AL/HR-TR-1993-0072

ARMSTRONG LABORATORY

RESEARCH, DEVELOPMENT, TRAINING, AND EVALUATION (RDTE) SUPPORT DELIVERY ORDER 1: COMPUTATIONAL COGNITIVE MODELS

Stephen E. Deutsch
Eva Hudlicka
Marilyn J. Adams
Carl E. Feehrer

BOLT BERANEK AND NEWMAN, INC.
10 MOULTON STREET
CAMBRIDGE, MA 02138

HUMAN RESOURCES DIRECTORATE
LOGISTICS RESEARCH DIVISION
2698 G STREET
WRIGHT-PATTERSON AFB, OH 45433-7604

SEPTEMBER 1993

FINAL TECHNICAL REPORT FOR PERIOD MARCH 1992 TO MAY 1993

Approved for public release; distribution is unlimited

AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation, or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Public Affairs Office has reviewed this report and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This report has been reviewed and is approved for publication.

Michael J. Young

MICHAEL J. YOUNG
Contract Monitor

BERTRAM W. CREAM, Chief
Logistics Research Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
Sep 1993

3. REPORT TYPE AND DATES COVERED
Final - March 1992 to April 1993

4. TITLE AND SUBTITLE
Research, Development, Training, and Evaluation (RDTE) Support
Delivery Order 1: Computational Cognitive Models

5. FUNDING NUMBERS
C - F33615-91-D-0009
PE - 62205F
PR - 1710
TA - 00
WU - 60

6. AUTHOR(S)
Stephen E. Deutsch
Marilyn J. Adams
Eva Hutlicka
Carl E. Feehrer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Bolt, Beranek and Newman, Inc.
10 Moulton Street
Cambridge, MA 02138

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Armstrong Laboratory
Human Resources Directorate
Logistics Research Division
2698 G Street
Wright-Patterson AFB, OH 45433-7605

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AL/HR-TR-1993-0072

11. SUPPLEMENTARY NOTES
Armstrong Laboratory Technical Monitor: Michael J. Young, AL/HRGA, (513) 255-8229

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This document reports on work associated with the three-phase development of a psychologically-based Human Performance Process Model that can be used in the assessment of system performance. The first phase produced a preliminary architecture consisting of two parts: A "mass memory" composed of long term memory structures that supported associative activation, and a set of sensory, perceptual, cognitive and motor agents that communicated via message passing - writing messages to or reading messages from the mass. Phase 2 produced a revised architecture in which the distinction between memory and process disappeared and the resulting structures became active processing elements in a data flow structure. The addition to model development, certain activities associated with enroute Air Traffic Control were also decomposed in order to develop the basis for a scenario within which to prototype, program and test aspects of the revised model. The activities selected provided an opportunity to model single- and multitasking behaviors and performance.

14. SUBJECT TERMS
human operator multitasking
human-oriented programming
human performance

15. NUMBER OF PAGES
86

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	SAR

CONTENTS

FIGURES

Figure Page

TABLES

Table

PREFACE

This report presents a summary of the results of a study aimed at developing a Human Performance Process (HPP) model based on theoretical and empirical findings in psychology and cognitive science. It is the last of three reports delivered under USAF Contract No. F33615-91-D-009. The first two, BBN Interim Report No. 7797 and BBN Interim Report No. 7752, present additional details on the development effort conducted under the Air Force's Research Development, Training, and Evaluation (RDT&E) Support program dealing with computational cognitive models.

The work reported herein could not have been accomplished without the enthusiastic support and vision of Mr. Michael Young, the Air Force contract monitor, and the technical contributions of Dr. Richard Pew and Mr. William Ferguson of BBN.

SUMMARY

This document reports on work associated with the three-phase development of a psychologically-based Human Performance Process Model that can be used in the assessment of system performance. The first phase produced a preliminary architecture consisting of two parts: A "mass memory" composed of long term memory structures that supported associative activation, and a set of sensory, perceptual, cognitive and motor agents that communicated via message passing--writing messages to or reading messages from the mass. Phase 2 produced a revised architecture in which the distinction between memory and process disappeared and the resulting structures became active processing elements in a data flow structure. In addition to model development, certain activities associated with en route Air Traffic Control were also decomposed in order to develop the basis for a scenario within which to prototype, program and test aspects of the revised model. The activities selected provided an opportunity to model single- and multitasking behaviors and performance differences between novices and experts--two dimensions thought to be important in a review of model adequacy. Limited aspects of the model were prototyped during Phase 3 using a Lisp-based procedure language.

RESEARCH, DEVELOPMENT, TRAINING, AND EVALUATION (RDT&E) SUPPORT

1.0 INTRODUCTION

Tasks performed by human operators of complex systems differ significantly with respect to the demands they place on sensory, perceptual, cognitive, and motor processes. Some tasks draw heavily on an operator's ability to make fine visual and/or auditory discriminations among "noisy" signals and, having accomplished those, to recall from memory one or more well-defined rules relating to proper disposition of whatever is or is not identified. Others tax an operator's perceptual capabilities only marginally, displaying their outputs in highly discriminable form. Many of these, however, make significant demands on abilities to interpret indications in light of "past experience" and to utilize whatever rules might exist in novel ways to predict future events and identify appropriate actions. Although it has always been difficult to incorporate convincing representations of most psychological processes in simulation models that are, at the same time, both descriptive and predictive, cognitive processes involved in these activities have proven to be the most refractory.

From the perspectives of psychological theory, human performance modeling, and system operability analysis, perhaps the most interesting tasks are ones that:

- make major demands upon cognitive and memory resources during concurrent execution of subtasks,
- must be accomplished at a rapid tempo with minimum error,
- reveal during their execution what may be important differences in the perceptual and cognitive operations of novices and experts, and
- constitute potential sources of "leverage" during efforts to improve system performance through redesign of operator equipment, procedures, or training.

The goal of RDTE, Delivery Order #1, is to address these diverse interests through a comprehensive research and development program that accomplishes the following three objectives.

1. Identify salient human performance perceptual and cognitive components and attributes.
2. Incorporate these components within a new, psychologically based, Human Performance Process (HPP) model that can be implemented on a computer.

3. Test, validate and refine key aspects of the model, preparatory to full implementation during a subsequent delivery order.

Following the program plan, these objectives were pursued over the course of three phases. The first of these was completed in May 1992, the second in November 1992, and the third in March 1993.

1.1 Organization of the Report

Section 2 is a review of the progress made in each of the three phases. Section 3 presents a summary of the decomposition and analysis of the Air Traffic Control (ATC) domain (which was chosen as a focal point for this effort), identifies several psychological attributes and behaviors of (human) controllers that have been incorporated into the design of an architecture called Chaos, and presents a review of the architecture.

Section 4 presents the aspects of automaticity and multitasking as they relate to the ATC problem. The revised set of functional entities (demons) and psychological architecture are also described here. Section 5 reviews additional concepts related to automaticity and multitasking, presents the conceptual architecture, and outlines how the model supports simulation of the psychological behaviors described in Section 4. Section 5 also includes a set of "Spec Sheets" that provides examples of the types of input, processing, and output associated with ATC tasks to be incorporated in future working versions.

Section 6 is an overview of programming conventions, particularly those related to the Lisp-based languages, that are expected to provide mechanisms for implementation. The results obtained in a limited effort to prototype aspects of the model using certain of these conventions during the final phase of the project are discussed in Section 7.

Section 8 contains references to prior works cited in the body of the report.

2.0 REVIEW OF THE PROJECT

2.1 Summary of Progress During Phase 1

Early in the project, an approach was developed to achieve mappings among elements deemed essential in the formulation of an HPP model that would be adequately grounded in psychological and cognitive science theory and that would be useful in studies of operability. This approach, conceived by an interdisciplinary team composed of human performance modelers, cognitive science/artificial intelligence (AI) researchers, and psychologists, called for identification of the following elements:

- one or more operational contexts in which human performance plays a major role in mission success;
- a set of behavioral building blocks and relationships that could help (a) to provide foci for reviews of the psychological and cognitive science literatures pertinent to human performance modeling, and (b) to describe and formulate a testable model architecture;
- a body of theory that could aid in defining the substance of the building blocks (i.e., that "operationalized" them); and
- a body of important empirical findings that would aid in (a) identifying parameters that should be included in the model and that should be accessible to the user, and (b) identifying aspects of the model that would require actual prototyping in order to judge suitability and validity as a research vehicle.

2.1.1 Operational Contexts

Three different operational command and control contexts that place high demands on perceptual and cognitive functioning were surveyed: (1) the Modular Control Equipment (MCE) environment of a Control and Reporting Center (CRC) previously identified in connection with the development of Armstrong Laboratory's Automation Impacts Research Testbed (AIRT) simulator (Corker & Cramer, 1989a; 1989b; 1991), (b) a shore-based naval underwater surveillance system reminiscent of the Sound Surveillance System (SOSUS), and (c) a scout attack helicopter mission.

Although the balances between perceptual and cognitive loads imposed on operators in these three environments differed (with MCE and SOSUS making stronger demands on perceptual resources than on cognitive resources and the helicopter mission making nearly equal demands on both), the contexts, in the aggregate, highlighted requirements for addressing, in the formulation of the model, such functions as retrieval of procedures and data from memory, planning and scheduling, attention, decision-making, and (performance) error generation. In addition, they pointed to two not-unrelated dimensions of performance that are of major interest in areas of theory, human performance modeling, training, and operability assessment:

- the ability to perform several tasks at the same time (multitasking), and
- the differences in "style" and output that can be observed in the performance of an expert as opposed to that of a novice.

Thus, to be interesting theoretically and to have adequate potential as a tool for operability assessment, the model must somehow be able to simulate these behaviors and allow the user to "experiment" with them.

2.1.2 Behavioral "Building Blocks"

Partly to offset the effects of differing (technical) vocabularies among team members and partly to establish a wide focus for the literature review, a set of entities was defined that collectively expresses the functional behaviors required in a model. Each entity was cast as a "demon" that, given a set of triggering conditions, performed specific perceptual or cognitive function(s) and could, depending upon circumstances, commit errors characteristic of human cognitive processing. The set agreed upon during Phase 1 and the goal-directed behavior associated with each are as follows.

Scanner: Assures that no "relevant" phenomena go undetected.

Recognizer: Filters input from the Scanner to facilitate preliminary categorization and interpretation.

Predictor: Anticipates future states of attentional phenomena in order to guide the Recognizer and Responder.

Conceptualizer: Detects, evaluates, and reconciles deviations between expected and actual information in order to maintain cognitive coherence.

Responder: Interacts with and intervenes in situations as specified by the Conceptualizer.

2.1.3 Phase 1 Chaos

The Phase 1 implementation architecture views the demons as being grouped around a central memory structure ("blackboard") onto which they post "messages" relating to the outputs of their processing and from which they retrieve "messages" requiring their (processing) attention. The architecture was given the name "Chaos" to underscore the dynamics and "emergent" properties it is anticipated to exhibit.

Chaos consists of two primary components (Figure 1):

- a collection of interlinked long-term memory structures (schemata) that support associative activation (referred to as a "blackboard" or a "memory mass"), and
- a collection of processes/demons that operate on the data contained in the memory structure and communicated either directly (via inter-process communication channels) or indirectly (by reading from or writing onto the structures that compose the blackboard/memory mass).

At any given time, the schemata composing the memory blackboard/memory mass have varying degrees of activation. This activation changes in response to the associative activation process within the memory mass and in response to activation by the processes/demons that "hover" around the blackboard and either "read" or "write" data into the schemata.

'The processes contain their own "private" stores of data and base their processing, in part, on such stores. The processes could be made to implement a variety of computational formalisms, including those associated with rule-based, model-based, and connectionist approaches, thus enabling the overall architecture to embody a variety of computational models.

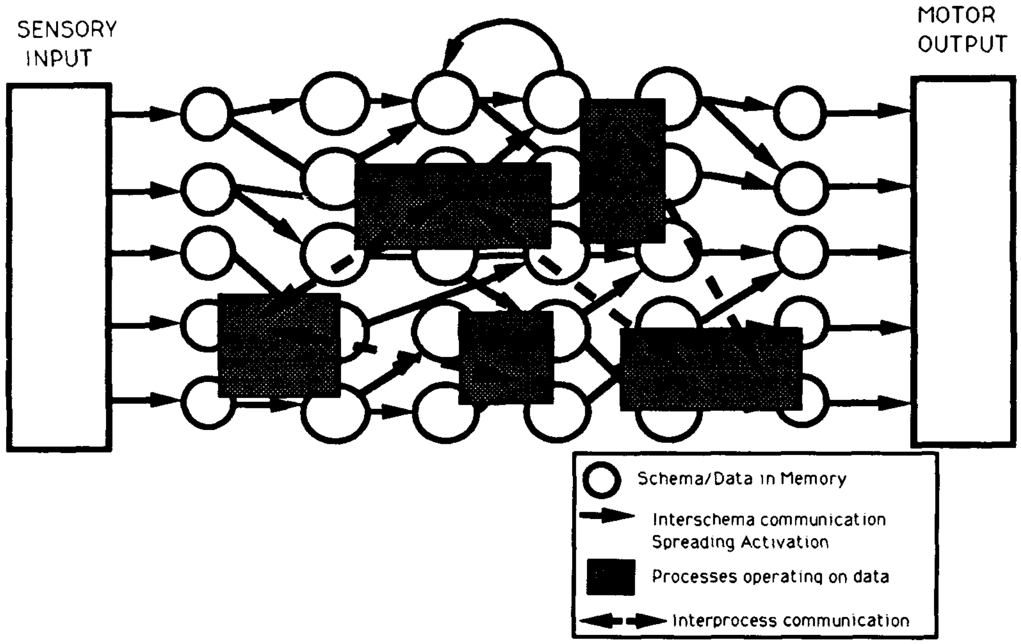

SENSORY INPUT
MOTOR OUTPUT
Schema/Data in Memory
Interschema communication Spreading Activation
Processes operating on data
Interprocess communication

Figure 1
Chaos: Phase 1

Parallelism occurs in both the associative activation among the memory mass schemata and within the processes, which are able to operate on their data in parallel. Sequentiality is induced because some data necessary for a particular process might not be available until some other process produces it.

Thus, although the memory structure in Phase 1 Chaos has some degree of data processing associated with it (i.e., the interschema activation), there remains a distinction between data and process in the sense that the bulk of the data transformation necessary for cognition is performed by the processes associated with the blackboard structure.

2.1.4 Review of Theory

A large body of classical psychological literature and contemporary cognitive science literature was reviewed during Phase 1. This review was an effort to identify existing theoretical positions in general areas of consciousness, attention, automaticity, and human error in order to lend substance to the functional view of the processes mentioned in Subsection 2.1.2 above. Special emphasis was placed on the review of models such as Adaptive Character of Thought (ACT) and Soar, which address behavior at global levels and, in that respect, present alternatives to the Chaos formulation.

2.1.5 Empirical Findings

Although the final selection of parameters depended on the precise form of the model that was finally implemented, a preliminary effort was made to identify broad categories of "switches" that would be desirable in an HPP to make it useful to researchers. If, for example, one were to theorize that short-term memory plays an important role in the accuracy of performance on a given task, then one would want to implement the model such that the memory size and/or persistence of its contents were parameterized and that its use was under the direct or indirect control of its user.

Obviously, much of the "data" required in a search of this sort is already available in the literature: All theories and models are built on some set of observations. However, attempts were made to add to this normal accumulation by reviewing sources that are less focussed on theory and more focussed on empirical aspects of performance, such as the data base which supports Generic Operator Modeling System (GOMS) modeling and data bases contained in human engineering handbooks written for equipment designers.

2.2 Summary of Progress During Phase 2

2.2.1 Modifications to the Chaos Formulation

The Chaos architecture sketched out in Phase 1 was modified during Phase 2. The most important difference between the two versions is that the distinction between data and processes narrowed from Phase 1 to Phase 2, and the processes/demons of Phase 1 are incorporated into the current memory space of Phase 2. In Phase 1, the memory blackboard/memory mass consists of schemata containing long-term memory. A temporarily activated subset of these schemata is defined as the short-term memory. In the Phase 2 version, the schemata that compose the long-term store are combined with the demons of Phase 1. The resulting structures are *active processing elements*. Most of the discussion about how the schemata are activated and the consequences of such activation continue to apply, but the units in Phase 2 are *active* processes/procedures rather than the passive schema structures in Phase 1.

The major difference between Phase 1 and Phase 2 versions is the convergence of data/knowledge structures and processing into a notion of "memory as process." The distinct processes of Phase 1 have been absorbed into the "network of schemata" that constitute the "mind," and the distinction between memory and processing has disappeared (Figure 2). Each retrieval and recognition is a reinterpretation of the existing stored information that alters the parameters of the network of procedures. In other words, what were referred to as memory schemata in Phase 1 are viewed as processes in Phase 2, and, in terms of the formalisms required to implement them, as agents or procedures. These procedures are organized into a network through which signals representing both external (sensory) data and internally generated data flowed. The resulting structure is a type of data-flow network with many feedback loops that channel partial results back toward the earlier sensory-data processing units and thus enable proactive, goal-directed behavior.

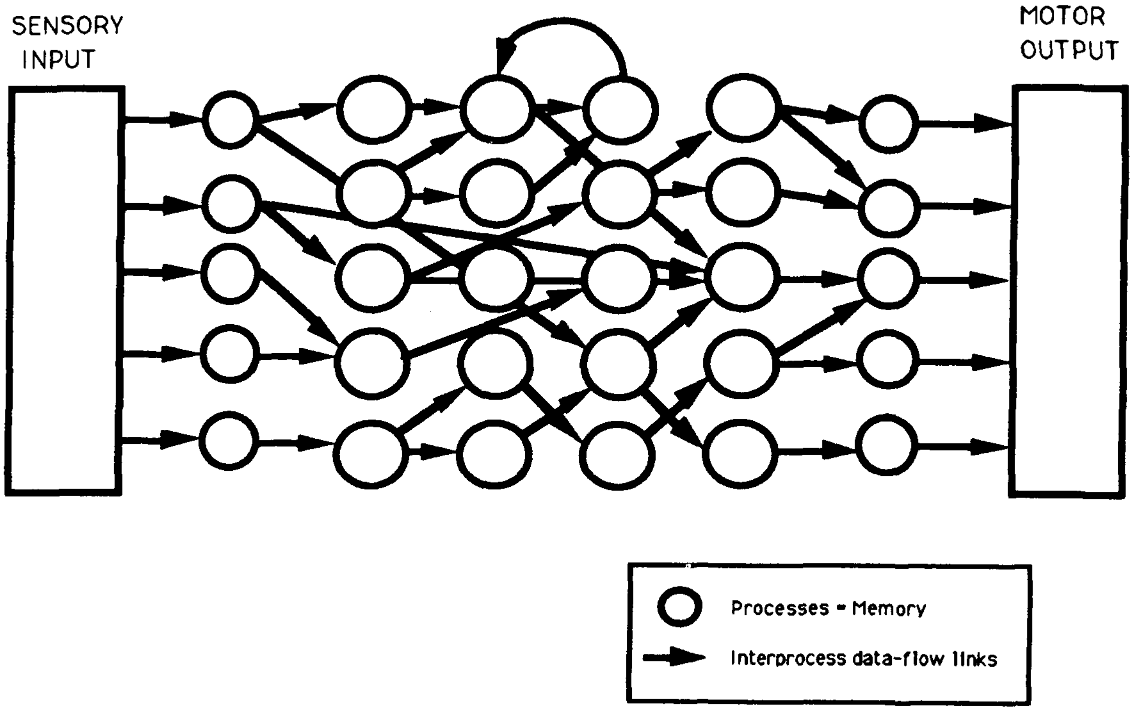


Figure 2
Chaos: Phase 2

The notion of a schema within this paradigm is not so much that of a static, distinct structure that is always recalled or used as an identical entity, but rather that of a set of procedures that are activated at any one time in response to a particular set of external and internal stimuli. As in the Phase 1 version, for which the blackboard schemata have varying degrees of activation, the procedures that compose memory in the Phase 2 version have varying, dynamic degrees of activation. Within this conceptualization, a schema is a collection of such highly activated procedures. As the internal and external stimuli change, so does the level of activation of these procedures and the set of schemata representing the current processing context. This view is consistent with recent conceptualizations of the notion of a schema in the literature and was inspired largely by connectionist models, which regard a schema to be a dynamic configuration of active processing elements (Rumelhart et al., 1988).

These procedures contain both declarative and procedural knowledge, and there is no "special structure" for different types of memories, as has been proposed in the past in the psychological literature and as modeled in some previous architectures of cognition (e.g., Anderson, 1983).

As in Phase 1, the individual procedures can have private stores that enable them to have internal states, thereby increasing the complexity of the data transformation possible. Also, as in Phase 1, the architecture combines parallel and sequential processing. The sequentiality is imposed on the processing by the sequential flow of data through the network of procedures, and parallelism becomes possible both because of parallel signal pathways and because subnets of procedures in various parts of the overall network can maintain a degree of processing activity.

The conception that evolved during Phase 2 remained essentially unchanged in Phase 3. The essential differences between the two versions are summarized in Table 1.

Table 1. Summary of Chaos Models in Phase 1 and Phases 2, 3

	Phase 1	Phase 2, 3
Parallelism	YES	YES
Spreading activation	YES (marker passing)	YES (data flow, activation)
Distinct parts of memory have levels of activation	YES (schemata)	YES (procedures)
Processes can have private data	YES	YES
Redundant processing of data	YES	YES
Centralized global blackboard	YES	NO
Combination of parallel/ sequential processing	YES	YES

2.2.2 Selection of Operating Domain

The effort begun in Phase 1 to identify an operational task domain that could aid in focussing the theoretical work concluded in Phase 2 with the selection of (en route) ATC. This selection was based on the following considerations.

- ATC presents an environment in which the performances of human operators associated with the use of diffɜrent display, control, and procedural alternatives can usefully be evaluateḋ. Hence, it is heuristic in suggesting the types of perceptual and cognitive behaviors the researcher might want to "audit" in an operability testbed.
- ATC places considerable demands on both perceptual and cognitive processing.
- Field studies of controller behavior exist that could be useful in validating a model once it has been implemented. In addition, practitioners who can aid in the assessment of model outputs are readily available.
- Domain data are available that can facilitate knowledge engineering during model implementation.
- Inputs to and outputs from control tasks are well-defined, and criteria for judging the adequacy of the controller's solution of a problem exist to aid in performance measurement.

Once the task domains were selected, attention turned to decomposition and analysis of controller tasks that could, in Phase 3, provide a basis for scenario and model prototyping.

2.2.3 Review of Literature

The review of literature that could provide a basis for development of both computational and implementational aspects of the model was broadened during Phase 2 to include the following:

- distributed AI architectures as examples of symbolic-level parallel models,
- high-level connectionist models—techniques and pros/cons,
- integrated symbolic ɩ ıd connectionist models—techniques and pros/cons,
- literature describing errorful processing at the cognitive, neurological, and neurochemical levels,
- psychological literature on spatial maps and AI work in direct representations, and
- literature on consciousness and attention.

2.3 Summary of Progress in Phase 3

With the essential aspects of the architecture in place as a result of Phase 2 activity and in accord with the program plan, nearly all effort in Phase 3 was devoted to the operational decomposition of selected ATC tasks and to the rapid prototypinɾ of limited aspects of the model.

2.3.1 Domain and Procedure Prototyping

Two products were developed to aid Phase 3 prototyping and to provide a model for additional analyses to be made during a fuller implementation of the model in a later delivery order. The first, a "Spec Sheet," lists the essential inputs, processing, and outputs associated with the handoff of an aircraft from an en route controller in one sector to an en route controller in another. In addition, it provides characterizations of the kinds of errors that might be observed in the performance of a novice controller.

The second product presents a time line of the procedures identified with a handoff. This time line is segmented to depict input, processing, and output at selected intervals during the handoff procedure. It also highlights the multitasking aspects of the controller's performance.

2.3.2 Prototyping

Because the resources available for prototyping during Phase 3 were limited, an effort was made to experiment with key features of the psychological framework with which little experience had been accumulated during earlier Bolt Beranek and Newman, Inc. (BBN) efforts at model construction. Three areas were chosen for the prototyping: (1) conflict resolution among multitasking agents; (2) the matching of task selections to agent performance levels, as might be associated, for example, in a comparison of novice with expert control behaviors; and (3) data flow and reentrant net aspects of the Chaos implementation.

The prototyping activity has resulted in a miniature view of the interaction of two controllers and three aircraft as the latter maneuver in two en route sectors modeled after Dallas and Houston. As described in Section 7, the current output of the prototype is a script that describes the interactions and activities of the agents, a simulated radar display of the airspace that aids in understanding the control problem, and a time line that depicts the sequence of controller and aircraft responses.

3.0 THE ATC TASK DOMAIN

This section summarizes the ATC domain in sufficient detail to enable the presentation of decompositions of several tasks and the illustration of how these might be implemented within the proposed architecture. Additional information relating to inputs, processing, and outputs is continued in Section 4.

3.1 ATC Task: Summary

All airspace, except immediate airport traffic areas, is divided into three-dimensional blocks of airspace called "sectors." Control of these sectors is performed at regional Air Route Traffic

Control Centers (ARTCCs). En route ATC typically involves one air traffic controller managing one sector of the traffic space. In very busy sectors, there may be two controllers, one for data (i.e., flight plan) control and one for radar control.

The major task of the en route controller is to ensure separation between (i.e., prevent collisions among) aircraft. There are rules for aircraft separation and for conflict avoidance once a potential conflict is observed. The behavior of the en route controller is thus highly standardized. Controllers are trained for several sectors, and their behavior is optimized with respect to these—although their training is generic enough so that they could take over any sector, presumably with some loss of efficiency.

Control tasks are typically well-defined. However, controllers must be ready to handle unexpected emergencies because of accidents, weather, or failures onboard the aircraft. The task thus provides for complex decision-making, both routine and novel.

En-route controllers obtain and share information that is made available through several media:

- radar screens displaying aircraft, information about each aircraft, and air routes;
- flight strip information associated with each aircraft that filed a flight plan involving the sector;
- telephone links to controllers in neighboring sectors; and
- radio links to aircraft.

Controllers communicate primarily with pilots within their sector, except for "handoffs," which they also communicate with neighboring controllers. (If there are two controllers per sector, these two also communicate.)

3.2 Controller Tasks

When assuming control of a sector and for the duration of the shift, a controller must successfully perform two important cognitive tasks: (1) develop a mental model of potential conflicts within the assigned airspace, and (2) when the potential for conflict(s) is detected, devise a strategy for avoiding them.

The controller's mental model of potential conflicts must be derived from:

- information contained on flight strips and radar displays;
- knowledge of standard "choke" points or "traps" in the airspace where changes in altitude are typically requested because of air corridor geometry, terrain, weather, or the desire to land at or take off from nearby airports; and
- the "intelligence" provided by the departing controller at the beginning of a shift.

When the potential for one or more conflicts is identified (i.e., when one or more aircraft are "traffic for each other") the controller must devise a strategy for avoiding them. This strategy must be within a set of rules and criteria established by the Federal Aviation Administration (FAA) for safe and responsive control of aircraft within the sector.

There is evidence that controllers follow a fixed script for deciding which aircraft are traffic for one another (Means et al., 1988). This script calls for checking the following in turn: Are aircraft at the same altitude? Do they have same source/sink airport? Do they cross over the same fix? Do they fly along the same airway? Are they at the correct altitude for their direction of travel?

Controllers also seem to follow a fixed script when deciding which conflict to solve first (urgency, then ease of solution) and which resolution strategy to apply (change altitude, airspeed, vector).

Two typical tasks that a controller performs are (1) receiving a handoff from another controller and (2) resolving a potential collision. Each task is analyzed with respect to the types of inputs to relevant sensory modalities, the types of outputs along relevant appropriate motor channels, and the types of cognitive processing underlying the mapping of the inputs onto the outputs, along with any changes in internal models/parameters.

Note that both tasks incorporate substantial amounts of "preprogramming." This aids in establishing the relevant structures in memory. These structures consist of internal representations[1] of the domain, the task and individual task steps, as well as control and metacognitive processes.

[1] Use of the term "internal representations" is not intended to imply static structures but rather patterns of activation among relevant procedures in memory.

3.2.1 Receive an Aircraft from Another Sector

This task is typically routine, involving coordination of sensory and motor processing, dialogue with a neighboring controller and a pilot, minimal updating of the internal representation, and, typically, minimal decision making. Because of its relative simplicity, this is a good "starter" task. A detailed description of some of its constituent procedures is included in Section 5.

The purpose of the handoff task is to transfer control of an aircraft from one sector to another—that is, to transfer control from one controller to another. Both controllers are alerted to the fact that the aircraft is approaching the boundary of a sector by the flashing of the aircraft icon on the radar screen. This signals the current controller to telephone the neighboring sector controller and request a handoff, which is typically accepted. The controller then contacts the pilot by radio and announces that the handoff has been accepted. From the receiving controller's point of view, the handoff task involves the following steps.

- Receive a phone call from a neighboring controller and accept the handoff.
- Shortly after, receive a radio call from the aircraft pilot, announcing flight information (e.g., "Chicago center, this is American 111 with you at <frequency>").
- Confirm aircraft's next waypoint or provide pilot with appropriate changes to the flight plan (e.g., "American 111 cleared direct to <next waypoint>").
- Turn off flashing icon to indicate a completed handoff.

A breakdown of this task, indicating the input to sensory modalities and the output from motor modalities and speech, is shown in Figure 3.

3.2.2 Detect and Resolve Potential Conflict

The major task of controllers is to monitor their sectors for potential conflicts (i.e., collisions or violated separation standards) and to resolve these by changing the altitude, airspeed, or vector of one or more of the aircraft. Specific scenarios during which there is a potential for collision are:

- same airway/same altitude: head-on collision;
- same altitude: converging aircraft; and
- two aircraft both of which are climbing (or descending) in the same airspace but at different speeds (the faster one will eventually overtake the slower one).

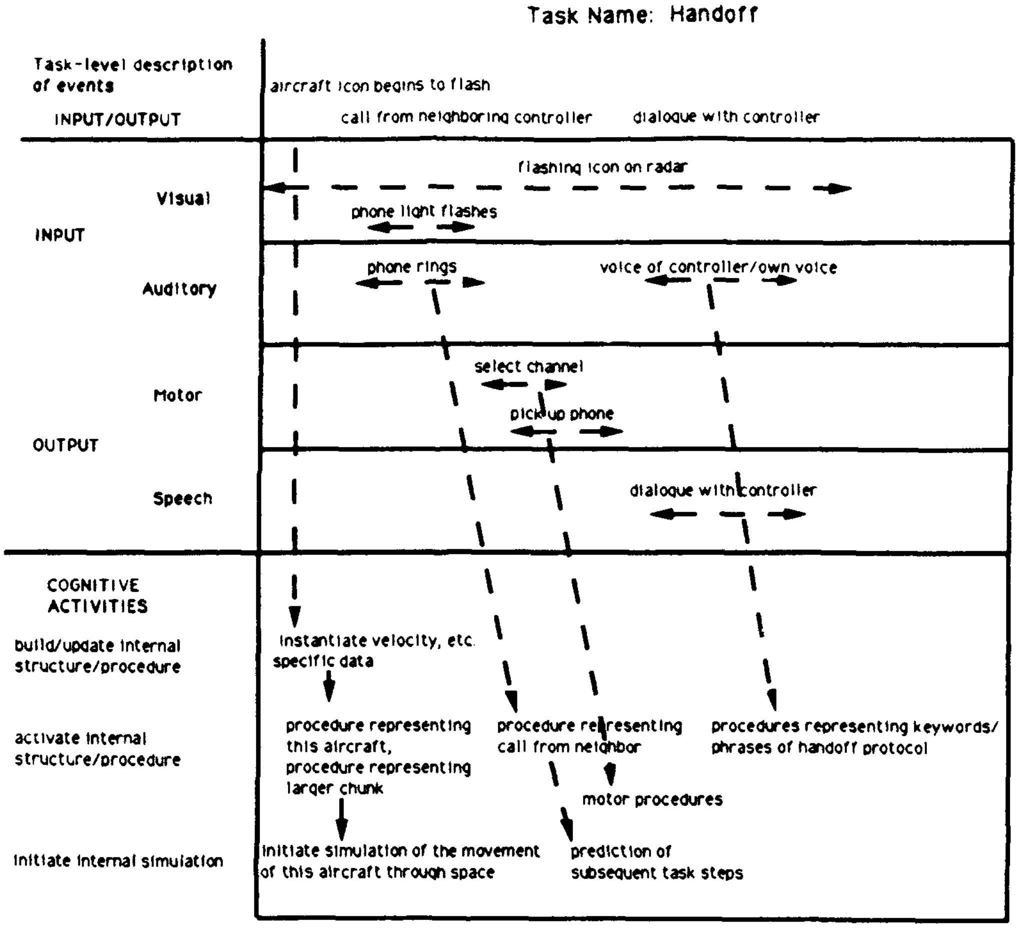

Task Name: Handoff
Task-level description of events
INPUT/OUTPUT
aircraft icon begins to flash
call from neighboring controller
dialogue with controller
INPUT
Visual
flashing icon on radar
phone light flashes
Auditory
phone rings
voice of controller/own voice
OUTPUT
Motor
select channel
pick up phone
Speech
dialogue with controller
COGNITIVE ACTIVITIES
build/update internal structure/procedure
instantiate velocity, etc. specific data
activate internal structure/procedure
procedure representing this aircraft, procedure representing larger chunk
procedure representing call from neighbor
procedures representing keywords/ phrases of handoff protocol
motor procedures
initiate internal simulation
initiate simulation of the movement of this aircraft through space
prediction of subsequent task steps

Figure 3
Timeline Showing Inputs, Outputs, and Cognitive Activities Associated with ATC Handoff Task

The difficult aspect of this task is that potential collisions may not be immediately obvious; thus, the controller must project the current positions forward in time. Furthermore, the solution to one conflict may cause a potential conflict for another group of aircraft. When making a decision, the controller must therefore consider present and future traffic patterns. In effect, each solution involves a relaxation across the set of traffic patterns to ensure that no conflict is caused. In addition to avoiding collisions, the controller must find a solution that will have minimal impact on the flight plan, cause little or no delay, and maintain maximal fuel efficiency. The controller can redirect the aircraft by issuing one or more of the following commands.

- via speed control: **slow down/speed up**
- via altitude control: **climb/descend**
- vector (detour) aircraft: **turn**

Controllers are highly trained for their specific sectors and traffic in each sector tends to form into patterns. Therefore, experienced controllers have well-defined procedures for handling collision patterns typical for their sectors and much of their behavior appears to be relatively automated.

3.2.3 ATC Task Domain Input/Output

To demonstrate the complexity and richness of the ATC task domain, a summary of the types of sensory inputs and motor (including speech) outputs of the controllers' tasks is presented below.

Input (modality; source; type)

Auditory

- Call from pilot (radio)
  - Indication that pilot is in sector: "Chicago center, this is American <xx> with you at <frequency> cleared direct to <next waypoint>
  - Acknowledgment of command
  - Request for rerouting (altitude/speed/vector change)
  - Clearance requests
  - Emergency
  - Request for weather
  - <other - argument? problem?>
- Call from neighboring controller (phone)
  - Request for handoff
- Response from neighboring controller (phone) after own call, indicating acceptance of the handoff

Visual

- Data from radar screen
  - Aircraft blip on radar screen (paired/unpaired) (flashing/solid)
  - Aircraft identification tags
  - Changes from one state to another (from paired to unpaired)
  - Site icons: beacons, airports
  - Airways
  - Weather information
  - Sector boundaries
- Data from flight strip

Aircraft

Flight Paths

Prerouted

Annotated later

Data from CRT works

Data "chunked" as potential traffic in a given time frame

Data "chunked" as potential traffic in the future

Output (data type and destination of data)

Voice

To pilot: acknowledgment, weather information, emergency information, maneuver instructions (change in heading/altitude/speed)

To neighboring controller: handoff/acceptance protocol

Motor

Put away/get/move flight

Place track ball cursor on displayed symbol: interrogate/modify display

Type on keyboard: modify formal flight plan parameters

Mark flight strip

Select phone line/pick up receiver

3.3 "Data" for a Theory of Novice/Expert Differences

A recent and very thorough study (Means et al., 1988) of controllers provides a number of important insights into the behavior of operators (particularly experts). The findings of this study include the following.

1. Controllers partition the airspace into regions of "tension" and "nontension" based on similarity of altitude, proximity, and headings of aircraft.

2. There is a significantly reduced level of recall for aircraft not actively controlled.

3. A cognitively maintained priority schedule is formulated for dealing with subsets of potentially conflicting aircraft ("First I'll deal with these, then with those, "etc.),

4. Site-specific factors (e.g., other nearby airports, common fixes, etc.) are critical in the control strategy. These include known, consistent regions of potential conflict that are largely independent of the traffic distribution on a particular day and that must be factored into the controller's plan. They are regions of "sensitivity" that even an expert must learn about when moving to a new center.

5. The constraints imposed by fixes and airways, and the relatively constant traffic patterns present a means for rapid assimilation of the airspace "picture."

A primary difference between the novice and the expert is the ease with which each identifies areas of tension in the airspace. Although immediate contenders might be screened out quickly because of significant (two-dimensional) geographical separation, the novice must painstakingly consider all possible pairs of aircraft that are apparent contenders. This takes time and may tax the novice's memory capacity.

An expert, on the other hand, accomplishes the task much more quickly and, perhaps, not by making pairwise comparisons. The "products" of the expert's analysis might also present less of a memory load. The difference in efficiency and load may be observed across a set of intellectual tasks, including the following.

- generating a priority structure (triage) for dealing with potential conflicts; and
- identifying strategies (combinations of plans and overt control actions) that satisfy five major considerations:
  - Does the strategy create additional conflicts?
  - Does the strategy interact with constraints associated with an aircraft?
  - Can the strategy be accomplished by the aircraft?
  - Is the strategy in agreement with experience?
  - How long will the "fix" hold?

There is psychological evidence (Means et al., 1988) that controllers build a three-dimensional mental model of the airspace and perform "chunking" as they become more experienced.

Chunks consist of groups of aircraft whose trajectories potentially conflict. Note that chunking incorporates the goal of the controller's task: aircraft separation. A chunk integrates knowledge about the task and the specific data available at the time. It includes both three-dimensional mental models and internal simulations of the aircraft movement in the near future.

The differences in efficiency across the set of intellectual tasks are likely to result in at least the following (nonindependent) characteristics of the expert's performance:

- a quicker plan;
- a better plan;
- a plan that can be modified more easily;

- a plan that can be modified more quickly; and
- a major savings in total planning time.

If the coherency of the plan is better and the generation time is less, the controller is likely to have more time to:

- monitor unfolding events and thereby test expectations against actual performance more frequently;
- acknowledge (handshake) pilot-initiated communications with shorter delays, including:
  - check-ins when arriving in a sector;
  - requests for rerouting; and
  - requests for information; and
- initiate control actions with shorter delays.
  - routine requests for rerouting; and
  - satisfying emergency requests.

The explicit gains in control "quality" may be accompanied by at least one implicit personal gain: an enhanced rate of accumulation of additional "experience" (i.e., "you just get to do it more often"). In addition, the potential for (relatively) higher rates of error among novices may result from the many more computational cycles required by pairwise comparisons and the higher workload created by less efficient strategies.

4.0 PSYCHOLOGICAL FRAMEWORK

4.1 Methodology for Establishing the Psychological Framework

In developing a psychological framework, it is important to identify and quantify the sources of human performance in a manner that will make it possible to develop a computational model of these behaviors (i.e., an HPP model). As a starting point for the effort, the literature on experimental psychology. was chosen. This literature reinforces the basic premise that human performance capability is rooted in the ability to manage the “simultaneous” execution of multiple tasks. Immediate concerns then focus on the structure of single-task performance and on automaticity as an important aspect of multiple-task performance. The mix of these attributes that shape both novice and skilled performance is important, as are the sources of human error at each skill level.

While the experimental psychology literature constituted the primary reference domain, the neuroscience and computer science literatures were also reviewed (Figure 4 identifies some of the pertinent areas of this literature). Neuroscience and, in particular, the work of Gerald Edelman (1987; 1989), proved especially supportive in the development of the framework. Edelman's reentrant and global maps have added a neuroscience basis for the procedure network that is described later in this paper as a model supporting a range of important ideas on multitask performance and automaticity. Computer science has played a lesser but important role in developing the psychological framework. It is reflected primarily in the representations of goals, plans, a computational view of procedure execution, and the relevance of data-flow computation to the parallel execution of procedures in a network. The activation of procedures by data arriving from multiple sources is very much in the spirit of data-flow computation. Since the effort is to elaborate a psychological framework that will support the development of a computational HPP model, computer science issues supporting the next important step of implementation, have also been addressed.

Experimental Psychology	*Neuroscience*	*Computation*
Single-Task Resources	Reentrant Maps	Objects and Message Passing
Automaticity	Global Maps	Data Flow
Multitasking	Parallelism	Parallelism
Goals, Plans, and Intentions	Activation/Inhibition	Goals, Plans, and Procedures
Connectionism	Connectionism	Connectionism
Memory and Comprehension	Degeneracy	

Figure 4
Resources for Developing the Psychological Framework

As a last point regarding methodology, several categories are repeated in more than one column of Figure 4, highlighting the fact that neither the terms nor the theory and research they subtend are the sole property of a single domain. An important part of the process of developing the psychological framework has been the attention of the team members to the task of explaining to their colleagues the unique character their discipline has brought to these terms.

4.2 Theoretical Orientation

In the Phase 1 report, five generic psychological processes or functions required by the human simulator were enumerated. These functions were labeled: Scanner, Recognizer, Conceptualizer, Predictor, and Responder. While these remain functions that the simulator must support, their interrelations within the cognitive architecture have been significantly reconceived. This redesign is more compatible with recent research, even while it offers cleaner and more sophisticated potential for mimicking human cognition with a machine.

In the current formulation of the psychological architecture, the Memory Module effectively comprises a large number of process or event memories of varying degrees of complexity and works with a variety of data. These memories communicate with one another via a complex of excitatory and inhibitory interconnections. Some of these bundles of connections are "strong," and a "weak" signal will be able to activate them; other connections are "weak," and will require a "strong" signal for the message to pass through. The strength of any given complex of connections reflects its frequency of use as well as its current level of activation, such that the strength of its response to a message is a joint function of learning, expectation or priming, and the compatibility of the message with its structure.

Complementing the Memory Module, we propose a Cognitive Model. Within the Cognitive Module, the activities and interests of oneself as well as the impinging world (including other people) are represented in terms of means-ends frames. This is consistent with Kant's (1781/1952) arguments that neither superficial similarities nor contiguities in space and time are enough to support rationality in an associative model. That is, a person's ability to sort and order experiences of the phenomenal world depend additionally on an understanding of causality. The knowledge structures of the Cognitive Module are envisioned to consist, in part, of goals with plan structures similar to those developed across the cognitive science literature (e.g., Allen & Perrault, 1980; Barsalou & Sewell, 1985; Bower *et al.*, 1979; Geddes, 1989; Newell & Simon, 1963; Schank & Abelson, 1977; Thibadeau, 1986).

The work of the system is carried out through the interarticulation of the Memory and Cognitive Modules in a variety of ways. Memory messages trigger means-ends frames of the Cognitive Module. Unresolved means-ends frames seek out relevant data from the Memory Module, selectively activating prior knowledge or priming the system for relevant messages from the environment. In addition, with particularization and resolution, the frames direct action and response by invoking and, hence, coordinating and monitoring the effects of the corresponding action knowledge in the Memory Module. Thus, it is the transactions between the Cognitive and

Memory Modules that govern the system's receptivity while weaving its fabric of cognitive coherence and behavioral integrity.

4.2.1 Neuroscience and the Psychological Framework

The potential for the contribution of neuroscience to the development of a psychological framework for HPP models is easily overlooked and, to some extent, distorted by the considerable interest in *artificial* neural networks. The contention here is that one needs to monitor closely the progress in the understanding of the operation of *real* brains. A comprehensive review of recent work in neuroscience is beyond the scope of this effort. However, the contributions of Gerald Edelman are of immediate interest and suggest how recent work in the field of neuroscience has the potential to become relevant to human performance modeling.

4.2.1.1 Edelman on Real Brains. In *Neural Darwinism: The Theory of Neuronal Group Selection* (1987), Edelman discusses the psychological functions of "development, perception (in particular, perceptual categorization), memory, and learning" and how they relate to the brain. In *The Remembered Present: A Biological Theory of Consciousness* (1989), he extends his analysis to consider "perceptual experience—the interaction of memory with the present awareness of the individual animal"; that is, perceptual awareness and conscious experience. On the biological side, Edelman is concerned with proposing explicit neural models to explain how each capability can arise, and relating the emergence of each during evolution and development. When it is to devote attention to the issues of learning in the development of a psychological framework, this aspect of Edelman's development will be more important.

4.2.1.2 Reentrant Maps, Global Maps, and Degeneracy. Neural maps refer to the ordered arrangement and activity of groups of neurons as distinct from single-neuron connections. They are highly and individually variant in their intrinsic connectivity. Changes in the behavior of the network are the result of changes within particular *populations* of synapses. "These structures provide the basis for the formation of large numbers of degenerate neuronal groups in different repertoires linked in ways that permit reentrant signaling" (Edelman, 1987, p. 240) where, in *degenerate* systems, functional elements in a repertoire may perform more than one function and a function may be performed by more than one element (Edelman, 1987, p. 57). Reentry is a basic mechanism suitable for synchronizing the neuronal activity across the mappings at diverse hierarchical levels.

Development of the reentrant maps is a two-step process: Networks are formed as a result of selection during ontogeny with resultant anatomical variance; then functional circuits are formed, once again via selection of populations of synapses. This is an adaptive process that generates behaviors associated with clusters of perceived categories.

Global mappings have a dynamic structure that reaches across reentrant local maps and unmapped regions of the brain to account for the flow from perception to action. Motor activity, an essential input to perceptual categorization, closes the loop.

4.2.1.3 A Neuroscience View of Memory as Active. The traditional view of memory is one of storage (i.e., as the formation and maintenance of a data base). Edelman takes a radically different view—that of memory as *process.* For him, memory is the "*ability* to categorize or generalize associatively" (Edelman's italics, 1987, p. 241). Categorization occurs at the level of a global map and is degenerate. Edelman is well aware of the distinctions between declarative and procedural memory in cautiously addressing his conjectures to procedural memory, but he is also quick to point out that these distinctions may be less than generally assumed. He suggests that there may be a procedural base supporting declarative memory.

Edelman's view of memory as process, conceives perception, categorization, generalization, and memory as closely linked. " . . . memory is a form of recategorization based upon current input; as such, it is transformational rather than replicative" (Edelman, 1987, p. 265). Memory is an active process of classification leading to recategorization and, thus, a partitioning of the world that is presented as one "without labels." Storage, to the extent that it exists, is one of procedures for mapping inputs to responses; hence, full representations of objects are neither stored nor required: *"It is the complex of capacities to carry out a particular set of procedures (or acts) leading to recategorization that is recollected"* (Edelman's italics 1987, p. 267). Memory is not replicative—it is not a set of particular features, attributes, or properties, however systematically transformed and stored.

4.2.2 Memory as Data versus Memory as Process

Within some psychological theories, the content of memory is cast as data residing in a data base. Most notably, this content is passive, references are made to it, it may fade with time, and in the case of short-term memory, new memories may reinforce or replace existing memories. Things happen to memories, memories don't do anything. Memory can be modeled as data on a blackboard or in a data base. In such schemes, something passes over the data, reinforcing some of it and degrading other parts of it.

Other theories, however, focusing on the dependence of learning on the learner's actions or responses, have posited that even ostensively static, declarative knowledge is represented in memory only as it is embedded in the larger context, including the actions and events through which it was learned and is used (e.g., Miller *et al.*, 1960; Neisser, 1976; Yates, 1985). That is, procedural knowledge is distinctly active. People have processes or procedures for doing things which they constantly modify and adapt to use in their everyday being. They take inputs from the world, process them (perhaps cognitively) and generate changes in the world around them. The processes work more or less well with each other depending on circumstances, and the paths through each of them are marked by branch points.

In this sense, memory is procedural knowledge and, in particular, has an active component that is dominant in our thinking about it. Accepting that contentive or "declarative" knowledge is embedded in procedural knowledge, semantic memory also becomes active and procedure-like, even actor-like. That is, memories are schemata in their most active sense. Short-term memory is a cluster of memory actors re-enforcing or suppressing one another, modulated by conflict resolution based on the class structure of the memory processes. Long-term memory and explicit and implicit memory are variations on this theme of memory as process.

Memory also changes form, as in chunking or in changing levels of abstraction. Perhaps several memory processes activate or coalesce into a single, well-integrated overarching process. In such a circumstance, many of the contributing memory processes may be lost to awareness in and of themselves. Hence, memory takes on quite an active role—much more in tune with the picture of mind as Edelman (1987, 1989) presents it. His reentrant and global mappings, in part, implement the processes of reinforcement, inhibition, and abstraction.

Abstraction, as presented in the preceding paragraph, is one form of activation rooted in memory. It is certainly possible to generalize this and think of a reinforced memory or a new abstraction as activating perhaps other procedures. An active view of memory fits nicely with the view of activation and inhibition as central to the dynamics of mind.

4.2.3 Multitasking, Automaticity, and the Psychological Framework

Broadly speaking, our concern here is the design of an HPP model with the capability to model faithfully the human operation of complex computer supported equipment. We would further like the model to be capable, of representing a range of human behaviors from novice to expert along one dimension and from errorful to error-free along another dimension. The range of tasks to be performed will have perceptual, cognitive, and motor components with varying degrees of complexity in each category and there will be occasions in which operators are required to execute several tasks "simultaneously."

From this perspective, the requirement to model multitask performance is an immediate inference with the corollary of recognizing the importance of the role of automaticity in supporting this level of human performance. The domain is very complex, but we are fortunate in that there is a wealth of knowledge on these issues in the experimental psychology literature.

The richness of the literature represents a diversity of views and our task has been one of establishing a psychological framework in which the more important threads in this research may be represented. A unique aspect of the framework under development is its ability to accommodate a reasonable portion of this diversity.

4.2.3.1 Dual-Task Performance. During this effort, we have closely examined two classes of task interference theory: one in which processing relies on graded resources, and another in which certain cognitive operations demand simultaneous access to one or a set of processors that can only serve one task at a time. We will refer to these as the Resource Model (Wickens, 1984) and the Postponement Model (Pashler & Johnston, 1989), respectively.

The Resource Model may be characterized by:

- processing that relies on resources, where the resources may be used in graded quantities, with greater allocation of resources producing more efficient or faster processing; and
- processing resources that may be divided into separate resource pools, where tasks that can proceed simultaneously do so with fewer resources available to each task and, hence, each task proceeds at a reduced rate.

In the Postponement Model:

- cognitive operations of each task demand simultaneous access to a processor or processors, where some processors can only service one task at a time;
- while a task is occupying a single-user processor or cluster of processors other tasks are postponed; and
- the individual stages of processing that constitute the single-channel bottleneck might be:
    - perceptual identification,
    - decision and response selection, or
    - response initiation and execution.

Within the group responsible for the development of the psychological framework, as in the broader community, there is not complete unanimity in the approach to be taken in addressing dual-task performance, but these apparently diverse viewpoints can be accommodated. The salient observation is that the single-user processors of the Postponement Model may be considered "resources" that are bimodal; that is, a task gets all of the resource or none of it. Indeed, we feel that we can go forward from this point utilizing the experimental results of both camps within our framework. Deferring a commitment to either of these camps or to a third approach to understanding dual-task performance is a reasonable step in the design of the HPP model.

4.2.3.2 Automaticity. Automatic processes typically differ from those of nonautomatic processes in the following ways (Logan, 1988a; Logan, 1988b; Newell, 1990; Shiffrin & Schneider, 1977; Schneider, 1985). Automatic processing is characterized as:

- fast,
- effortless,
- autonomous or obligatory,
- able to operate in parallel,
- consistent or stereotypic, and
- unavailable to conscious awareness.

Significantly, some theorists have stressed that some of these are *relative* characteristics (Cheng, 1985). That is, automatic processes are *relatively* effortless, *relatively* autonomous, *relatively* consistent and stereotypic, and *typically* unavailable to conscious awareness. Indeed, it is largely by probing the *relative* expression of these characteristics across experience and training that psychologists have begun to disentangle their underlying nature. Again, these explanations may be divided into two categories: those that attribute increases in automaticity to the increased efficiency of processes per se, and those that attribute it to increased efficiency resulting from more comprehensive and complete memory support for practiced tasks.

Within the view of automaticity as a product of processing efficiency:

- automatic processing is processing without attention;
- the development of automaticity represents the gradual withdrawal of attention; and
- attention is viewed as a resource, the demand for which diminishes with practice.

Within the view of automaticity as a product of memory support:

- automaticity is rooted in the memory or knowledge of the task;
- performance is automatic to the extent that it depends on single-step, direct-access retrieval of solutions from memory;
- automatization reflects the build-up of relevant traces in memory; and
- this build-up of traces underlies expert performance.

There are two alternative conceptions within the automaticity-as-memory viewpoint. According to one conception, most fully developed by Logan (1988a, 1988b), each experience, even of the same nominal event, results in an additional trace in memory. According to the other, repeated experiences strengthen existing memory traces (e.g., Cohen *et al.*, 1990). Notably, these theories are even now in active development. Within either category, there are some theories that are logically isomorphic with those in the other, but there are also some aspects of instance and strength theories of memory that are still a matter of controversy.

For the process of developing the psychological framework, it is important to recognize all such points of controversy. We wish to argue, once again, that in recognition of these differences and their potential importance, our goal is to put forward a framework in which the contrasting views held within our own group and in the larger community can be explored.

Most importantly, within both the resource view and the memory view, the capacity for parallel execution of multiple tasks depends on automaticity. Within the memory view, parallel execution of a task is possible to the extent that its solution or procedure has been compiled or learned as a whole; prior to that point, execution can be algorithmic or stepwise, at best. Within the resource viewpoint, nonautomatic tasks consume processing resources, effectively creating partial bottlenecks that lengthen completion times and delay execution. Once again, although the theoretic differences are significant and not to be ignored, they can be accommodated by recognizing the necessary refinements in the parallel execution of the nonattended portions of the tasks that make up human performance (see especially Navon, 1985). The flexibility in this aspect of the psychological framework is one that is intended to support further exploration of these issues.

4.3 Overview of the Functional Architecture

The flexibility in approaching multiple task behavior and automaticity, developed here in the psychological framework, is based in part on the commitment to the view of memory as process. The procedures or "knowing how" constitute the principal elements of memory, where each procedure in turn possesses or has access to the semantic or "knowing that" memory essential to its execution. The arrangement of these processes in a network facilitates their parallel execution. The bases for modulating this parallel execution are the views of automaticity and multiple task execution outlined in the previous paragraphs. The breadth of viewpoints spanned by the psychological framework being developed in this effort lies in the ability to express each view in the details of modulating the parallel execution of procedures.

In this subsection, we describe our current view of the functional architecture. The individual modules have been presented as distinct entities solely for expository reasons; because the functioning of the system depends exactly and only on the links within and between modules, the boundaries between them are essentially nonexistent. Thus, a sentence such as "The Scanning Module sends information to Memory Module" is intended to denote a continuous flow of information through processing elements, not a physical transmission of packet-like data through distinct "boxes."

4.3.1 The Scanning Module

The Scanning Module corresponds to the human operator's collective sensor apparatus. Its job is to transduce environmental information.

Operational Constraints and Input

The uptake of sensory information occurs automatically and in parallel within and across modalities. To the extent that the system is not intensely preoccupied with any particular task, the Scanning Module's sensitivities are diffusely receptive. However, when the system is actively seeking resolution, it may direct all the Scanner's modalities to attend to the sources it deems of interest and/or to minimize or avoid encounters with distracting sources of information. This is accomplished through priming by exciting or inhibiting the memory structures that are respectively expected to be relevant or irrelevant to the task at hand.

Decisions/Questions

(Passive.)

Temporal Requirements

None of its own except the time needed to reorient sensors physically.

Output

Sends scanned information to the Memory Module.

Hazards

Across modalities, important information may be missed because of physical competition or interference such as poor signal-to-noise ratios or misorientation of receptors.

4.3.2 Memory Module

The Memory Module contains a content-addressable, associative repository of experientially and inferentially gained knowledge. Its interactive and procedural dimensions notwithstanding, the Memory Module corresponds roughly to long-term memory in conventional information-processing models. Its job is to record and retrieve complexes of knowledge in service of perception, learning, thought, and action.

Operational Constraints and Input

Within this module, complex concepts, procedures, and events are recorded as interrelated constellations of their simpler parts. It is assumed that both recognition of new information or events and direct or indirect recall of old information depends on the activation of the knowledge complexes that comprise this memory. It is also assumed that each time a fragment of memory is attended (whether in service of perception, action, or thought) it results in a new entry or trace for that memory as modified by the context accompanying its activation (Hintzman, 1986; Logan, 1988a, b; Rumelhart & McClelland, 1986).

Recognizing New Information. As consistent with research and theoretically enabled by the connectivity of the memory substrate, it is assumed that all scanned events are automatically propagated "upward" from sensory elements to their highest familiar levels of coherence (see especially Marcel, 1983). In this way, the structure of the memory automatically affects produces preliminary categorizations and interpretations of the information as consistent with its prior knowledge.

At the same time, however, research suggests the existence of several factors that modulate the resonance of the memory to scanned information and, of these, several may be intrinsic to the memory. One such factor is frequency. Very familiar stimuli gain registration faster and with less environmental support than less common items. Presumably, this effect is due, at least in part, to the abundant representation and connectivity of the stimuli and the ensuing patterns of activation in memory. In keeping with this, a second factor directly related to the success and speed of stimulus recognition the extent to which its presenting context resembles that of past encounters. A third factor is recency. Repeated stimuli gain registration faster and with less environmental support than novel ones. One explanation for this effect is that activation of the memory decays, for example, exponentially, such that repeated stimuli benefit from the residual activation of prior presentations. Given the presumed nature of the memory, such residual activity may also be responsible, at least in part, for the fact that strong semantic relations with a prior stimulus seem to facilitate registration of incoming information. At the same time, however, research suggests that, to the extent overlapping but functionally distinct knowledge bundles are simultaneously activated, they tend to mutually inhibit one another. Finally, it is assumed that stimulus intensity is reflected in the speed and strength of memory registration; this may be due to nothing more than the proportional propagation of the strength of sensory activation.

In addition to such intrinsic factors, research suggests that human responsiveness to environmental information and the ability to access previously learned information are influenced by attentional factors. At a gross level of analysis, these factors can be divided into two categories. The source of both are extrinsic to the Memory Module, reflecting instead the involvement of the Cognitive Module (see below).

The first category of attentional factors expresses itself in awareness. Although the Memory Module processes arriving stimuli autonomously and in parallel, to the most comprehensive representations it can support, only those mappings that are acknowledged by the Cognitive Module reach awareness (Marcel, 1983).

The second category of attentional factors expresses itself in receptivity. In particular and distinct from the less-structured sorts of effects that might arise through associative overlap alone, attention seems to facilitate registration of information that is relevant to the observer's reigning expectations, while impeding registration of incompatible input or irrelevant sensory sources. At the moment, it seems that attentional influences on receptivity are accomplished through top-down priming or inhibition of the memory structure via the Cognitive Module.

Retrieving Knowledge. Like the act of recognizing input, the act of retrieving knowledge involves activating corresponding memory structures. Retrieval and recognition are thus similarly sensitive to frequency, context, and recency.

Yet, there are also certain important differences between the processes of recognizing new input and retrieving recorded knowledge. First, without the guidance of an impinging stimulus, the sought memory may be more difficult to locate. Whereas recognizing or registering incoming information consists, more or less, of a content-addressable matching process, retrieval of old memories can take on certain needle-in-a-haystack qualities. Thus, in efforts to recall past events, common or familiar stimuli are found and brought to mind more easily than less common or less familiar ones. In contrast, when the challenge is to recognize or discern whether a given stimulus has or has not occurred in the recent past, uncommon or unusual candidates are more easily accessed than common ones. This pattern has been demonstrated across a number of informational categories, including word frequency, text memory, memory for complex visual arrays, and argument.

A second difference in the activation processes underlying input recognition and knowledge retrieval is that the latter is an active process. Knowledge retrieval is only undertaken in service of a thought or activity in attention. Because the active efforts of the Cognitive Module are presumably single-channel and serial, retrieval efforts must compete with one another and with efforts to establish awareness of arriving information except where the pertinent knowledge is part and parcel to the same thought.

Finally, the Cognitive Module presumably has ready access only to the terminal or, equally, the highest levels of representation within the Memory Module. To the extent that a memory structure, even if active, is deeply embedded in the memory complex, it is difficult for the Cognitive Module to retrieve or gain awareness of it unless specifically trained to do so. This is the levels of processing effect (e.g., Craik & Lockhart, 1972). This effect is believed to be a major determinant of the reflective inscrutability of automatic behaviors.

Storing Information. Research evidence suggests that scanned information leaves an accessible imprint in memory only to the extent that it has been attended. Note that accessibility seems more a some-or-none than an all-or-none phenomenon. Research indicates that, provided that it has been attended (albeit momentarily), even information that has been dismissed as irrelevant or unimportant leaves an enduring trace. Even where such information is not recallable, its impact can be demonstrated through more sensitive techniques, such as recognition, priming, or saving measures. On the other hand, evidence suggests that unattended information leaves no such trace, even if it has been clearly sensed. This pattern has been demonstrated across a range of paradigms, including for example, studies of iconic memory, eye movements, and dichotic listening. Therefore, it is assumed that scanned information leaves an accessible imprint in memory only to the extent that it has been acknowledged, considered, or used by the Cognitive Module.

Temporal Requirements

While research has documented systematic variation in the time required for registration of arriving information within memory, the magnitudes of these differences are small. Their principal value in this effort lies in the hints they provide with respect to the architecture and dynamics of the system. In contrast, research indicates that the time and reliability with which established memories can be recalled or reactivated varies substantially and is influenced by such factors as recency, frequency, and the relatedness of sought knowledge to whatever has most recently been active.

Decisions/Questions

(Passive.)

Output

The Memory Module's responsiveness, as reflected in the differentially activated complexes of knowledge, represent its resonance to both incoming and internal stimuli. The Memory Module presents arriving or retrieved information to the Cognitive Module for its consideration. Arriving information that gains representation in the Memory Module but is not considered by the Cognitive Module does not become available to conscious attention and does not leave any enduring trace in memory. On the other hand, it does temporarily facilitate reception of repeated or similar signals.

Hazards

Although arriving information is registered in parallel, functionally independent sources may interfere with one another through crosstalk or overlap of representational structures (see Navon, 1985). In addition, whether for reasons of incomplete prior knowledge or spuriou. contextual cues, the Memory Module may occasionally register information in ways that are obscure or misleading with respect to the system's reigning goals and interests. Finally, the most available relational structure of knowledge in the Memory Module may be relevant to but not amenable for any given activity of interest to the Cognitive Module; in these cases, thinking (i.e., attention and cognitive effort) is required to recover or construct relations that are of direct interest.

4.3.3 Cognitive Module

At any moment in time, the Cognitive Module contains a record of all pending plans and goal structures. Its job is to accept, retrieve, and—as necessary—reconstrue knowledge from the Memory Module so as to enable pending plans and goals to be triggered, pursued, or satisfied. Pending plans and goals can be particularized only by instantiating their slots with knowledge from the Memory Module. The Cognitive Module thus encompasses the functions of consciousness from pre-Titchner theories, the goal structures of recent cognitive science models, and the behaviors of the short-term or working memory as envisioned by conventional information-processing models.

Operational Constraints and Input

At any moment, the Cognitive Module may be maintaining a number of open goal structures. This can be distinguished as either active or open.

Active Goals. A goal is active when the Cognitive Module is actively engaged in its instantiation or pursuit. With respect to active goals, the Cognitive Module may be seen as a single-channel, serial processor in the sense that only one goal structure may be active at a time.

Phenomenologically, a goal is active when goal-relevant considerations accrue attention and, typically, awareness. Functionally, a goal is active when the Cognitive Module is actively evaluating the relevance of presenting or retrieved memories to the goal structure. This process requires cognitive effort because it involves the consideration of relationships within and among memories whose definitions are extrinsic to the memories themselves; that is, whether or not these relations are directly and accessibly represented in the memory, their relevance is defined by the goal structure. Minimally, then, this process involves mapping a selected subset of the presenting relations of some knowledge structure onto the goal structure. Sometimes it involves analyzing otherwise disjoint memory structures and evaluating relationships among their components against the goal structure. Sometimes the identification of the memory structures to-be-related must itself involve cycles of searching, evaluating, and searching again. The process is single-channel because its success depends critically not on *finding* relationships among active fragments of memory, but on selecting from among the many available relationships that meet its needs. Such a process necessarily entails the ability to focus on candidate relations while ignoring or gating out other associations.

By implication, the cognitive effort and time required for this process will depend on the definiteness and compactness with which the goal's requisite data and responses are specified. Yet, the degree of specification possible depends, in turn, on two factors.

The first factor is the person's experience with similar instantiations of the goal. With sufficient familiarity and appropriate context, a goal can be activated by the mere presentation of any of its triggering events. However, no event can activate a goal's larger structure, nor can it effectively prime the Memory Module for information reception or prepare it for a response unless the corresponding knowledge within each has been well-learned and consolidated. Once consolidated, the whole routine may become so well defined, and thoroughly represented and supported in memory, that it can be triggered and accomplished, start to finish, with near automaticity. In contrast, to the extent that relations must be actively constructed and assessed rather than retrieved, goal definition, monitoring, and attainment will necessarily require time and attention.

The second determinant of the definiteness and compactness of goal specification is, of course, the nature of the goal. Some goals carry data analysis and response requirements that are sufficiently complex or ill-specified that, even for experts, they are attentionally demanding.

Open Goals. For a goal to be open, it need only be among those currently represented in the Cognitive Module. It is assumed that a number of goals may be open at once. In the ATC environment, for example, goals that might be simultaneously open include accepting handoff of aircraft, maintaining separation for aircraft in a sector, inputting decisions into the computer, reorganizing the board to reflect current work space, receiving communications from pilots, assisting an apprentice, completing the current shift, general monitoring of the environment and general maintenance of personal comfort, and any variety of goals unrelated to the immediate context. An open goal is closed when it no longer begs completion. Either it is no longer of interest or it has attained closure.

The observable characteristic of an open goal is that arriving information that is presented by tne Memory Module and relevant to the goal's triggering, pursuit, or satisfaction is likely to evoke awareness by activating the goal. The mechanism by which this happens is presumably that open goals interarticulate with those fragments of the knowledge in the Memory Module that they deem relevant. Indeed, this may well be the mechanism underlying the "aha!" or "now-I-remember!" phenomena wherein the discovery or remembrance of the missing piece of an abandoned but unclosed goal may cause it unexpectedly to pop back into awareness. More generally, the Cognitive Module's means of searching for information to instantiate its goals serves equally to prime their arrival. Moreover, the more precisely a goal's sought information is defined, the more focused will be the interarticulation between the Cognitive Module and the Memory Module, and the more effective the priming.

Dynamics of Attention. An important dilemma emerges defined through the preceding discussion. First, it has been hypothesized that the attentional capacity of the Cognitive Module is serial and single-channel; that is, it is capable of supporting thought about only one problem at a time. Second, it has been asserted that the relevance of any arriving datum cannot be assessed without lending attention to it. Thus, the following problem arises. When a goal is active, attention is focused on its definition and attainment. Yet, the distinctive characteristic of complex semi-automated environments is that when confronted with a number of tasks simultaneously, the operator must be able to manage the ensemble as a whole. This capability depends critically on the operator's ability to notice events, track the status of systems, and evaluate all such variables in a timely and appropriate manner. How is it possible for an operator to work on an active goal while monitoring and updating the status of others that are open and pending?

In an attempt to resolve this dilemma, it was hypothesized that the goals represented in the Cognitive Module differ from one another in their level of activation. More specifically, the level of activation may be a joint function of their urgency, costs of mishandling, and recency. This rule generally ensures that the active goal has the highest level of activation at any moment. To capture the spirit of Kahneman's (1973) argument that there exists some maximum available capacity that can be devoted to any given task, a maximum total activation across goals may be assumed. A corollary of this assumption is that the presence of one or more highly activated goals in the space will result in concomitant, if temporary, diminution of a person's sensitivity to the needs of other open but pending goals.

To enable the operator to stay vaguely on top of open but deferred tasks, we additionally hypothesize that the Cognitive Module is capable of momentary time-slicing. Through such time-slicing, the Cognitive Module may briefly divert attention from an active goal to make preliminary assessments of the nature of arriving data relative to other open goals. If the data are judged irrelevant, attention is quickly returned to the task from which it was diverted. (An interesting corollary of this assumption is that where data are relevant but not obviously so, they may be also dismissed in this process. This is an error to which novices are shown to be more susceptible than experts.) If the data are judged relevant to some open goal, the activation level of that goal will be boosted through the mapping process (the recency effect); beyond that, the activation level will be in proportion to any implications as to its urgency and associated costs. (Note that the latter depends on the transparency of the data's implications, which in turn must covary with the operator's expertise.) On rare occasions, such arriving data may push the activation level of a pending goal structure beyond that of the goal from which attention was diverted; in these cases, the operator will shift attention to this newly active goal, abandoning the one from which attention was diverted. More often, the original goal will retain preemptive activation such that it will recapture the operator's attention as soon as the new data have been entered. Even in these cases, however, the activation level of the newly modified goal structure may be sufficiently high to capture the operator's attention upon completion of the currently active goal or subgoal.

But still there remains a problem. The Cognitive Module cannot assess the relevance of *every* arriving datum. Otherwise it would be continuously interrupted and would never have time to complete even the simplest active goal.

To solve this dilemma, it is assumed that, against the background of diffuse and continuous stimulation that bubbles up through the Memory Module, only those signals that pass some critical level of activation in the Memory Module receive attention. Thus, arriving signals are more likely to exceed this level to the extent that they are physically salient, intense, or recurrent. Otherwise, the likelihood with which arriving information will exceed this level will depend on how strongly it is primed by some goal structure in the Cognitive Module.

In turn, the strength with which arriving data can be primed by a goal structure depends on two factors: (1) the level of activation of the goal structure, and (2) the definiteness with which the nature of goal-relevant data is specified by the goal structure. For example, the goal of receiving a telephone call is generally open when you are in your home or office. Much of the time this may be a relatively low priority goal. Nevertheless, the sound and location of your phone are so tightly specified by the goal such that, want to or not, you generally alert to the bell when it rings. If the goal is also highly activated for example (i.e., that is, if you are urgently expecting a call), you may reach for the telephone before anyone else in the room has heard it ring. In contrast, where goal-relevant information is poorly defined, priming can be diffuse at best and possibly misguided. Thus, you may be actively wrestling with an interpretation of a difficult problem, but without a relatively definite and accurate understanding of its structure, you may fail to notice key elements even when they are right before your face.

Easy and difficult problems may differ from each other not merely in familiarity but also in the complexity of their goal structure. That is, difficult problems may entail a host of interrelated subgoals. Consequently, an operator may be inclined to neglect other goals while working a complex problem. First, the longer the complex problem is worked, the greater will be its relative activation due to recency of attention. Second, the outcome of each subgoal of a complex problem, will typically trigger activation of another subgoal in the complex.

Temporal Requirements

The temporal requirements of the Cognitive Module are significant. Depending on its novelty and complexity, the solution of any active goal may require minutes or even hours. Though it is possible to maintain currency on pending goals through time-slicing, this also requires time and effort. Research indicates that, even where no reorientation of the Scanning Module is required (e.g., no eye movement), the process of switching attention between simple informational sources requires several hundred milliseconds; the time required for mental shifts between topics or semantic domains is several times greater (Adams et al., 1991). Moreover, the temporal requirements of the Cognitive Module—in receiving, evaluating, and synthesizing relevant knowledge as well as in response specification—are expected to vary with the person's experience with the problem; all else being equal, it always takes less time to activate than to construct a memory.

Decisions/Questions

What is the nature of the information needed to resolve open goals? Is this information available in memory? Can it be searched for it in the environment? What actions can be taken to further or satisfy pursuit of this goal?

Output

The Cognitive Module articulates with the Memory Module to prime reception of arriving information, activate previously known relevant information, to make records of the relations with which it has worked, and trigger response or action schemata in service of goal pursuit or satisfaction. Moreover, conscious awareness is taken as the phenomenal showing of the Cognitive Module's active focus.

Hazards

The principal hazards of the Cognitive Module derive from its single-channel property. Although the Module may shift attention to any arriving information, it can do so only at the expense of interrupting any erstwhile active task. The less time and effort the operator invests in interpreting a new datum, the greater the likelihood of misconstruing the nature and priority of its implications. Alternatively, the more time and effort tne operator invests in interpreting a datum, the greater its disruptiveness to ongoing activities and its potential for blocking notice or proper interpretation of other available data must be. Moreover, research demonstrates that to the extent people are mentally involved in a task, they are literally less able to notice unrelated events around them. Reaction times slow, the functional visual field shrinks, and even highly salient events may fail to evoke notice (Adams et al., 1991). Finally, to the extent that goals are disjoint but the nature of their supporting data are not, people must be vulnerable to errors of misinterpretation. In particular, an ever-present threat is that an incoming datum may map readily onto the knowledge in one open goal and, within that context, may be thoroughly examined and willfully dismissed as inconsequential—even though it is, in fact, critically important to some other pending goal.

4.3.4 Response Module

Like the Scanning Module, the Response Module is a peripheral slave system. Its actions are triggered by the Cognitive Module and sculpted by the Memory Module. An action may be carried out completely and autonomously only to the extent to which it has been overlearned or practiced. To the extent that an action requires attentive monitoring or stepwise specification, then assessment, periodic intervention, and redefinition by the Cognitive and Memory Modules are commensurately required.

5.0 A "CHAOS" ARCHITECTURE

Section 4 discusses a set of psychological attributes and human cognitive functions that one would hope to capture in an HPP model of operator behavior. The findings of that section rest, for the most part, on two foundations: (1) empirical observations of operators of varying skill levels performing complex tasks and (2) attempts to "explain" the observations from the global perspectives of psychology and cognitive science (i.e., to fit them into a comprehensive and comprehensible theory of behavior).

Reducing such findings to practice and devising a working simulation that is simultaneously faithful to theory, successfully emulates the observed behaviors, and accurately predicts behaviors that have not yet been observed is a difficult task for many reasons. Four of these reasons are identified below.

- The conceptual level at which the theory is formulated and/or the language used to express it may not provide an adequate set of "pointers" to the necessary formalisms and structure that must be incorporated in the model.
- The data required to determine the validity of the model's behavior (and hence the adequacy of its fine structure) may not be available.
- Although the theoretical underpinnings of the implementation are locally sound over the range of empirical observations from which they are derived, they may not be globally sound and may not produce model behaviors that are appropriate over the full range of interesting parameter values.
- The software and hardware tools available to package theoretical elements are simply not adequate for the purpose.

Except for Reason (4), such difficulties could easily be addressed—perhaps they would not even exist—if it were possible to observe, and then to model, the organization and functioning that gave rise to the operator's behavior in the first place. Unfortunately, despite the steady accumulation of information relating to neural properties (see, for example, Churchland & Sejnowski, 1990; Crick & Koch, 1990; Damasio, 1989a, 1989b; Eckhorn et al., 1988; Edelman, 1987, 1989), an understanding that is sufficient for the sufficient understanding of modeling of global human performance processes does not yet exist.

The approach taken here to bridging gaps among theory, observation, and mechanism has been to formulate what might be described as a "neurologically plausible" model that captures the psychologist's and cognitive scientist's functional view of an apparatus that could produce the behaviors observed and, at the same time, appears not to violate what is known concerning brain organization and dynamics. Additionally, although it was not an explicit criterion during model formulation, what has emerged promises to be implementable within the constraints of existing hardware and software.

This section begins with a discussion of the general outlines of an architecture, called Chaos within which the behaviors and theoretical propositions identified in Section 4 can be modeled and tested. Examples of the functioning of this architecture in the context of ATC tasks is then presented. These are followed by a series of specifications that identify specific inputs and outputs associated with each task and with characterizations of the types of errors anticipated with respect to the completion of each task by a novice controller.

5.1 Overview of the Chaos Architecture

The Chaos architecture can be thought of as consisting of three distinct layers:

- a **conscious process**, which "glues" together possibly disjoint inputs of
- **attention**, which "shifts" among the
- **parallel processes/procedures** organized in a data-flow network, which actually perform the signal transformation underlying the information processing that produces intelligent behavior.

5.1.1 Conscious Process

While some of the characteristics of the conscious process can be described (e.g., it appears to be single-channel), it is difficult to formulate a clear model of what consciousness does and why it is necessary. One view of the conscious process component assigns it properties similar to attention as described above, namely single-channel, accessible to introspection, and (probably) having some "privileged" abilities to construct a trace or otherwise be enabled to reconstruct "where it has been." Consciousness may thus be closely related to metacognitive abilities. Many theorists believe that consciousness is the source of "free will" and goal-directed behavior.

The purpose of consciousness may be, as Dennett postulates (Dennett, 1991), to integrate time-slices of attention and to provide the illusion of continuity, as a film generates an illusion of motion. But the question then arises, what ***function*** does this serve and why is it necessary? As was indicated above that the "movement" of attention may be directed in part by the conscious process. This makes intuitive sense and "feels" right, but leaves unexplained the issue of the basis for such attention-directing decisions by consciousness.

As was suggested in Section 4, consciousness may contain a similar type of activation-based network of units as does memory—except that in the case of consciousness these units are goals to be achieved rather than memories of past experiences and inferences. However, if the existence of two similar but distinct structures is postulated, two questions arise: Why are two structures necessary? Can both goals and memories reside within the same space?

Within the proposed framework, it will be possible to experiment with configurations ranging from ones in which consciousness plays the role of final arbiter among competing goals to ones in which consciousness has no causal relationship to the processing and is, therefore, purely epiphenomenal. The latter variety, it should be noted, has been suggested by some in the recent literature (Jackendoff, 1987; Dennett, 1991).

5.1.2 Attention

Attention seems to be single-channel, mobile (analogous to Reason's (1983) "blob"), and accessible to introspection. Furthermore, it is affected by external stimuli (e.g., attention is directed towards an unexpected noise), unconscious processing (e.g., attention is directed towards an idea that "pops" into our heads), assuming that consciousness plays a causal role in cognition conscious processing, as it implements its goals.

As far as the functionality of attention is concerned, one hypothesizes that the agent's attention is closely tied to sensory information processing and is necessary to mediate the construction, refinement, and "execution" of internal models of the external environment. Thus, attention might be required to initiate an action, monitor the progress of some action, or mark the completion of a particular action-all of which involve a greater degree of direct, tightly interwoven interaction of the organism with the external environment. Once the task is under way, the organism may run "automatically" with only periodic checks.

5.1.3 Parallel Processes/Procedures

The core of the Chaos architecture is an assembly of communicating procedures that form a data-flow network and constitute the memory and the processing apparatus of the Chaos model.

The data-flow network of procedures forms the "cognitive layer" in the transformation of sensory input signals into motor output signals. Throughput occurs in one direction only, but there are many feedback links, called reentrant maps by Edelman (1987), that support proactive behavior and, in general, integrate the sensory-signal directed processing with the organism's current goals and past experience, as stored within the procedures. These reentrant links thus allow the integration of internally generated data, including goals and metacognitive assessments, with the incoming sensory data.

Individual procedures within the network resemble the individual units of a connectionist model in that they receive inputs from other procedures and their output is some function of the incoming data. Like connectionist units, individual procedures have distinct levels of activation and thresholds which determine whether processing will take place and whether any output will be produced. The activation level of a procedure is a function of numerous factors, including the strength/type/recency of incoming data, current level of activation, local thresholds for activation, and presence of inhibitory signals.

However, the similarity with connectionist units ends there. Unlike those models, Chaos procedures work at a symbolic level and the individual procedures can process highly complex incoming data, not simply zeroes and ones.

The procedures exist at varying levels of abstraction and complexity, ranging from simple individual feature-detectors to complex "chunks" representing, for example, a group of potentially colliding aircraft. Primitive procedures/processes combine to form more complex ones. These combinations have different degrees of stability. In general, the more practiced a task is , the more entrenched and stable the relevant complex process combinations and the paths linking them will be.

An abstract view of the procedural network is shown in Figure 5. This figure represents the general organization of the network, the flow of signals through the network, and the network's connection with the sensory and motor outputs. An example of an instantiation of this type of network in the context of an ATC scenario is shown in Figure 6. In this figure, the sets of both the expected sensory inputs and the specific procedures required to perform the ATC tasks are shown.

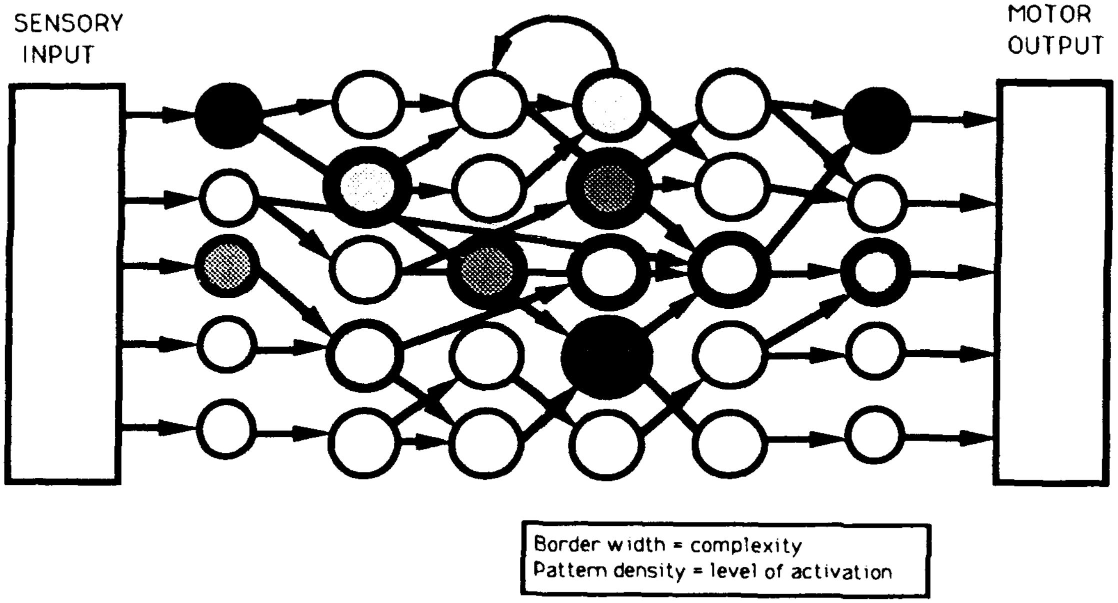

SENSORY INPUT
MOTOR OUTPUT
Border width = complexity
Pattern density = level of activation

Figure 5
Patterns of Process Complexity and Levels of Activation

The procedures in the network "do the work" by gradually transforming the incoming sensory data, merging these data with internally generated goals or existing data, and eventually generating motor output and/or modifying related internal structures. Procedures can be created, updated, activated, reactivated, and inhibited, or they can "decay" and become disconnected.

When sensory or internally generated signals flow through the network, they begin to create a spatiotemporal pattern of activation among the linked procedures. It is this pattern that corresponds to what psychologists refer to as "schemata" and AI researchers refer to as frames, scripts, or objects. Figure 7 is an example of a type of patterning that could occur in an ATC

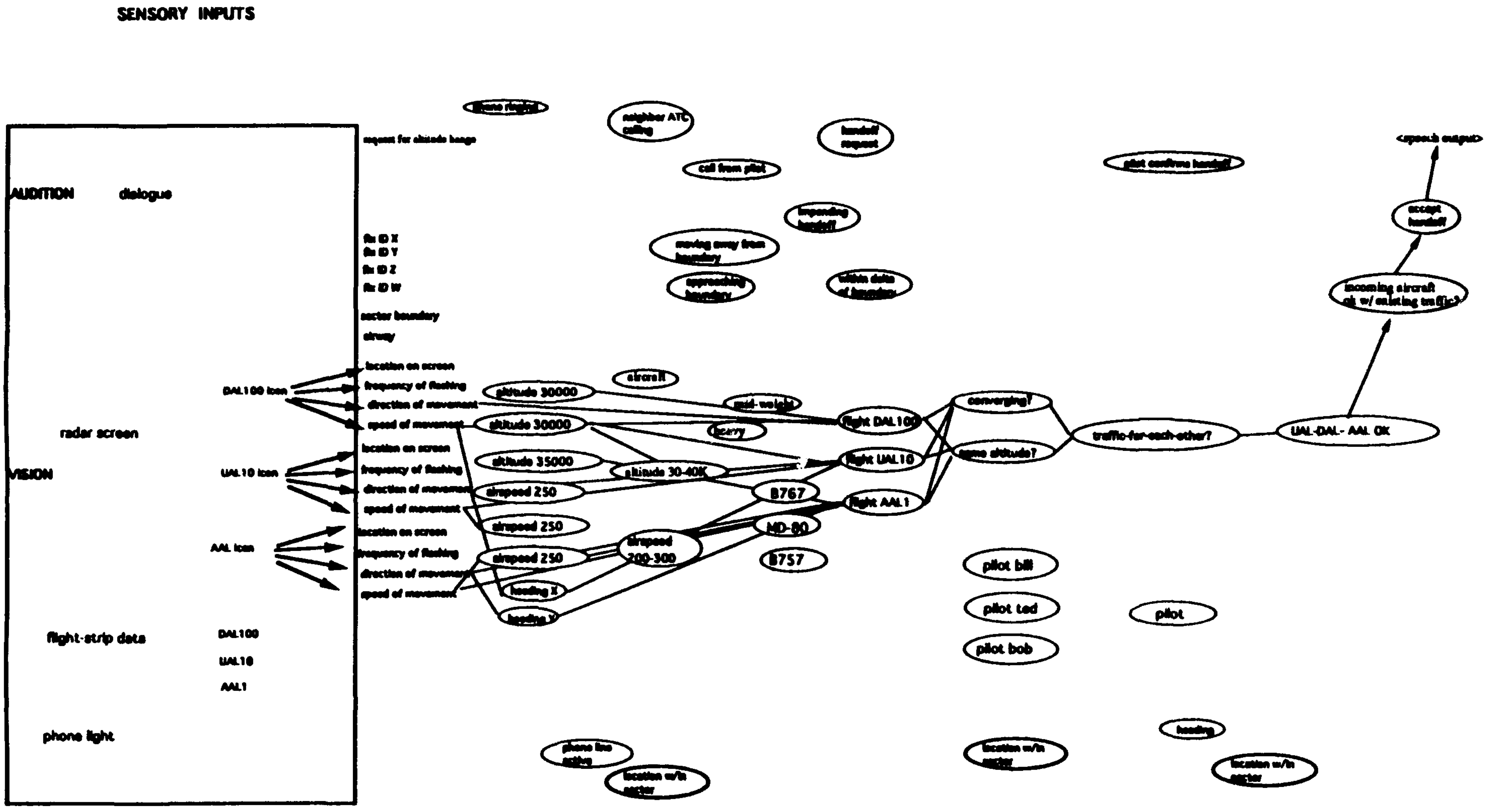


Figure 6
Sample Data Flow Through a Set of ATC Procedures

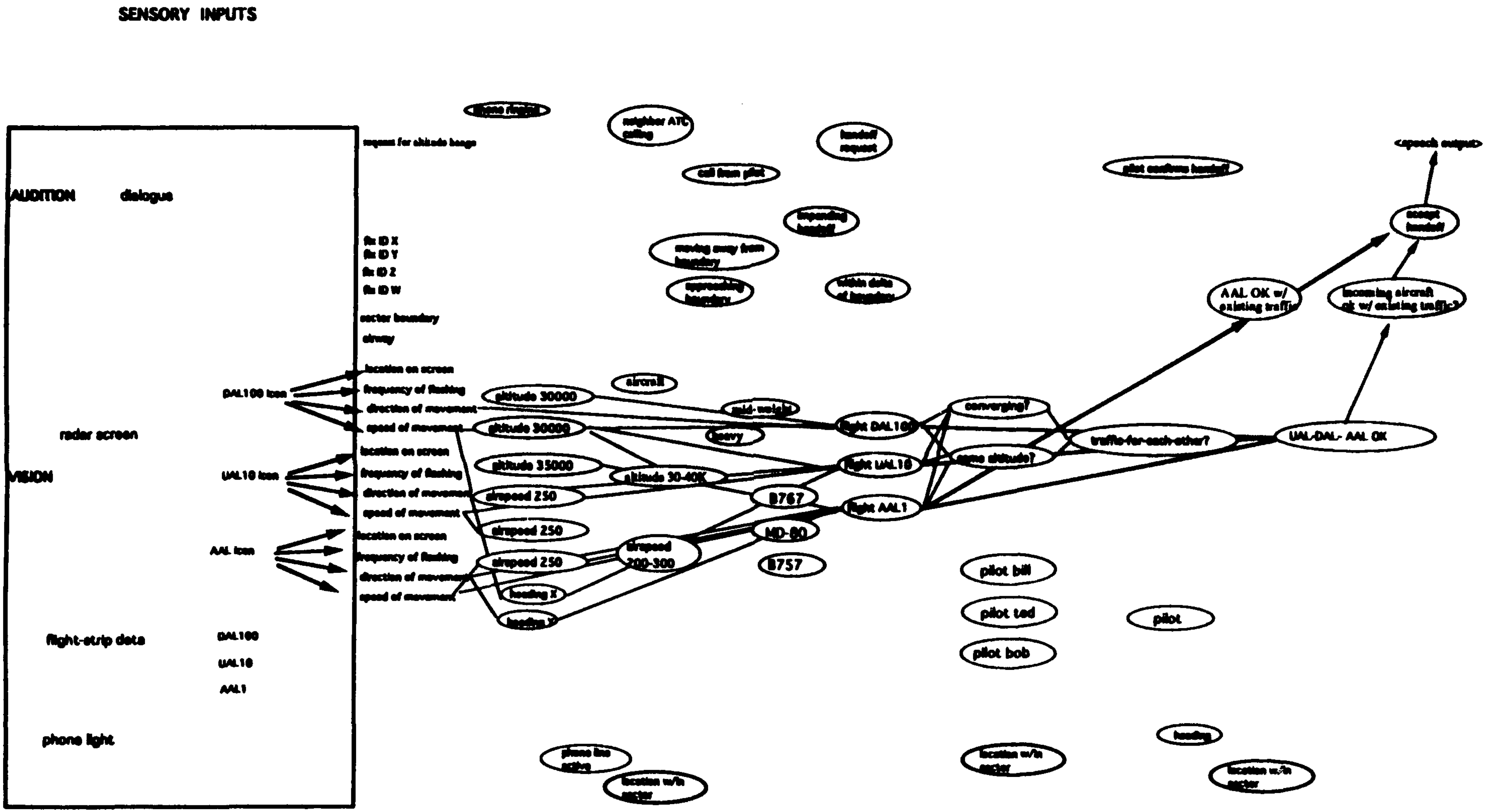


Figure 7
Sample Data Flow Through the Procedure Network During Preliminary Scanning of Flight Strips

scenario at the point at which a controller begins to scan a set of flight strips in order to decide whether three flights (e.g., DAL 100, UAL 10, and AAL 1) represent "traffic for one another."

A critical property of the schema is its dynamics. Given a certain set of sensory input and internally generated flow, a subset of procedures is activated to varying degrees as the signals flow through the network. The more practiced a task is (i.e., the more practiced the input-output mappings), the more similar the individual signal flows will be to one another for distinct instances of the task. However, the exact signal flows will probably never be the same in a network of any complexity. It is this type of variability that allows organisms to generalize and produce novel behavior (Edelman, 1987).

Processing within this network is both sequential and parallel. Sequential processing results from the forward flow of data through the network. Parallel processing results from the existence of subnetworks of procedures within this overall structure that can carry on isolated computation that may or may not be modified by further incoming data.

5.2 Review of Novice/Expert Differences

This section summarizes the differences between novice and expert controller performance in terms of the elements of the proposed implementation.[2] The variations in performance reflect differences in internal organization and processing and fall into the following general categories: (see Figures 8 and 9):

- extraction of environmental stimuli/features;
- internal representation and processing of relevant data; and
- control and management of subtasks.

5.2.1 Feature Extraction

The internal processing of a novice is characterized by a poorly differentiated extraction of the appropriate features from the external environment, lack of coordination among the internal procedures, and wrong or suboptimal motor output. In terms of the "signal flow through a procedure network" model of processing that characterizes the Chaos architecture, the cognitive processing of the novice is characterized by slower, more deliberate processing of smaller data chunks.

[2] It is important to note that by the time controllers have received instruction at the FAA Academy, become certified, and begun their assignments at an ATC sector, they are already functioning at well beyond novice level.

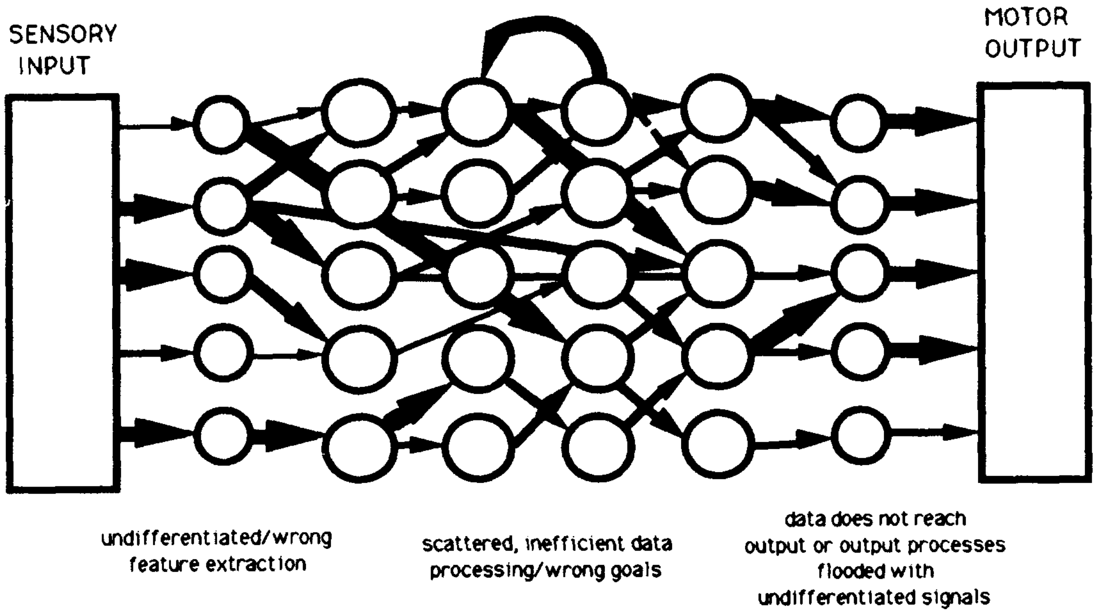

SENSORY INPUT
MOTOR OUTPUT
undifferentiated/wrong feature extraction
scattered, inefficient data processing/wrong goals
data does not reach output or output processes flooded with undifferentiated signals

Figure 8
Signal Flow Through a Novice's Network of Processes

Note: The processes connected by bold lines are strongly associated while those connected by normal lines are weakly associated.

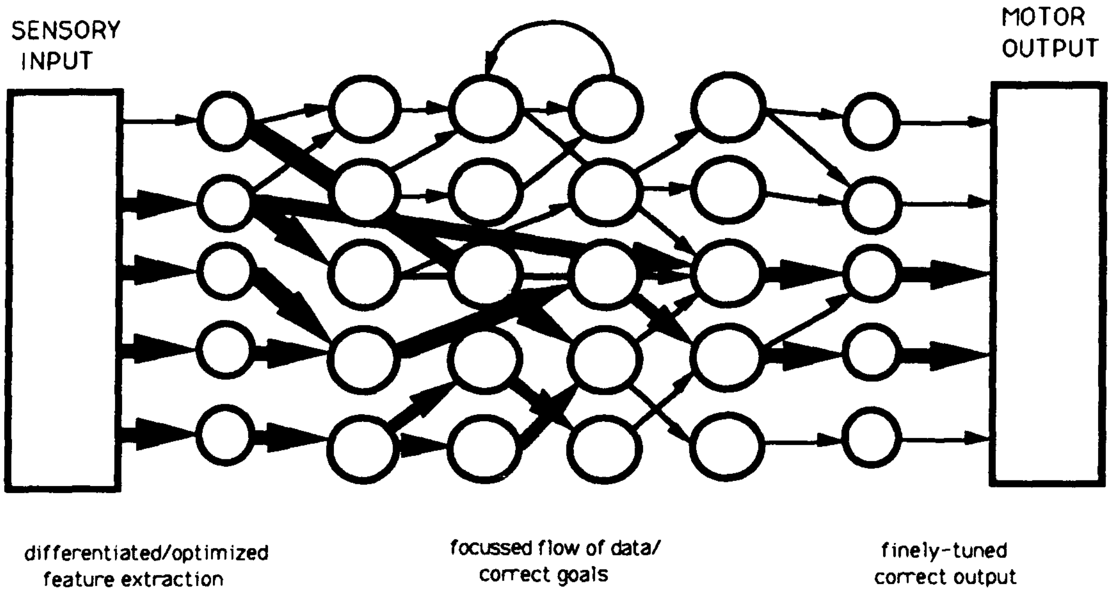

SENSORY INPUT
MOTOR OUTPUT
differentiated/optimized feature extraction
focussed flow of data/ correct goals
finely-tuned correct output

Figure 9
Signal Flow Through an Expert's Network Processes

Note: The processes connected by bold lines are strongly associated while those connected by normal lines are weakly associated.

By contrast, both the performance and internal processing of an expert is characterized by finely tuned feature extraction; efficient, quick data processing; and optimal or nearly optimal motor output. Experts' behavior is also characterized by a high degree of automaticity and multitasking, as well as efficient task-splicing and task-switching.

5.2.2 Internal Representation and Processing

An experts' performance is highly organized, combining both domain and task control knowledge into a small number of efficient procedures that channel the flow of signals to the proper motor outputs. Experts have more detailed domain knowledge and more accurate task control knowledge; this knowledge affords them greater control over their processing. Experts have a better idea about which data are required, where to get them, and how long a given task will likely take. Thus, they can better plan their behavior. Novices are typically overwhelmed by the complexity of the task, have only a rudimentary idea about the exact processing and data requirements for each step, and must engage in much repetitive behavior to "remember" the partial problem-solving states in order to achieve the goals. The experts' internal structures are more stable, thereby requiring less "conscious refreshing" than those of novices.

The differences among the key features of novice and expert behavior are summarized in Table 2.

Table 2 Summary of Novice-Expert Differences

FUNCTION	NOVICE	EXPERT
Feature Extraction from Environment	slow undifferentiated wrong	fast differentiated optimized for context/task
Internal Structures	unstable lack "tuning" unchunked	stable fine-tuned for task/context
Signal Flow through Procedure Network	inefficient not well localized	optimized channeled through minimal-length pathways
Degree of Automaticity	none or low	high
Ability to Perform Multiple Tasks	none or low	high

5.3 "Specification" Sheets for Sample Procedures

To make the discussion above more concrete, examples of several procedures involved in the handoff of an aircraft by one ATC sector controller to another are described in detail below. A "spec sheet" format is used that identifies different inputs and outputs and ties each procedure to the processes described in Section 4. For each procedure, both external (sensory) and internal (generated within memory) inputs are described, an indication is made as to the degree of automaticity that can be achieved for the procedure (which is related to the operator's ability), and possible errors that might occur are identified in notes below each sheet.

For example, in the first spec sheet, one input is, "Rate of flashing of aircraft icon on radar screen." This means that a procedure that is sensitive to the appropriate rate of flashing is expected to be triggered by the aircraft icon and to send an excitatory message to the "Notice flashing icon" procedure being described. Note that the procedures fall into several categories. For example, in the first spec sheet, the procedure representing a particular aircraft is part of the internal representation of the domain knowledge and the current environment, while the procedure "Phone call from neighboring controller," is a procedure representing a step in the handoff task.

Note also that the sample procedures shown in the "spec sheets" describe activities at varying levels of complexity, ranging from simple stimulus-response behavior, (such as detecting a particular frequency of flashing) to highly complex, cognitive tasks (such as deciding whether two aircraft are on a collision course). Although these activities are described as individual procedures, it is to be expected that the more complex procedures will consist of large numbers of simpler ones.

Spec Sheet #1

Procedure: Notice flashing aircraft icon

Relationship to Processes in Section 3: This is a relatively low level (e.g., closest to the raw, sensory data) procedure and is thus associated with the scanning function. It does, however, receive signals from the internal expectations and current goals.

Relationship to ATC Scenario: This procedure is invoked when the icon representing the incoming aircraft begins to flash, indicating that it is nearing the sector boundary and that a handoff is therefore imminent.

Input

- Signals from procedures processing primarily external environment/sensory data
- Rate of flashing of aircraft icon on radar screen
- Size of flashing icon on screen
- Object within delta of sector boundary
- Particular aircraft generated from flight strip examination
- Particular aircraft generated from knowledge of standard pattern of traffic

Output

- Particular aircraft
- Phone call from neighbor controller

Notes on Spec Sheet #1

Automaticity. The degree of awareness/automaticity varies, depending upon the degree of expertise, current workload, and specific contextual factors (e.g., a particularly busy area or an unexpected aircraft). In general, the more "regular" the particular aircraft is (i.e., the more it is part of a fixed, recurring pattern), the more automatic the processing of these procedures will be and the less conscious awareness the operator will have that the processing is taking place. If the aircraft is regularly part of a pattern, this activation may activate the remainder of that pattern and set up further expectations.

Error Potential. It is possible that, in situations where workload is particularly high and the operator is under stress or otherwise distracted, a procedure representing the wrong aircraft will be activated. For example, the controller may have just finished handling a particularly difficult situation and the procedures representing the participating aircraft may still be in a state of high activation. When a signal arrives, indicating that a flashing icon has been detected, the signals from the still-active procedures will become associated with the signals representing the flashing

Task A may have met all its data requirements, but simply not yet received sufficient activation icon, resulting in a mistaken association of the flashing icon (i.e., aircraft to be handed off) with the still-active representation of the aircraft that the operator has just finished handling. This association remains "consistent" until the controller receives a phone call from the neighboring controller with information about the new aircraft. At that point, the current aircraft is no longer consistent with the controller's representation and a correction must be made by activating the correct procedure via metacognitive processing.

Another example of a potential error situation occurs when a wrong pattern of aircraft (chunk) is activated, causing false expectations about the anticipated movement of the individual aircraft.

The two examples above describe errors in activating the wrong internal representation of an aircraft. Another type of error can occur when unexpected events "collide" in the middle of a task. For example, if the operator is primed for a phone call from the neighboring controller and another phone call arrives, the operator may take longer to process this call because the internal context needs to be changed. Furthermore, the operator is more likely to make a mistake by pressing the wrong line-selector on the phone.

Spec Sheet #2

Procedure: Notice call from neighboring controller

Relationship with Processes in Section 4: This procedure is a low-level procedure associated with the scanning function.

Relationship to ATC Scenario: This procedure is invoked while the icon representing the incoming aircraft is flashing, to recognize the set of stimuli that represent an incoming call from the neighboring ATC controller.

Input

- Signals from procedures processing primarily external environment/sensory data
- Phone ringing
- Signals from procedures processing internal expectations and current goals
- Notice flashing aircraft icon

Output

- Direct gaze towards phone-line selection buttons (novice response)
- Immediately (automatically) reach for correct line button and press to select line (expert response)

Notes on Spec Sheet #2

This procedure is a low-level procedure that is likely to be highly automated. Upon receiving the auditory signal of a phone ringing in the context of a handoff, the novice will look toward the line-selection buttons on the phone. An expert will already have this mapping fixed and will automatically reach for the phone; only at the last instant, if at all, will the expert look at the exact line button before pressing it.

Spec Sheet #3

Procedure: Select phone line

Relationship to Processes in Section 4: Low-level procedure associated with the response function, since it involves motor activity.

Relationship to ATC Scenario: This procedure is invoked as part of the initial handoff, where the controller accepts a phone call from a neighboring ATC controller as the AAL aircraft is about to enter the sector.

<u>Input</u>

- Signals from procedures processing primarily external environment/sensory data
- Rate of flashing of phone-line button
- Location of phone-line button
- Signals from procedures processing internal expectations and current goals
- Phone call from neighbor

<u>Output</u>

- Press button to select line
- Procedures handling dialogue with controller

Notes on Spec Sheet #3

This is a relatively simple procedure, although it does involve (as would any low-level motor procedure) a tight coupling between motor action and feedback regarding the location and activity of the effector. Because of its simplicity, the procedure is likely to be highly automated, even at the novice level.

<u>Error Potential</u>. An example of a simple error is pressing the wrong line-selection button. A more complex error may occur when the controller is expecting a "flashing" stimulus (e.g., from an incoming aircraft), mistakes this stimulus for the flashing of the phone line, and presses a line-select button.

Spec Sheet #4

Procedure: Dialogue with controller

Relationship to Processes in Section 4: This is a highly-complex procedure that, in reality, will be implemented as an aggregation of simpler procedures. It requires close cooperation/coordination among the scanning function, the cognitive and memory functionality, and the response function. Since it requires language processing, both the cognitive and memory functions are highly involved in the procedure. However, much of the processing is automated or partially automated, since the dialogue tends to be fairly restricted and there is little complex problem-solving. The scanning function is involved to process the auditory speech signals from the neighbor and to provide feedback for controller's own speech. The response function is involved to generate speech.

Relationship to ATC Scenario: This procedure forms the core of the handoff procedure, where the controllers are communicating about the impending handoff.

Input

External

- Phrases and key words which are part of the handoff protocol

Internal

- Some degree of input from all procedures relevant to the current context
- Procedures representing phrases/words relevant for handoff protocol
- Procedures representing other relevant aircraft in a related chunk, if appropriate

Output

- Handoff decision procedure
- Indicate completion of step-either inhibit procedures that were "waiting"for the call or stop activating them.

Notes on Spec Sheet #4

The error potential here concerns mistakenly associating the current dialogue with the wrong aircraft (i.e., focusing on a different aircraft from the one being handed off). This can occur if the controller has recently interacted with another aircraft and the procedure representing that aircraft may be highly activated. In such a case, it may send "confusing" signals during dialogue, thus potentially causing the controller to associate the handoff with a different aircraft.

Spec Sheet #5

Procedure: Identify a particular flight from the radar image

Relationship to Processes in Section 4: The procedure receives (as input) the individual features that identify a particular flight. If these inputs are received within a certain period of time, this procedure "fires," indicating that the particular aircraft has been detected. These are intermediate-level feature detectors as well as priming inputs from expectations associated with larger patterns of aircraft.

Relationship to ATC Scenario: The procedure is activated for flights when ATC first begins to scan the radar screen. It is then re-activated when a planned handoff is imminent.

Input

External

- Procedure detecting location of icon on radar screen
- Procedure detecting frequency of icon flashing
- Procedure detecting direction of movement on the radar screen

Internal

- Procedure representing "chunk" of aircraft that are "traffic for one another"
- In more expert ATC, output may go directly to the procedure that agrees to accept the handoff (e.g., in the case of AAL) or to a procedure that confirms that the aircraft are not on a collision course

Notes on Spec Sheet #5

Automaticity. The more expert a controller is, the less that controller will be aware of the individual primitive features that help to identify a particular flight, and the more the controller will be able to recognize the flight instantly. It is possible that this increased degree of expertise, associated with an increased level of automatic processing, may lead to some errors.

Error Potential. The error potential is that the set of features associated with one flight may be misinterpreted as another, which could lead to the wrong action being taken later in the scenario, or to the wrong "chunk" being activated as representing the current radar situation. This type of error can propagate further and cause serious problems if undetected. For example, the controller might mistakenly think that another flight is the American Airlines (AAL) flight that has already been dealt with and determine that there is no danger of a collision, when in fact the flight in question might be on a collision course with another aircraft.

Spec Sheet #6

Procedure: Detect converging aircraft

Relationship to Processes in Section 4: This is a complex pattern-matching process which involves cooperation among the scanning, memory, and cognitive processes described in Section 4. The cognitive and memory functions play an integral role in this process, both at the conscious and unconscious levels. Tight integration between reactive and proactive perception is required as the controller systematically scans the radar screen and/or flight strips for potentially conflicting aircraft, matching existing conflict patterns with current data.

Relationship to ATC Scenario: This procedure is invoked many times throughout a scenario. The frequency depends on the controller's expertise. It is invoked at least once initially to establish that the aircraft are not on a collision path. It is invoked at least once during the handoff to ensure the new aircraft will not conflict with existing aircraft. It is invoked when a given aircraft requests a change in altitude. As the degree of expertise grows, the procedures may be bypassed as the controller begins to associate flight paths with the symbolic representations of the particular flights (e.g., Procedure 5) and to do less of the actual processing of the flight's behavior.

Input

External

- Procedures representing specific flights
- Primitive features representing direction of icon movement

Internal

- Procedure 7: detecting collisions
- Specific chunks of flights that tend to converge

Output

- Procedure 7: detecting potential collisions

Notes on Spec Sheet #6

Automaticity. As controller expertise grows, this procedure will be more and more automated, to the point where the expert will no longer be aware of executing individual steps. Another important change that occurs as expertise increases is the ability of this procedure to process more than two aircraft at a time. The expert is able to process triples or even larger sets of aircraft in order to detect whether some subset of them is converging.

Spec Sheet #7a

Procedure: Detect pending collision - *expert performance*

Relationship to processes in Section 4: This is a complex pattern-matching process that involves cooperation among the scanning, memory, and cognitive processes described in Section 4. The cognitive and memory functions play an integral role here, both at the conscious and unconscious levels. Tight integration between reactive and proactive perception is required as the controller systematically scans the radar screen and/or flight strips for potentially conflicting aircraft, matching existing conflict patterns with current data.

Relationship to ATC Scenario: This procedure is invoked many times throughout the scenario. The frequency depends on the controller's expertise. It is invoked at least once initially to establish that aircraft are not on a collision path. It is invoked at least once during the handoff to ensure the new aircraft will not conflict with existing aircraft. It is also invoked when aircraft request a change in altitude.

Input

External

- Known patterns of aircraft icons and associated data blocks
- Individual uncorrelated icons and associated data blocks
- Velocities of relevant icons

Internal

- Fixed geometrical patterns of "collision courses"
  - same airway, head-on
  - same altitude, heading towards one another
  - same airway, same direction/different speeds
  - same "airway", changing altitude

Notes on Spec Sheet #7a

Error Potential. Because of the complexity of this procedure, there are many potential error types. One is the possibility that the wrong pattern will be identified, thereby resulting in an effort to perform collision avoidance for aircraft which are not on a collision course. Thus, the controller is distracted from dealing with actual problem aircraft. Another is the possibility that some specific pattern is highly active at the time and causes the controller to wrongly identify particular aircraft as part of a "collision chunk," again leading to inappropriate use of controller resources.

Spec Sheet #7b

Procedure: Detect pending collision (*novice performance*)

Relationship to processes in Section 4: This activity involves cooperation among the scanning, memory, and cognitive processes described in Section 4.

The primary differences between Spec Sheets 7a and 7b are the decreased degree of automaticity in 7b, the consequent slow down in the overall process, and the complexity of the patterns that can be detected as chunks.

Relationship to ATC Scenario: This procedure is invoked many times throughout the scenario, the frequency depends on the controller's expertise. It is invoked at least once initially to establish that the aircraft are not on a collision path. It is invoked at least once during the handoff to ensure the new aircraft will not conflict with existing aircraft. It also is invoked when a given aircraft requests a change in altitude.

Input

External

- Individual aircraft icons which activate their corresponding procedures
- Simple geometrical patterns of "collision courses"
  - same airway, head-on
  - same altitude, heading towards one another
  - same airway, same direction - different speeds
  - same "airway", changing altitude

Output

- Type of identified conflict
- Conflict resolution procedure

Notes on Spec Sheet #7b

The novice is, in general, more likely to perform errors during this process. An additional category of errors for the novice has to do with controlling the overall performance. Novices may forget what has already been done, be inefficient and repetitive in both the scanning and the cognitive processing, and, in general, use their resources in a suboptimal and inefficient manner.

6.0 IMPLEMENTING THE PSYCHOLOGICAL FRAMEWORK

The most critical aspect of the architecture discussed in this report relates to our concept of coordinated, purposeful behavior emerging from a set of individual, autonomous processes. From the point of view of implementation, the next challenge will be to build, initialize, and operate a model such that the behaviors it exhibits satisfy at least the following two requirements.

1. When given the inputs that are normally associated with a particular task, the outputs of the model will be correct and produced at an appropriate tempo.

2. When they are not correct and/or are not produced with the appropriate tempo, model outputs are "appropriately wrong"; that is, they are traceable to cognitive and attentional differences associated with given (simulated) levels of operator expertise.

In this section, we summarize several aspects of the procedural language that are expected to provide the basis for the successful implementation of the psychological framework presented in Section 4 and that hold considerable promise for meeting the behavioral requirements for the model.

6.1 Goals as the Basis for Modeling Proactive Human Behaviors

The psychological framework requires a view of goals as the basis for proactive human behaviors. A computational view of goals and their decomposition into subgoals and procedures is provided through the semantics of the Goal-Plan-Procedure (GPP) language (Abrett *et al.*, 1990). The concept of goal directed behavior as defined by GPP is close to that required by the psychological framework. In GPP, a *success-condition* defines the purpose of a goal. One represents a *plan* for achieving a goal as an *and/or* graph of subgoals and procedures, with temporal precedence explicitly represented among the subgoals and procedures making up the plan *pstop* and *one-of* are recent extensions to the language to express particular macro-like variations in behavior among the nodes of the *and/or* graph. These language extensions were introduced in the latest implementation of GPP, based on the Simulation CORE (SCORE) language (Deutsch & Palmucci, 1992). The ease with which GPP semantics and the recent extension were developed, based on SCORE, suggests that any further refinements will be easily accommodated.

6.2 Foundations for Multiple-Task Modeling

The modeling of multiple-task behaviors requires a formalism to represent the interactions between tasks contending to execute. The psychological framework suggests basing this competition on the accrual of activation credits vis-à-vis other tasks contending to execute. For example, in a resource based model, contention will exist among tasks competing for a portion of the resource. Similarly, in a model explicitly separating the conscious and nonconscious portions of tasks, there will be competition for attention among the conscious portions. That is, as tasks are developed, they naturally form *contention* classes—classes within which the competition to execute will need to be modeled.

Goals and procedures will be classified, with their definitions based on the Simple Frame Language (SFL) (Abrett *et al.*, 1990) concepts defined within a hierarchy. The ability to cluster goals and procedures along multiple dimensions is one factor supporting multiple-task modeling by representing relationships among the contending groups of goals or among the goals within a group. Further support for multiple-task modeling is provided by the dynamic priority associated with goals. Given the classification of goals, SCORE provides a basis for developing the protocols for implementing the explicit contention among the goals and procedures within the classes. The essential generic functions *maybe-suspend*, *suspend*, and *resume* are available for managing the suspension and resumption of individual goals and procedures. The *with-conflicts-mixin* provides one example of their use—a particular suspend-resume structure based on goal priority and classification. Further, a *suspend handler* enables a suspended task to dictate its own behavior during the period for which it is suspended—perhaps by simply adjusting the level of activation of the task over time. As the activation levels of tasks are adjusted over time, these primitive functions provide the basis for tasks to assert their right to execute vis-à-vis other tasks in their contention classes. A loss in activation level for an active task will open a window of opportunity for competing tasks, while a gain in activation level by a nonexecuting task will enhance its opportunity to execute. An additional macro, *atomic*, provides for establishing restart points when resuming a task. Particular forms of contention-to-execute will be developed and represented in appropriate procedure objects as needed to support the implementation of the psychological framework.

6.3 Multiple-Tasking and Emergent Behaviors in SCORE

The section above on foundations for multiple-task modeling suggests goal priorities as a basis for selecting among contending tasks. Tasks impacting the activation levels of other tasks suggests another dimension on which tasks might contend. Inhibition, as well as activation, must be looked at, since a running task will impact other running tasks and tasks contending to run, either positively or negatively.

As an example, *attention* can be viewed as a resource that tasks contend for: Either a task has it or it doesn't—there is no middle ground. A task accrues activation credits in a quantity sufficient to reach the point where it becomes an attended activity, and it pours out inhibitors suppressing the previously attended-to and currently competing nearly-attended-to tasks.

Reaction to events can drive attention. Similarly, a cognitive decision might promote a task to an attentional state which in turn inhibits other attention seeking tasks. Agent behaviors emerge from the dynamics of the activation levels of tasks.

There are two potential places to represent activation and inhibition in the SCORE implementation. The first is as an elaboration of the initial-conditions of goals; the second is as an elaboration of asynchronous-waits. In either case, the activation/inhibition representation will be explicit. Further ideas regarding task activation are discussed below.

6.4 Tasks on Agents as SCORE Procedures

SCORE procedures are a good basis from which to start implementation of a model of human task execution. They are invoked by being called with a set of arguments and then run either as subroutines of their calling procedure or independently and in parallel with their calling procedures. In the latter case, we say that the procedure is *spawned*. When called as subprocedures, they return to their calling procedure upon completion, returning *success* or *failure*, and possibly *values*, much like a Lisp function. Procedure invocation is all or nothing. In contrast with typical procedure invocation, many SCORE procedures are written to begin execution and immediately go into an asynchronous-wait for an event—best described as a "low" level of activation. At the goal level, initial-conditions perform a similar function in managing the execution of a goal. That is, in the SCORE simulator a goal may be activated, but it may have to meet additional requirements as dictated by its initial-conditions before it may continue execution. Generic tools are in place to accommodate several viewpoints on staging the activation level of tasks.

Procedures are executed by agents in a SCORE simulation, and an agent may execute more than one procedure at a time. A simulation object may be a single agent that executes a wide variety of procedures or a composite of agents running procedures.

6.5 Managing Parallel Task Execution

Language features to express the sequential and parallel execution of tasks are available at two levels of abstraction. At the SCORE level, the primitives *race* and *join* manage the execution of parallel subprocedures. Parallel procedures operating under *race* all complete when the first one completes, thereby allowing the execution of subsequent procedures. For *join*, each subprocedure must complete before subsequent procedures execute. Similar capabilities are available at the level of GPP semantics. GPP *goals* have *plans* defining their execution in an *and/or* tree with parallelism implicit at each level. Temporal precedence may be controlled explicitly by dictating sequential procedure execution where desired.

6.6 Reentrant Map Semantics Based on SCORE Procedures

Data driven task activation, with the data arriving from multiple sources, suggests a data-flow architecture. This approach finds neuroscience support in Gerald Edelman's (1987) development of the Theory of Neuronal Group Selection (TNGS), drawing on his ideas of *reentrant* and *global maps*. Here, SCORE procedures correspond roughly to reentrant map elements that are driven by multiple inputs and that generate outputs for multiple consumers. The data-flow in the network imposes sequentiality along the paths of its flow, while also facilitating parallel execution among other nodes.

The data-flow architecture suggests several changes surrounding SCORE procedure invocation. A SCORE procedure is very much like Lisp function: it is invoked from another procedure with a set of arguments. Data-flow suggests that all arguments need *not* come from a single source; that is, one procedure may provide some of the arguments to a subprocedure, but additional arguments from another procedure may be required before the subsequent data-flow node may execute. Breaking with a traditional data-flow architecture, not all the arguments will be required to be present for a procedure to execute. While data-flow ideas will be used in the development of the psychological framework, the formulation will not be strictly bound by them.

A SCORE procedure is also like a Lisp function in that it may return values to its calling procedure; an event that simply does *not* occur in data flow networks. Instead, a procedure is likely to make use of the SCORE *loop-forever* macro, sending out its results as events or further subroutine calls when it nears the completion of an iteration and returning to the top of its *loop-forever* to await its next activation. For present purposes, it will be useful to breadboard the management of the delivery and arrival of procedure arguments using basic SCORE functionality loosely modeled on the implementation of initial conditions for goals, and then to put in place data-flow-like macros to code this new SCORE extension more directly. As this and other issues are addressed, the implementation will evolve into a reentrant map semantics that supports the implementation of the psychological framework.

6.7 Task Activation

6.7.1 Activation and Inhibition in Initiating Task Execution

The primary questions regarding task activation and the contention between tasks are:

- What is the basis on which tasks apply for activation?; and
- How is arbitration to be handled when there is contention between tasks that are ready to execute and those that are already executing?

There will not be a superordinate manager to aıbitrate disputes; rather, procedures will compete to run by accruing activation credits. Behavior will be emergent rather than directed. The data, or arguments, required by a procedure are the primary means of establishing credits for execution; the arrival of input data is one form of activation. The question of whether activation or inhibition may be accrued independent of input data is still open. Possible roles for *pure* activation and inhibition are considered below. Additionally, more than one instance of a required argument may arrive before a procedure executes, which raises two additional questions: Which argument value is the procedure to execute with? How is this arbitrated?

6.7.2 Data as Activation for Tasks

A "data as activation" stance is somewhat akin to a connectionist network architecture in which links between nodes carry data, and each link has an associated *strength* or *weight*. The same datum may be used as input to different procedures with different weights. As data (the arguments to procedures) arrive, they may deliver activation credits as well as argument values that enhance the procedure's claim to execute vis-à-vis contending procedures. Important open questions here are: How will initial weights be established? How will they be modified during execution? These are issues in human and machine learning that should be addressed once some experience with model implementation has been accrued.

6.7.3 Task Activation and Inhibition Independent of Input Data

Activation associated with the delivery of input argument values to procedures has been considered. The possibility of delivering quanta of activation or inhibition to procedures independent of input data remains to be considered. Consideration of the contention for attention by tasks suggests a case for the use of inhibition. An attended task that begins execution might would generate inhibitory quanta directed at other tasks requiring attention to execute. In a sense, this would provide a newly activated attended-task with the inertia necessary to prevent rapid flip-flopping among tasks with very similar levels of activation.

The tasks contending for attention may be grouped together to form an SFL class, and it has been suggested that a newly attended task deliver a (perhaps computed) quanta of inhibition to each task in the class. Considerations of contention among tasks requiring attention may be sufficient to establish the case for *pure* activation or inhibition. On one hand, alternate ways to deal with inertia will be explored, while simultaneously looking for additional examples to bolster the case made in this report for activation in its pure form.

Conscious, proactive tasks have a requirement that motivates the role of pure activation—activation independent of data. An agent makes a decision to pursue or resume Task A rather than to initiate or to continue Task B, a decision that is effected by further activating Task A.

Task A may have met all its data requirements, but simply not yet received sufficient activation to displace the "current" attended task. Caution must be exercised here: The actual current attended task might be the task in which the decision is being made rather the "current" task.

A case can also be made for pure inhibition within the unattended portion of tasks. Following (Logan, 1988a; 1988b) considerable parallelism can be postulated in the pursuit of unattended tasks. That is, in automatic tasks, the search for a response is a race that goes on in parallel along many threads. The retrieval times for any particular thread form a distribution; hence, a particular solution may prevail on one occasion, but not on another. Using the first response to apply requires that all trailers be stopped; that is, a blanket of inhibition must deactivate the losers.

6.7.4 Task Activation with Missing or Redundant Data

The level of activation necessary to initiate task execution will accrue over time, leaving open the possibility of receiving more than one instance of an argument necessary to execute a task. However, it is also possible that a task may accrue enough activation credits to begin execution under circumstances in which all arguments have not been received. The issues raised here open a path for considering data-seeking behaviors and a possible role for feedback loops in the data-flow network.

The arrival of input data to a task over time immediately suggests that memory decay has the potential to intervene before task execution. Consider, for example, a verbal communication with a data item such as a requested altitude for an altitude change. During an interruption that prevents an immediate response by the controller, the requested altitude may be forgotten. Moments later the controller has the opportunity to respond to the request, but finds the key datum missing. Hence, the controller must proactively initiate a verbal task to request a repeat of the altitude information, that ultimately results in the feedback of the missing data as a new input. The controller is once again ready to execute the response task, this time with its full complement of input data.

Similarly, the delay in time between the receipt of data and the initiation of task execution can generate errorful performance. The intervening tasks might utilize "similar" data that is misappropriated for the resumed task, or previously supplied data may simply be degraded during the time between receipt and utilization.

9.0 GLOSSARY

6.8 Interfaces for Model Development

6.8.1 Flexibility/Parameter Tweaking: Developer's and Analyst's Handles

In order to fully understand the nature of the processing within the proposed Chaos architecture, it is important to provide both the developer and the analyst with a series of visualization tools and an easy means of modifying model parameters. During development, these parameters can be used to tune the model and to experiment with various implementation strategies. Since the parameters will correspond to both structural features and to details of dynamic behavior, analysts/developers will be able to experiment with a variety of specific model configurations to emulate the corresponding human attributes.

6.8.2 Modeling Parameters

The list below describes some of the modeling parameters:

Memory/Schema/Internal Model Parameters

- accessibility of some data/knowledge;
- frequency of access to a particular piece of data;
- accuracy of internal model - both static (e.g., "attribute values of a "schema") and, when the model represents a dynamic entity, the model accuracy with respect to the behavior over time;
- stability of internal models - how "established" a particular representation is; how easily it is interfered with;
- speed with which internal models can be created - individual schemas grouped into larger structures; and
- flexibility of integrating a number of overlapping task features - i.e., check same altitude; different airspeeds; same airway, etc.

Control Parameters

These parameters modify features of the control knowledge applied to the task performance--the control knowledge may be domain-specific or domain-independent.

- subjective assessment of task difficulty;
- subjective assessment of own skill level (with respect to a particular task); and
- assessment of risk associated with inaccurate task performance;

Meta-Cognitive Parameters

These parameters modify the subjective assessment of the operators' performance and overall confidence in their ability to perform the tasks.

- confidence factor on other parameters (if highly confident, may tolerate some discrepancies; if not confident, may require too much "certainty match" before committing to a decision, recognition, etc.); and
- subjective "familiarity" factor ("It FEELS like I've seen this before . . .").

Attention/Consciousness Parameters

These parameters will control the functionality of the attention and conscious processes (e.g., the degree of conscious awareness required for performance).

Network-Level Parameters

These parameters will control properties of the network as a whole (e.g., stability/persistence of activation patterns, and speed with which contexts can be switched).

6.8.3 Visualizing Model Performance

As with the parameters, the ability to visualize both the internal structures and dynamics of the model, and to graphically display statistical data about the observable model performance is important—both for the developer and for the analyst. Below are examples of the types of visualization and statistical analysis tools that could be provided within the modeling environment.

- Structure of procedure network, including ability to filter by specific modalities (e.g., sensory or motor procedures); content (e.g., collision avoidance procedures, hand-off procedures, control procedures etc.);
- Dynamic behav·or of a procedure network during particular task execution;
- Frequency of activation of a particular procedure or set of procedures over time;
- Correlation of such activation frequencies across several types of procedures;
- Tracking of the evolution of a procedure "chunk" over time; and
- Once an internal model of some phenomenon has been established (e.g., a model of a particular aircraft in flight), the ability to display the physical analog of this model (reverse mapping) (i.e., the ability to see both what is actually out there and what the operator THINKS is out there, as determined by the operator's internal model).

7.0 THE PROTOTYPING EFFORT

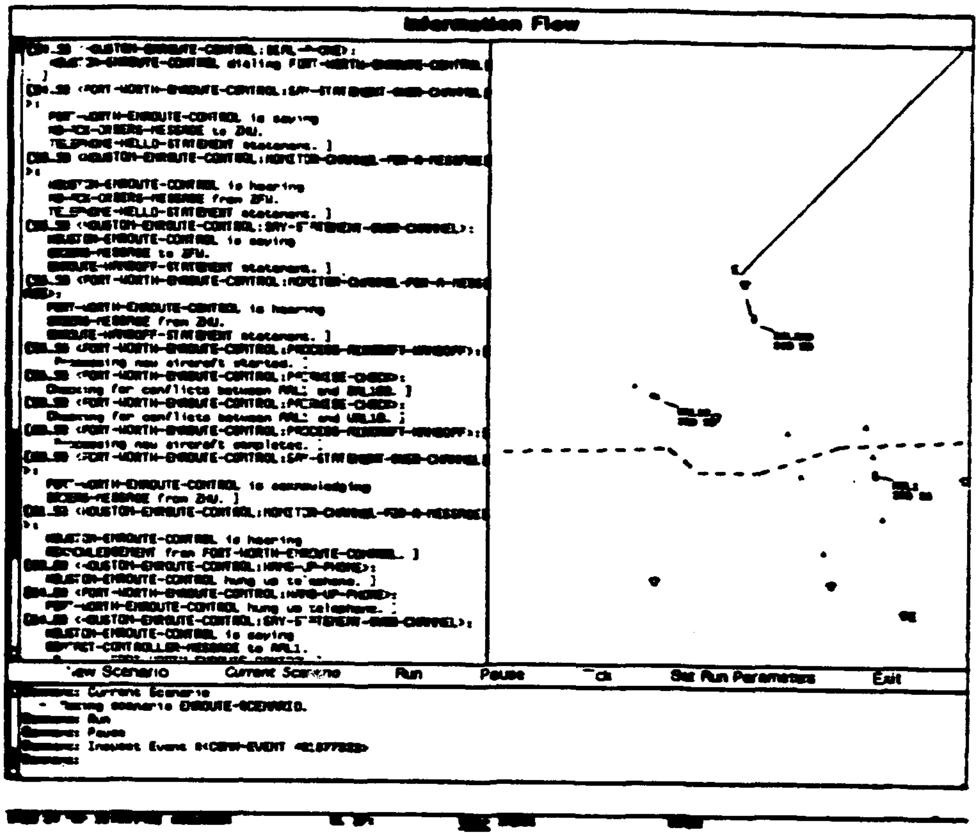
In discussing the psychological framework, it was argued that multitasking is essential to expert performance and that the analysis of multitask performance is central to developing an HPP model. In addition, it was suggested that the ability to emulate a range of operator skill levels and the performance of errors associated with them represented important ingredients in a model that might later be employed in a systems' operability testbed. The effort to incorporate these various elements within a single framework has proven to be synergistic in the sense that pursuing any one of them has tended to uncover material relevant to the others.

Once an initial version of the psychological framework was in place, a parallel effort was initiated to examine issues related to the implementation of a model based on the framework. As reported in the previous two sections, the intent was not a full-blown design effort, but rather a first look at implementation issues with a subordinate goal of providing feedback on the psychological framework. The analysis was then taken one step further by selecting critical implementation areas for early prototyping. The selection of target areas for prototyping and the results of the prototyping effort are reported in this chapter.

7.1 The Scenario for the Prototype

The domain selected for modeling was en route ATC. The particular agents focussed on for the prototype effort are the en route controllers operating at the ARTCCs at Dallas and Houston. Additional agents modeled are the flight-deck crews for the aircraft in the airspace. Modern, glass-cockpit aircraft with two-person crews (a Captain and a First-officer) populate the airspace.

Figure 10 provides a screen image of the airspace. The heavy dashed line is the boundary between the Dallas airspace to the north and the Houston airspace to the south. Two aircraft are in the Dallas airspace, United Airlines (UAL) 10 and Delta Airlines (DAL) 100, and one aircraft, AAL1, is in the Houston airspace. The prototype effort implements ATC and flight crew tasks conducting the handoff of AAL1 from the Houston en route controller to the Dallas en route controller. The process is initiated by the Houston controller placing a telephone call to the Dallas controller requesting the transfer of AAL1 to the Dallas airspace. The Dallas en route controller checks for airspace conflicts with aircraft currently in the Dallas airspace and, in this case, accepts the new aircraft. The Houston controller then contacts AAL1 to inform its crew of the Dallas controller's radio frequency and to ask the crew to contact the Dallas en route controller.



```
[illegible] <HOUSTON-ENROUTE-CONTROL:[illegible]>:
   HOUSTON-ENROUTE-CONTROL dialing FORT-WORTH-ENROUTE-CONTROL
. ]
[illegible] <FORT-WORTH-ENROUTE-CONTROL:SAY-STATEMENT-[illegible]-CHANNEL
>:
   FORT-WORTH-ENROUTE-CONTROL is saying
   NO-[illegible]-ORDERS-MESSAGE to ZHU.
   TELEPHONE-HELLO-STATEMENT statement. ]
[illegible] <HOUSTON-ENROUTE-CONTROL:MONITOR-CHANNEL-FOR-A-MESSAGE
>:
   HOUSTON-ENROUTE-CONTROL is hearing
   NO-[illegible]-ORDERS-MESSAGE from ZFW.
   TELEPHONE-HELLO-STATEMENT statement. ]
[illegible] <HOUSTON-ENROUTE-CONTROL:SAY-STATEMENT-[illegible]-CHANNEL>:
   HOUSTON-ENROUTE-CONTROL is saying
   [illegible]-MESSAGE to ZFW.
   ENROUTE-HANDOFF-STATEMENT statement. ]
[illegible] <FORT-WORTH-ENROUTE-CONTROL:MONITOR-CHANNEL-FOR-A-MESS
AGE>:
   FORT-WORTH-ENROUTE-CONTROL is hearing
   [illegible]-MESSAGE from ZHU.
   ENROUTE-HANDOFF-STATEMENT statement. ]
[illegible] <FORT-WORTH-ENROUTE-CONTROL:PROCESS-AIRCRAFT-HANDOFF>:
    Processing new aircraft started. ]
[illegible] <FORT-WORTH-ENROUTE-CONTROL:[illegible]-CHECK>:
   Checking for conflicts between AAL[illegible] and [illegible]. ]
[illegible] <FORT-WORTH-ENROUTE-CONTROL:[illegible]-CHECK>:
   Checking for conflicts between AAL[illegible] and UAL[illegible]. ]
[illegible] <FORT-WORTH-ENROUTE-CONTROL:PROCESS-AIRCRAFT-HANDOFF>:
    Processing new aircraft completed. ]
[illegible] <FORT-WORTH-ENROUTE-CONTROL:SAY-STATEMENT-[illegible]-CHANNEL
>:
   FORT-WORTH-ENROUTE-CONTROL is acknowledging
   [illegible]-MESSAGE from ZHU. ]
[illegible] <HOUSTON-ENROUTE-CONTROL:MONITOR-CHANNEL-FOR-A-MESSAGE
>:
   HOUSTON-ENROUTE-CONTROL is hearing
   ACKNOWLEDGEMENT from FORT-WORTH-ENROUTE-CONTROL. ]
[illegible] <HOUSTON-ENROUTE-CONTROL:HANG-UP-PHONE>:
   HOUSTON-ENROUTE-CONTROL hung up telephone. ]
[illegible] <FORT-WORTH-ENROUTE-CONTROL:HANG-UP-PHONE>:
   FORT-WORTH-ENROUTE-CONTROL hung up telephone. ]
[illegible] <HOUSTON-ENROUTE-CONTROL:SAY-STATEMENT-[illegible]-CHANNEL>:
   HOUSTON-ENROUTE-CONTROL is saying
   CONTACT-CONTROLLER-MESSAGE to AAL1.
```

```
[illegible]: Current Scenario
 - [illegible] scenario ENROUTE-SCENARIO.
[illegible]: Run
[illegible]: Pause
[illegible]: Inspect Event #<[illegible]-EVENT [illegible]>
[illegible]:
```

Figure 10
On-Line Scenario View

As the scenario evolves, three levels of verbal communication among controllers and flight crew members develop: the telephone conversation between en route controllers, the party-line radio conversation between controllers and the flight crews of the aircraft for which each has responsibility, and the in-person conversations between crew members on the flight deck. The simulation environment used as a baseline for the prototype included only approach and tower controllers; hence, it was necessary to add en route controllers and their telephone communication capability. At the present time, the baseline simulation deals with contention on the radio party line but does *not* properly handle contention between in-person and radio conversations. In the prototype, the new phone conversations are internally consistent, but it was beyond the scope of this effort to fully work out the possible conflicts generated by the three modes of conversation.

Additional details in the display portray the Very High Frequency Omni Range Tactical (Air Navigation) (VORTAC) radio beacons, the optional and required reporting points making up the flight plans for the aircraft, and the radar sweep which updates the display of the aircraft. The aircraft displays show the flight designators, the altitude in hundreds of feet, and the airspeed in tens of knots per hour. The "E"s are icons for the two en route controllers. The panel on the left provides a running trace of selected events of the simulation.

7.2 Target Areas for Prototyping

The implementation of the psychological framework will require a large suite of computational tools. Some of them, such as frame languages and rule-based languages, are reasonably mature and available in various forms from our basic AI tool boxes. Others, such as procedure languages and their simulators, are more specialized and less widely available. The environment used for prototyping was composed of the frame language VSFL, the procedure language SCORE, and the SCORE simulator.

In the prototyping effort, concentration centered on particular areas considered essential to the implementation of the key features of the psychological framework not previously modeled (i.e., those that used existing features in new ways or demanded basic changes to the larger modeling components as described above). The areas prototyped were task *conflict resolution* in multitasking agents, matching task selection to agent *performance levels*, and the *data flow* aspects of the semantics of reentrant maps (described in Section 6 as reentrant net semantics).

7.2.1 Conflict Resolution for Multitasking

Multitasking is a critical human capability that must be represented in an HPP model. Several approaches may be taken in implementing a multitasking capability. The first-order distinction is between resource models that apportion a capability across two or more tasks and time-slicing models that give an entire resource to an active task. The tool-building perspective being developed in this effort attempts to provide the capability to model either of these approaches or perhaps a third, as yet unspecified, approach.

The prototype example is based on the response of the Dallas en route controller to the request from the Houston controller to accept another aircraft into the Dallas airspace. This verbal request is initiated by a telephone call from the Dallas controller. The key message is the ENROUTE-HANDOFF-STATEMENT, which includes the flight designator of the new aircraft. The message is fielded by the MONITOR-CHANNEL-FOR-A-MESSAGE procedure, executed by the Dallas controller. Receipt of the message by the Dallas controller initiates two actions; it is these actions that must be mediated. The first is concerned with generating the action that is required by a particular message type; the second is to reply to the message if a reply is necessary. These could conceivably occur in parallel, which is the default behavior. In the particular case used for the prototype, that behavior is not acceptable—the Dallas controller must check for conflicts between the new aircraft and those already in the airspace before responding to the Houston controller.

The details are complicated, but have been included for completeness. MONITOR-CHANNEL-FOR-A-MESSAGE eventually calls PROCESS-ORDER, which is specialized on the statement type, leading to an action that (see Section 7.2.3) frees up the paused procedure WAIT-FOR-HANDOFF-STATEMENT. WAIT-FOR-HANDOFF-STATEMENT in turn calls PROCESS-AIRCRAFT-HANDOFF, in which the Dallas controller evaluates the potential for conflicts in the airspace. Meanwhile, MONITOR-CHANNEL-FOR-A-MESSAGE returns the new message to MONITOR-CHANNEL, and MONITOR-CHANNEL uses CHOOSE-RESPONSE to generate the response to be used for a reply. To preserve the proper sequencing of actions, MONITOR-CHANNEL must be interrupted until PROCESS-AIRCRAFT-HANDOFF has been completed. To accomplish this, SPR::CONFLICTS-WITH-PROCEDURE was included in the definition of PROCESS-AIRCRAFT-HANDOFF, and the CONFLICT-WITH? method was defined to represent the conflict with MONITOR-CHANNEL, thus forcing the suspension of MONITOR-CHANNEL when PROCESS-AIRCRAFT-HANDOFF executed.

The particular conflict resolution strategy for tasks used in the prototype code is based on task priority and the class structure of the procedures involved. It was implemented using a generic set of SCORE functions provided as a basis for developing conflict resolution strategies. This generic capability will provide the ability to tailor particular conflict resolution strategies to particular multitasking approaches.

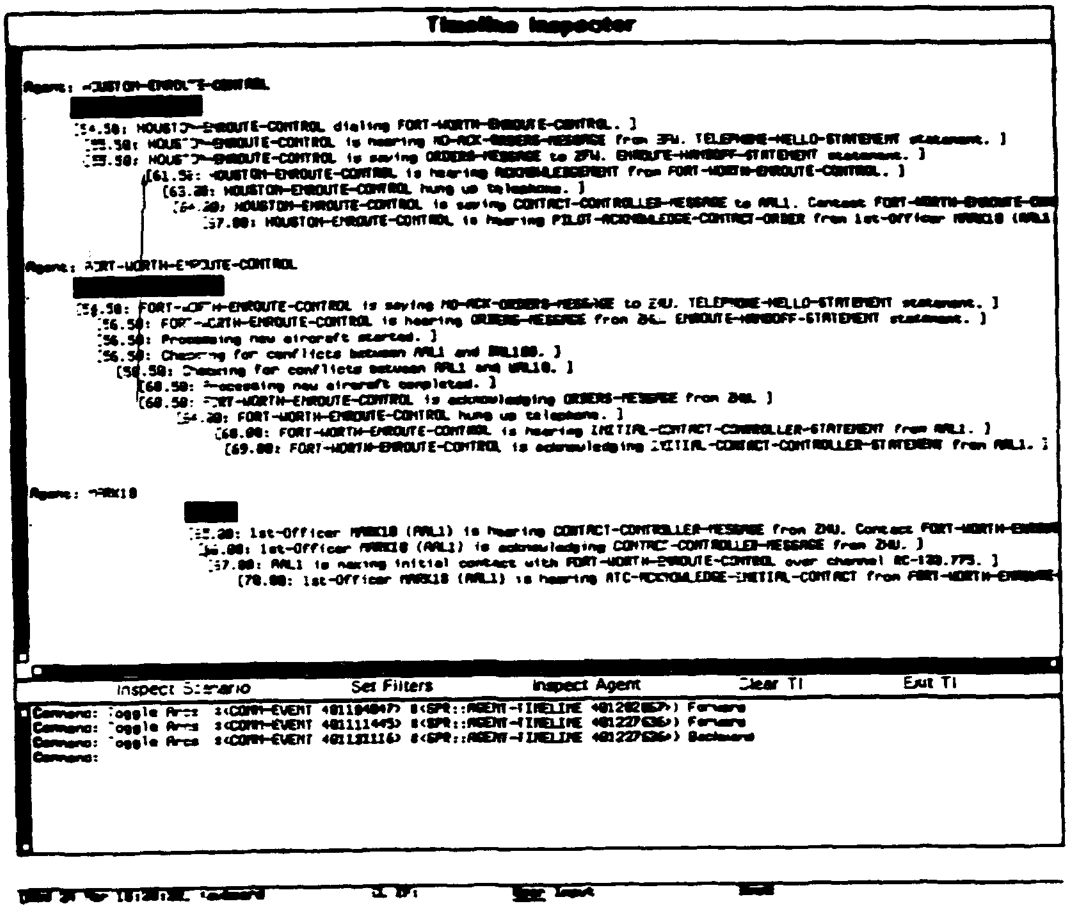
7.2.2 Matching Task Execution to Agent Performance Levels

Different human agents perform at different skill levels based on their individual ability and experience. We wish to capture this range of performance in the tasks that we build in modeling ATC operators, and then to select particular tasks appropriate to the skill level of the agent about to execute them. In the particular example from the scenario described above, the Dallas controller is called upon to accept a new aircraft into the Dallas airspace. Here, it was speculated that, early in training, a novice en route controller might need to execute a pairwise comparison of the new aircraft with each aircraft already in the airspace. In contrast with the novice controller, in a light traffic situation an expert en route controller might be able to "instantly" assess conflicts caused by the new aircraft. The purpose of the prototype effort was to establish a mode of task selection, rather than the validity of the particular algorithms used.

Task selection by skill level is actually a particular refinement of the preference semantics capability recently added to SCORE; hence, it was implemented within this framework. The SCORE special form *oneof* encloses a collection of procedure calls, only one of which is to be executed. The selection of the particular procedure to execute is done by a function identified within the *oneof* form. Here, the function PROCEDURE-SELECTION-BY-SKILL was implemented to find a match between the agent's skill level and the skill levels required by the enclosed procedures. In this particular case, the two candidate procedures were PAIRWISE-HANDOFF-

CHECK and GESTALT-HANDOFF-CHECK, representing novice and expert behaviors, respectively. The definition of each of these procedures included a Very Simple Frame Language (VSFL) concept, SKILL-LEVEL-MIXIN, to provide a slot in which to denote the skill level required by an agent to execute the procedure. The definition of an en route controller similarly included the concept, SKILL-LEVEL-SELECT-MIXIN, through which to define the agent's skill level. This concept also provided the capability to interactively set the skill level for each controller.

Figures 10 and 11 present two views of the handoff dialogue between the Houston and Dallas controllers, where the Dallas controller is an expert. SCORE *events* mark the beginning and end of the expert, or "Gestalt," conflict check done by the Dallas controller begun at 37.50 and completed at 38.50. Figure 10 details the on-line trace of these events; Figure 11 shows the time line presentation. If the Dallas controller were a novice, the start and stop events would enclose a longer period of time with two separate executions of PAIRWISE-HANDOFF-CHECK (one instance executing for each aircraft currently in the airspace).



```
Agent: HOUSTON-ENROUTE-CONTROL

  [54.50: HOUSTON-ENROUTE-CONTROL dialing FORT-WORTH-ENROUTE-CONTROL. ]
   [55.50: HOUSTON-ENROUTE-CONTROL is hearing NO-ACK-ORDERS-MESSAGE from ZFW. TELEPHONE-HELLO-STATEMENT statement. ]
   [55.50: HOUSTON-ENROUTE-CONTROL is saying ORDERS-MESSAGE to ZFW. ENROUTE-HANDOFF-STATEMENT statement. ]
        [61.50: HOUSTON-ENROUTE-CONTROL is hearing ACKNOWLEDGEMENT from FORT-WORTH-ENROUTE-CONTROL. ]
         [63.20: HOUSTON-ENROUTE-CONTROL hung up telephone. ]
          [64.20: HOUSTON-ENROUTE-CONTROL is saying CONTACT-CONTROLLER-MESSAGE to AAL1. Contact FORT-WORTH-ENROUTE-C
            [67.00: HOUSTON-ENROUTE-CONTROL is hearing PILOT-ACKNOWLEDGE-CONTACT-ORDER from 1st-Officer MARK18 (AAL1

Agent: FORT-WORTH-ENROUTE-CONTROL

  [54.50: FORT-WORTH-ENROUTE-CONTROL is saying NO-ACK-ORDERS-MESSAGE to ZHU. TELEPHONE-HELLO-STATEMENT statement. ]
   [56.50: FORT-WORTH-ENROUTE-CONTROL is hearing ORDERS-MESSAGE from ZHU. ENROUTE-HANDOFF-STATEMENT statement. ]
   [56.50: Processing new aircraft started. ]
   [56.50: Checking for conflicts between AAL1 and AAL100. ]
    [58.50: Checking for conflicts between AAL1 and UAL10. ]
     [60.50: Processing new aircraft completed. ]
     [60.50: FORT-WORTH-ENROUTE-CONTROL is acknowledging ORDERS-MESSAGE from ZHU. ]
       [64.20: FORT-WORTH-ENROUTE-CONTROL hung up telephone. ]
         [68.00: FORT-WORTH-ENROUTE-CONTROL is hearing INITIAL-CONTACT-CONTROLLER-STATEMENT from AAL1. ]
          [69.00: FORT-WORTH-ENROUTE-CONTROL is acknowledging INITIAL-CONTACT-CONTROLLER-STATEMENT from AAL1. ]

Agent: MARK18

          [65.20: 1st-Officer MARK18 (AAL1) is hearing CONTACT-CONTROLLER-MESSAGE from ZHU. Contact FORT-WORTH-ENROUTE
           [66.00: 1st-Officer MARK18 (AAL1) is acknowledging CONTACT-CONTROLLER-MESSAGE from ZHU. ]
            [67.00: AAL1 is making initial contact with FORT-WORTH-ENROUTE-CONTROL over channel RC-130.775. ]
             [70.00: 1st-Officer MARK18 (AAL1) is hearing ATC-ACKNOWLEDGE-INITIAL-CONTACT from FORT-WORTH-ENROUTE-
```

```
Command: Toggle Arcs  #<COMM-EVENT 401104047> #<SPR::AGENT-TIMELINE 401202057>) Forward
Command: Toggle Arcs  #<COMM-EVENT 401111445> #<SPR::AGENT-TIMELINE 401227636>) Forward
Command: Toggle Arcs  #<COMM-EVENT 401131116> #<SPR::AGENT-TIMELINE 401227636>) Backward
Command:
```

Figure 11

Timeline Scenario View

7.2.3 Reentrant Nets as Data Flow

The last prototype target is probably the most interesting. It brings together Edelman's (1987) neurobiological formulation of reentrant and global maps and data-flow approaches to computation. The idea put forward in the psychological framework and further refined in the discussion of the computational implementation of the framework is to represent the reentrant maps as a network of SCORE procedures. Unlike more traditional Lisp functions, network nodes (or procedures) were posited to have inputs from multiple sources as in a data-flow network. It was also suggested that the ability to specify a subset of the arguments as optional inputs should be retained. Optional arguments are standard features of Lisp functions, but not generally features of data-flow networks. This prototype example was developed to implement this style of procedure activation.

Data flow task activation was used in the WAIT-FOR-HANDOFF-STATEMENT (Figure 12) procedure executed by the Dallas en route controller. As outlined in Section 7.2.1, the incoming data (i.e., the new aircraft about to enter the sector) is provided by PROCESS-ORDER using the SCORE function *signal-event.* As shown in the code in Figure 12 the signal, RECEIVING-NEW-AIRCRAFT-EVENT, is fielded by the *with-signal* function which assigns values to the variables ?WHO and ?MODEL. ?WHO is a discriminator designating the Dallas controller as the target for the event. The aircraft argument is ?MODEL, designating the Dallas controller's private view of the actual aircraft (e.g., the Dallas controller knows the last position of the aircraft as painted on the radar screen, but not its actual position in the airspace right now). Any additional arguments required by the procedure would be similarly included in the *join* statement. The subsequent *join* statement manages the receipt of the optional arguments. The *signal-event* following the *join* for the required arguments together with the *with-signal* in the *race* of the optional arguments free up the body of the procedure for execution when all the required arguments have been received.

Figure 12 illustrates the form of the code as it was developed for the prototype example. For the data flow initiation of the procedure, the salient items are the required and optional arguments. To simplify the use of this functionality, a macro will be developed to hide the details elaborated in this figure, making the code easier to generate and understand. In addition, the implementation will be reviewed with the goal of improving its efficiency.

```
(defproc WAIT-FOR-AIRCRAFT-HANDOFF (en route-atc-procedure)
    0
 (with-slots (agent) self
  (loop-forever
   (let ((handoff-model nil)
         (optional-arg1 nil)
         (optional-arg2 nil))
    (join
     (sequentially
      (join
       (with-signal ((receiving-new-aircraft-event ?who ?model)
               :test (eql ?who agent))
              (setf handoff-model ?model)))
      (signal-event `(data-flow ,self)))
     (race
      (join
       (with-signal ((optional-arg ?optional-arg1))
              (setf optional-arg1 ?optional-arg1))
       (with-signal ((optional-arg ?optional-arg2))
              (setf optional-arg2 ?optional-arg2)))
      (asynch-wait `(data-flow ,self))))
    (process-aircraft-handoff :priority 20 :handoff-model handoff-model)))))
```

Figure 12

An Example of a Data Flow Node

8.0 REFERENCES

Abrett, G., Deutsch, S.E., Downes-Martin, S. (1990). Two AI Languages for Simulation. *Transactions of the Society for Computer Simulation Special Issue on Artificial Intelligence and Simulation,* 7(3).

Adams, M.J., Tenney, Y.J., & Pew, R.W. (1991). *State-of-the-Art-Report: Strategic Workload and the Cognitive Management of Advanced Multi-Task Systems.* Wright-Patterson Air Force Base, OH: Crew System Ergonomics Information Analysis Center (CSERIAC) Publication Series.

Allen, J., & Perrault, C. (1980). Analyzing intentions in utterances. *Artificial Intelligence, 15,* 143-178.

Anderson, J. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Barsalou, L., & Sewell, D.R. (1985). Contrasting the representations of scripts and categories. *Journal of Memory and Language, 24,* 646-665.

Bower, G.H., Black, J.B., & Turner, T.J. (1979). Scripts in memory for text. *Cognitive Psychology, 11,* 177-220.

Cheng, P.W. (1985). Restructuring versus automaticity: Alternative accounts of skill acquisition. *Psychological Review, 92,* 414-423.

Churchland, P.S, & Sejnowski, T.J. (1990). Foundations of cognitive. In M. Posner (Ed.), *Brain and Cognition* (pp. 301-356). Cambridge, MA: The MIT Press.

Cohen, J.D., Dunbar, K., & McClelland, J.L. (1990). On the control of automatic processes: A parallel distributed processing account of the stroop effect. *Psychological Review, 97,* 332-361.

Corker, K.M., & Cramer, N.L. (1989a). Methodology for Evaluation of Automation Impacts on Tactical Command and Control (C2) Systems: Domain Selection and Approach. (AFHRL-TR-89-17). Brooks AFB, TX: AFSC.

Corker, K.M. & Cramer, N.L. (1989b). Methodology for Evaluation of Automation Impacts on Tactical Command and Control (C2) Systems: Implementation (AFHRL-TR-90-9). Brooks AFB, TX: AFSC.

Corker, K.M., & Cramer, N.L. (1991). Methodology for Evaluation of Automation Impacts on Tactical Command and Control (C2) Systems. (AFHRL-TR-91). Brooks AFB, TX: AFSC.

Craik, F.I.M., & Lockhart, R.S. (1972). Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behavior, 11*, 671-684.

Crick, F., & Koch, C. (1990). Towards a neurobiological theory of consciousness. *Neurosciences, 2*, 263-275.

Damsio, A.R. (1989a). The brain builds entities and events by multi-regional activation from convergence zones. *Neural Comp., 1*, 123-132.

Damsio, A.R. (1989b). Time-locked multi-regional retroactivation: A systems-level proposal for the neural substrates of recall and recognition. *Cognition, 33*, 25-62.

Dennett, D. (1991). *Consciousness explained.* Boston, MA: Little, Brown, and Company.

Deutsch, S.E., Palmucci, J. (1992). *Information Flow - Interim Report.* BBN Technical Report No. 7687, Bolt Beranek and Newman, Cambridge, MA.

Eckhorn, R., Bauer, R., Jordan, W., Brosch, M., Kruse, W., Munk, M., & Reitboeck, H.J. (1988). Coherent oscillations: A mechanism of feature linking in the visual cortex? *Biol. Cyber., 60*, 121-130.

Edelman, G.M. (1987). *Neural darwinism: The theory of neuronal group selection.* New York: Basic Books.

Edelman, G.M. (1989). *The remembered present: A biological theory of consciousness.* New York: Basic Books.

Geddes, N.D. (1989). *Understanding human operators' intentions in complex systems.* Unpublished doctoral dissertation, Georgia Institute of Technology, Atlanta, GA.

Hintzman, D.L. (1986). "Schema abstraction" in a multiple-trace memory model. *Psychological Review, 93*, 411-428.

Jackendoff, R. (1987). *Consciousness and the computational mind.* Cambridge, MA: The MIT Press.

Kahneman, D. (1973). *Attention and effort.* Englewood Cliffs, NJ: Prentice Hall.

Kant, I. (1781/1952). *The critique of pure reason.* In J.M.D. Meiklejohn and R. M. Hitchins (Eds.), *Great books of the western world* (pp. 1-252). Chicago: Encyclopedia Britannica.

Logan, G.D. (1988a). Toward an instance theory of automatization. *Psychological Review, 95*, 492-527.

Logan, G.D. (1988b). Automaticity, resources, and memory: Theoretical controversies and practical implications. *Human Factors, 30*, 583-598.

Marcel, A.J. (1983). Conscious and unconscious perception: An approach to the relations between phenomenal experience and perceptual processes. *Cognitive Psychology, 15*, 238-300.

Means, B., Mumaw, R., Roth, C., Schlager, M.A., McWilliams, E., Gagne, E., Rice, V., Rosenthal, D. & Heon, S. (1988). *ATC Training Analysis Study. Design of the Next-Generation ATC Training System.* HumRRO International, Inc., OPM Work Order 342-036.

Miller, G.A., Galanter, E., & Pribram, K.H. (1960). *Plans and the structure of behavior.* New York: Holt, Rinehart, & Winston.

Navon, D. (1985). Attention division or attention sharing. In M.I. Posner and Oscar S. M. Marin (Eds.), *Attention and performance XI* (pp. 133-146). Hillsdale, NJ: Lawrence Erlbaum Associates.

Neisser, U. (1976). *Cognition and reality: Principles and implications of cognitive psychology.* San Francisco: W. H. Freeman.

Newell, A. (1990). *Unified theories of cognition.* Cambridge, MA: Harvard University Press.

Newell, A., & Simon, H.A. (1963). GPS, a program that simulates human thought. In E.A. Feigenbaum and J. Feldman (Eds.), *Computers and thought* (pp. 279-293). New York: McGraw-Hill.

Pashler, H., & Johnston, J.C. (1989). Chronometric evidence for central postponement in temporally overlapping tasks. *Quarterly Journal of Experimental Psychology*, 41A, 19-45.

Reason, J. (1983). *Absent-mindedness and Cognitive Control.* London: Academic Press.

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel distributed processing, volume 1.* Cambridge, MA: The MIT Press.

Rumelhart, D.E., Smolensky, P., McClelland, J.L., & Hinton, G.E. (1988). Schemata and sequential thought processes in PDP modles. In J.L. McClelland and D.E. Rumlehart (Eds.), *Parallel distributed processing* (pp. 7-57). Cambridge, MA: The MIT Press.

Schank, R.C., & Abelson, R. (1977). *Scripts, plans, goals, and understanding*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Schneider, W. (1985). Toward a model of attention and the development of automatic processing. In M I. Posner and O.S.M. Marin (Eds.), *Attention and performance XI* .(pp. 475-492). Hillsdale, NJ: Lawrence Erlbaum Associates.

Shiffrin, R.M., & Schneider, W. (1977). Controlled and automatic information processing: II. Perceptual learning, automatic attending, and a general theory. *Psychological Review, 84*, 127-189.

Thibadeau, R. (1986). Artificial perception of actions. *Cognitive Science, 10*, 117-149.

Wickens, C.D. 1984. Processing resources in attention. In R. Parasuraman & R. Davies (Eds.), *Varieties of attention* (pp. 63-101). New York: Academic Press.

Yates, J. (1985). The content of awareness is a model of the world. *Psychological Review, 92*, 249-284.

9.0 GLOSSARY

AAL	American Airlines
ACT	Adaptive Character of Thought
AI	Artificial Intelligence
AIRT	Automation Impacts Research Testbed
ARTCC	Air Route Traffic Control Centers
ATC	Air Traffic Control
BBN	Bolt Beranek and Newman, Inc.
CRC	Control and Reporting Center
CRT	Cathrode Ray Tube
DAL	Delta Airlines
FAA	Federal Aviation Administration
GOMS	Generic Operator Modeling System
GPP	Goal-Plan-Procedure (Language)
HPP	Human Performance Process
MCE	Modular Control Equipment
RDT&E	Research Development, Training, and Evaluation
SCORE	Simulation CORE
SFL	Simple Frame Language
SOSUS	Sound Surveillance System
TNGS	Theory of Neuronal Group Selection
UAL	United Airlines
VORTAC	Very High Frequency Omni Range Tactical (Air Navigation)
VSFL	Very Simply Frame Language

U S G P O 1993-550-057/81024