Quantitative procedures provide a principled means of separating latent structure from random variations.

4.3. Pathfinder Networks

In an effort to reveal the latent structure in our time-weighted transition matrices, we turned to the Pathfinder scaling algorithm (Schvaneveldt, 1990; Schvaneveldt & Durso, 1981; Schvaneveldt et al., 1989). Unlike multidimensional scaling, Pathfinder is capable of representing asymmetric structures where, for example, the transition from WRITE to MANIP might be more frequent than the transition from MANIP to WRITE. Given the inherently asymmetric nature of time, Pathfinder appears particularly suitable for ESDA applications. The algorithm has been used successfully in a variety of domains within cognitive psychology, engineering, and artificial intelligence (see Schvaneveldt, 1990). Cooke, Neville, and Rowe (in press) used Pathfinder as a basis for their ESDA PRONET toolkit. The mathematical foundations can be found in Schvaneveldt, Dearholt, and Durso (1988).

Pathfinder reduces a matrix of proximities or transitions to an interpretable model by eliminating those connections that do not satisfy metric properties. Any metric must satisfy the triangle inequality, which holds that the summed length of two sides of a triangle must be greater than the length of the third side. Because statistical noise cannot satisfy the triangle inequality, the connections retained by Pathfinder are the latent structure. For Pathfinder, the structure it produces is a geodesic graph or network. That is, the $k$ transitions chosen by Pathfinder should be the shortest distance between all events $i$ and all $j$, given $k$ transitions. Every connection in a Pathfinder network is a connection on some minimal path between two nodes.

Pathfinder is capable of producing a family of networks, depending on the metric used to determine path distance (the Minkowski $r$ parameter) and the maximum path length for which the triangle inequality must be satisfied (the $q$ parameter). That is,

$$w(p) = [w^r]^{1/r}$$  \hspace{1cm} (2)

where $w(p)$ is the weight of the path, $w$ is the weight of an arc on the path, and $r$ is the Minkowski exponent. For example, when $r = 2$, the distance is the common Euclidean distance, where the distance between two points is given by the square root of the squares of the arcs on the path (or $a^2 + b^2 = c^2$). When $r = 1$, the distance between two nodes is the sum of the distances along the existing paths (the city-block metric). When $r = \infty$, in the limit the distance becomes the maximum (the dominance metric).
The particular application of Pathfinder that we used was one guaranteed to produce the simplest network, the minimal-cost network (MCN). This involved setting the \( r \) parameter to infinity, thus causing Pathfinder to compute distance using the dominance metric, and setting \( q \) to \( k - 1 \), where \( k \) is the number of nodes. This forced Pathfinder to eliminate violations of the triangle inequality in paths of any length (cf. Hutchinson, 1989). Thus, the connections remaining are those that are ordinally necessary (Schvaneveldt et al., 1988). Finally, weighted transitions that occurred less than 1% of the time were defined as being infinitesimal and were ignored.

The networks representing the high-complexity and low-complexity scenarios for individuals appear in Figure 4; those for teams appear in Figure 5. The nodes represent controller activities using the terminology...
Figure 5. Pathfinder MCNs for low-complexity (top) and high-complexity (bottom) scenarios for controller teams.

introduced in Figure 2. For teams, behaviors were prefixed by R and D to identify R-side and D-side, respectively. The arcs between nodes, when present, indicate the direction and, by their thickness, the frequency with which transitions occurred.

The transitions included by the Pathfinder algorithm represent a small percentage of the possible total. For individuals, the MCNs retained only 18 arcs (of the possible 121) at each level of complexity; for teams, only 16 arcs (low complexity) or 17 arcs (high complexity) were represented (of a possible 529).3

3. It soon became apparent that an analysis of Level 2 events using this technique was prohibitive. The full Pathfinder networks involving all Level 2 classifications for the individuals condition were constructed, but their interpretation supplied no additional information.
Fundamental Singles

We first attempted to discern if the various networks were similar—that is, if there were any fundamental components that transcended the situational variables (i.e., complexity and staffing). The graphs were quite similar across levels of traffic complexity: There were 13 arcs (of 18) in common for individuals and 14 arcs (of 16 or 17) in common for teams. The fundamental (shared) arcs for the two levels of staffing appear in Figure 6. The amount of overlap between the high- and low-complexity scenarios cannot be attributed to chance (hypergeometric probabilities are virtually zero).

We calculated in-degree and out-degree as a graph-theoretic measure of the central tendency of the graphs. The higher the in-degree, the greater the number of arcs terminating on that node; this is a measure of the
prestige of a node. The higher the out-degree, the greater the number of arcs emanating from that node; this is a measure of the influence of a node. In all four graphs, the (D)WRITE node serves as both the prestige and influence centers of the graph. Here again, the analysis points to a surprising degree of agreement. Given these striking similarities, our first choice was to interpret the fundamental graphs within the criteria for modularity outlined earlier.

4.4. Modular Board-Management

In our data, the WRITE \(\rightarrow\) MANIP connection satisfied the first two criteria—namely, (a) that transitions are frequent between events in close temporal proximity and (b) that they occur in different situations. The connection was present in all four Pathfinder solutions, and its strength with time-weighted data indicated that the two activities tended to occur in close temporal proximity. The remaining two criteria postulate (c) the presence of consistent triggers in a variety of situations and (d) the persistence of a connection when a second person is present. Both of these criteria were met in the present data: Beginning with the individuals, the WRITE \(\rightarrow\) MANIP connection appeared to be consistently triggered either by a sector transition (SECTOR) or by a controller command (CCOM). SECTOR was a reliable trigger to the WRITE/MANIP pair: 65\% of the transitions in the data matrix following SECTOR led to the WRITE/MANIP module. CCOM, although not a trivial trigger, was somewhat less reliable: In the data matrix, if a CCOM had just occurred, WRITE or MANIP would occur next about 1 in 4 times.

Looking for similar configurations in the teams graph (see Figure 5), it became clear which member of the team was performing the corresponding functions identified in the individuals graph (see Figure 4). FPS management was clearly handled by the D-side, as would be expected, with DWRITE connecting to DMANIP. RMANIP is represented only as an isolated node, not connected frequently enough to other activities to be of any note. Hence, the teams data support the notion of modularity of FPS activities, WRITE \(\rightarrow\) MANIP, because these activities are not distributed across individuals. Instead, the triggers for a module may involve actions by the other member of the team. This is exactly what was observed in the present case: The triggers were the same as those in the individuals case, but they tended to be R-side, rather than D-side, actions. Thus, despite being performed by a different person, RSECTOR and RCCOM triggered the DWRITE \(\rightarrow\) DMANIP sequence with a regularity comparable to that for individual controllers: RSECTOR was again quite reliable (50\% of the RSECTORS led to DWRITE \(\rightarrow\) DMANIP), and RCCOM was again a fair, but not flawless, predictor (33\%).

FPS writing and manipulation thus seem to be fundamental to en route controller behavior, and this modular sequence is typically triggered by
sector transitions and controller commands. This is true for all levels of traffic complexity and regardless of whether one person or a team is responsible for traffic control. With a team, the responsibility for the board-management module rests with one controller, the D-side, and the triggers for that module seem to rest clearly with the other controller, the R-side.

Edwards et al. (in press) supplied some convergent validation for several of the conclusions reached here. Edwards et al. conducted a traditional time-series analysis of the same teams data and were able to predict frequency of FPS writing at time $t$ from previous writing (at time $t - 1$) plus the two triggers identified here.

Overall, according to our set of criteria, the WRITE $\rightarrow$ MANIP sequence appears to have all the characteristics of a cognitive/behavioral module—implying that a new automated flight-data interface should not fractionate that important unitized process. Does that imply, then, that the board-management module, on account of representing tedious routine activities, should be completely automated? Or does it follow from the centrality of WRITE that writing or some corresponding keyboard activity should be preserved in the next-generation system? Data from a recent experiment by Vortac, Edwards, Fuller, and Manning (1993) favor the former alternative. Vortac et al. compared en route controller performance under conditions currently prevailing in the field with a situation when all but visual access to the FPSs was eliminated. Briefly, their data suggest that the removal of board-management responsibilities did not impair performance or cognitive processing and, in fact, appeared to benefit some cognitive processes—implying that complete automation of board management (WRITE $\rightarrow$ MANIP) would not be expected to be disruptive.

### 4.5. Polysemous Behaviors

All our conclusions about the controller's task are tied to the behavioral categories that were chosen during the ethnographic phase of the experiment. By their very design, those categories were known to be unambiguous, mutually exclusive, and sufficient to classify controller behaviors. Now, after data analysis, the obvious interpretability of the Pathfinder solutions suggests that, in addition to being technically sufficient, the categories also provide an inherently meaningful description of the controller's task. Nonetheless, future research may benefit from further refinement of the choice of categories, and that refinement can be informed by the present Pathfinder solutions.

Specifically, consider the CCOM nodes in Figure 6. In this analysis, we have considered CCOM as a single category, although in reality a host of commands is associated with that label. Interestingly, the Pathfinder solutions suggest that CCOM is not one indivisible construct. Notice that, for
individuals, CCOM is characterized by a loop on itself and a connection to WRITE. In the teams graph, on the other hand, where there is an RCCOM and a DCCOM, the two arcs appear to be distributed across individuals: There is a loop, but no connection to WRITE, for the D-side, and there is a connection to WRITE, but no loop, for the R-side. It thus appears that the connection to WRITE is intrinsic to the role of the R-side, whereas the loop on CCOM reflects the role of the D-side. In the individual staffing situation, where a controller combines the functions of R-side and D-side, we observed the superposition of these two types of arcs.

This facet of categorizing behaviors—which we refer to as *polysemous behaviors*—is, of course, nothing new. What is interesting, however, is that it may soon be possible to identify polysemous behaviors algorithmically (R. W. Schvaneveldt, personal communication, 1992). This will supply a means of distinguishing different senses of an action.

5. CONCLUSION

We have shown that complex sequences of behaviors in a computer-supported task environment can be explored and understood by applying the Pathfinder algorithm to time-weighted transition matrices. A defining characteristic of the analysis was that it did not consider specific situational cues or controllers’ reactions to them. Nonetheless, by focusing on behaviors that transcended different environmental situations, we were able to provide some strong recommendations for interface design and automation.

5.1. Implications for ESDA

This article presents another application of the Pathfinder scaling algorithm to the analysis of dynamic temporal data. A critical decision in this domain concerns the type of input for Pathfinder; some of our decisions have been similar to those made in other scaling applications, and some have been modeled here using Pathfinder for the first time. Three specific technical issues merit discussion.

First, as in other applications (e.g., Cooke et al., in press), we used co-occurrence data to reflect the relatedness between events. We used matrices containing frequencies of transitions between events as the dynamic analog of static dissimilarities between pairs of stimuli. As is not the case with most applications, however, we used overall, or unconditionaled, proportion of occurrence, rather than conditional transition probabilities. An additional unique feature was the use of time-weighted transition matrices that combined contiguity and temporal separation into a joint index of dissimilarity.

Second, it is important to remember that our transition matrices ignored absolute time. That is, a transition from behavior A to behavior B
that occurred late in the observational period was equivalent to that same transition early in the scenario. The assumption that a transition between two events is context independent and can be analyzed without regard to absolute time is a powerful one that reduces the amount of data considerably.

Finally, a prerequisite for construction of any transition matrix is the creation of mutually exclusive and exhaustive categories to unambiguously classify the ongoing stream of behaviors and events. In our case, these categories were developed during a phase some would term the *ethnographic stage*, when we observed controllers in the field and interviewed various subject-matter experts. For practical purposes, we suggest that the number of categories be kept manageable; our experience dictates a maximum of 15 to 20 categories. If a finer grain of analysis is demanded by the situation, one solution is to create a hierarchy of categories, similar to our Level 1 and Level 2 categories. The presence of a second level of resolution would allow further exploration of the Pathfinder graph. Owing to the complexity of that step, we suggest that it be performed only when seeking confirmation of explicit hypotheses.

Turning to the wider implications of our analysis, the useful properties of the Pathfinder graph must be emphasized. Note that most of the complexity from the original behavior stream is not directly represented in the Pathfinder graph. Far from being a drawback, this reduction provides considerable interpretative power, in the same way that a grammar eliminates much of the complexity of language by reducing it to a set of generative rules or transitions. The Pathfinder solution might therefore be considered to be an ATC "grammar" that represents the typical sequences of behaviors that could be generated by the controller. Although such an empirically derived grammar will likely lack properties of its more formal counterparts, its limited set of behaviors and transition probabilities nonetheless suffices to produce a much larger sequence of legal behavioral sequences. That the grammars derived under different situations were remarkably similar gives additional credence to the notion that one structure generates the apparent complexity of controller behavior.

5.2. Implications for ATC

Two principal findings emerged from the Pathfinder analysis. First, numerous ATC activities, and their sequences, are fundamental to the overall task and do not differ with complexity of the air traffic. We focused on the connection between writing and manipulating FPSs—a connection we refer to as *board management*. This connection was fundamental and substantial under both staffing conditions. Moreover, there exists a set of consistent triggers for these FPS-related activities. Finally, when teams of controllers were considered, we saw that board management was the
purview of one controller, who relied on the actions of the other to trigger relevant activities. For these reasons, it appears that board management has properties that one would hope to see in a sequence of behaviors targeted for automation. In our view, board management is a module.

The second major finding concerns the fluidity of staffing the ATC sector. We found that triggers of FPS activities were not changed by the addition of a second individual into the overall flow of task performance. Duties performed by the individual controller were typically performed by either one member of the team or the other—without substantial alterations in the overall graph structure. Together, these two findings suggest that numerous aspects of ATC performance, and the interrelations among them, are (a) exceedingly robust and (b) unaffected by complexity and, perhaps most strikingly, by the presence of another person.

We have argued and supplied empirical data to support the contention that automation should focus on identifiable modules of behavioral sequences that can be replaced without great disruption. This is especially true when highly skilled personnel are being transferred to a newly automated system. In such cases, there is a risk that the demands placed on the human operator may require pieces of the modules that before were units.

Our views do not stand in conflict with either static or dynamic approaches to automation (e.g., Rouse, 1977). For a system with static automation, we would argue that having the automation take responsibility for a complete module would allow easy interdigitization of the human and the machine. For a system that adapts to the needs of the operator, we would argue that the system should be informed of the modules and that delegation or transfer of modules between human and machine should take place in an all-or-none fashion. If, instead, adaptive automation took responsibility for subtasks that were constituents of a larger module, we would expect transfer and retraining on the new system to be slow and difficult.

A promising example of adaptive, yet modular, automation in the ATC context was provided by Debernard, Vanderhaegen, and Millot (1992). Debernard et al. examined an automated conflict-detection system, SAINTEX, and found that controller workload was reduced when SAINTEX shared in the conflict-detection-and-resolution process. Importantly, when adaptively given responsibility for a conflict, SAINTEX would handle all aspects of its resolution, including communication with the aircraft. This seems to conform to the modularity notion presented here.

Finally, fractionation was implicated in the April 26, 1994, crash of the China Airlines airbus: "Essentially, the crew had to choose between allowing the aircraft to be governed by its automatic pilot or flying it manually. Instead, they flew a half-way measure" (Mecham, 1994, p. 31).
NOTES

Background. A subset of the individuals data from this experiment was reported in Vortac, Edwards, Jones, Manning, and Rotter (1993).

Acknowledgments. We thank the staff and controllers at Mike Monroney Aeronautical Center, Oklahoma City, where this study was conducted. Thanks to Mike Wayda, Dave Schroeder, Dana Broach, and three anonymous reviewers for helpful comments on earlier versions of this article.

Support. This research was supported by Federal Aviation Administration Contract DTFA-0291C91089 to the members of O. U. Vortac.

Authors’ Present Addresses. O. U. Vortac (Francis T. Durso, Scott D. Gronlund, Stephan Lewandowsky), Department of Psychology, University of Oklahoma, Norman, OK 73019. E-mail: ouvortac@oucog1.psy.uoknor.edu; Mark B. Edwards, US West Technologies, 1005 17th Street, Denver, CO 80202. E-mail: mbedwar@denitqm.mnet.uswest.com; Carol A. Manning, Civil Aeromedical Institute (CAMI), Federal Aviation Administration, AAM-514, 6500 South MacArthur Boulevard, Oklahoma City, OK 73169. E-mail: carol_manning@mmacmail.jcobi.gov.

HCI Editorial Record. First manuscript received November 23, 1992. Revisions received May 17, 1993 and October 6, 1993. Accepted by Penelope M. Sanderson and Carolanne Fisher. Final manuscript received June 28, 1994. – Editor

REFERENCES


