N89-20694

Use of Artificial Intelligence in Supervisory Control

Human-machine **mix** where artificial intelligence, advanced automation, robotics, and human supervisory control are integrated in an effective human-machine system.

Aaron Cohen & Jon D. Erickson

Johnson Space Center
Advanced Technology Advisory Committee
NASA Technical Memorandum, April 1985

Major Research Issue

How to use artificial intelligence in system control?

Replace human operator

or

Amplify human operator's abilities to monitor the system and detect, diagnose and compensate for system failures?

Objective of OFMspert Research

Design an architecture to provide the human operator with an *intelligent* decision support system

-- to augment, not replace, the versatile human skills with skills provided by machine intelligence.

-- to compensate for known human limitations.

-- to complement individual human preferences

Develop a *theory* of human-computer interaction in the control of complex, dynamic systems (normative, plausible)

Build a *model* of the theory to demonstrate and empirically evaluate the proposed architecture (operator's associate)

Requirements for an

Intelligent Operator's Associate

Operator's Associate must provide information and control abilities at the right time, of the right kind and with the ease of a human associate

- -- understanding
- -- control
- -- interface

Understanding requires a model of the operator and system that can allow the OA to *infer* the operator's current control goals given knowledge of the control task, system functions, and current control state.

OFMspert Characteristics

OFMspert (operator Function Model Expert System) is an intelligent operator's associate based on the operator function model (Mitchell, 1987)

OFMspert uses the Operator Function Model (OFM) to represent knowledge about the operator

OFMspert uses the blackboard model of problem solving (Nii, 1986) to maintain a dynamic representation of operator *goals*, *plans, tasks,* and *actions* given previous operator actions and current system state

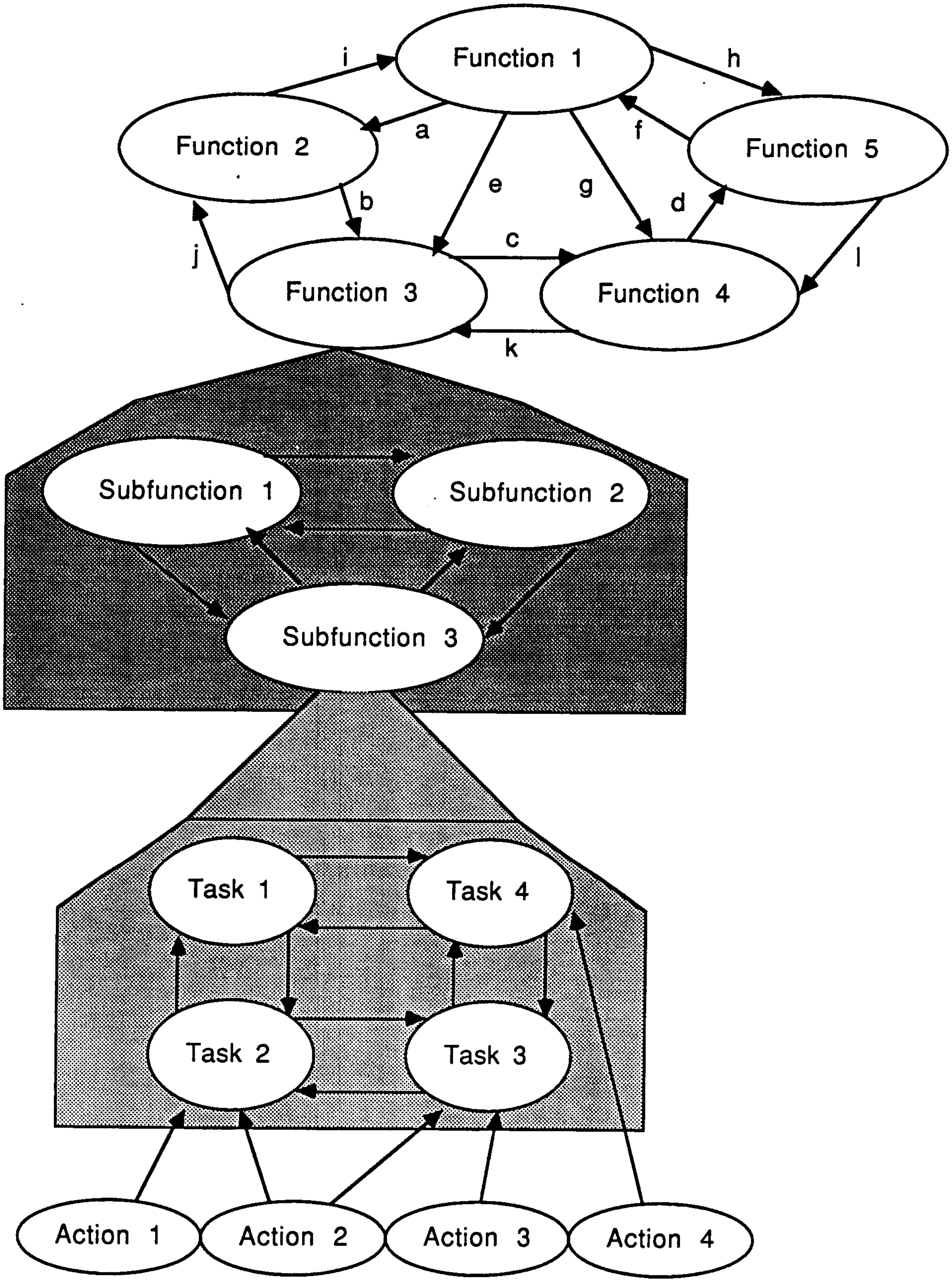


Figure 1. A Generic Operator Function Model

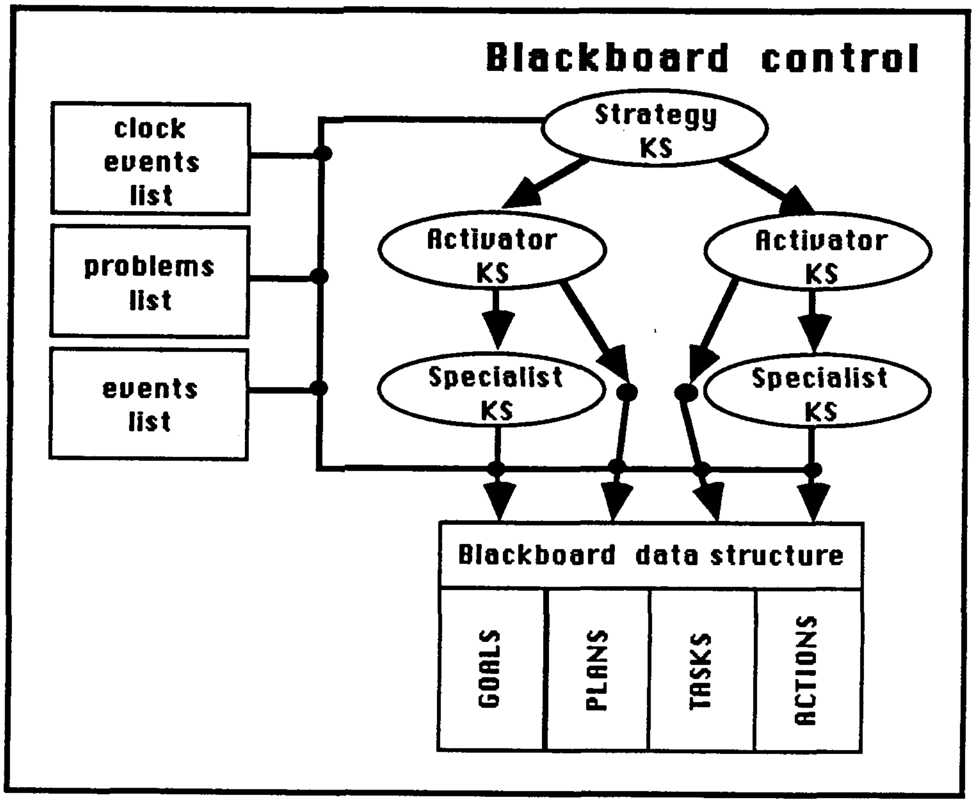
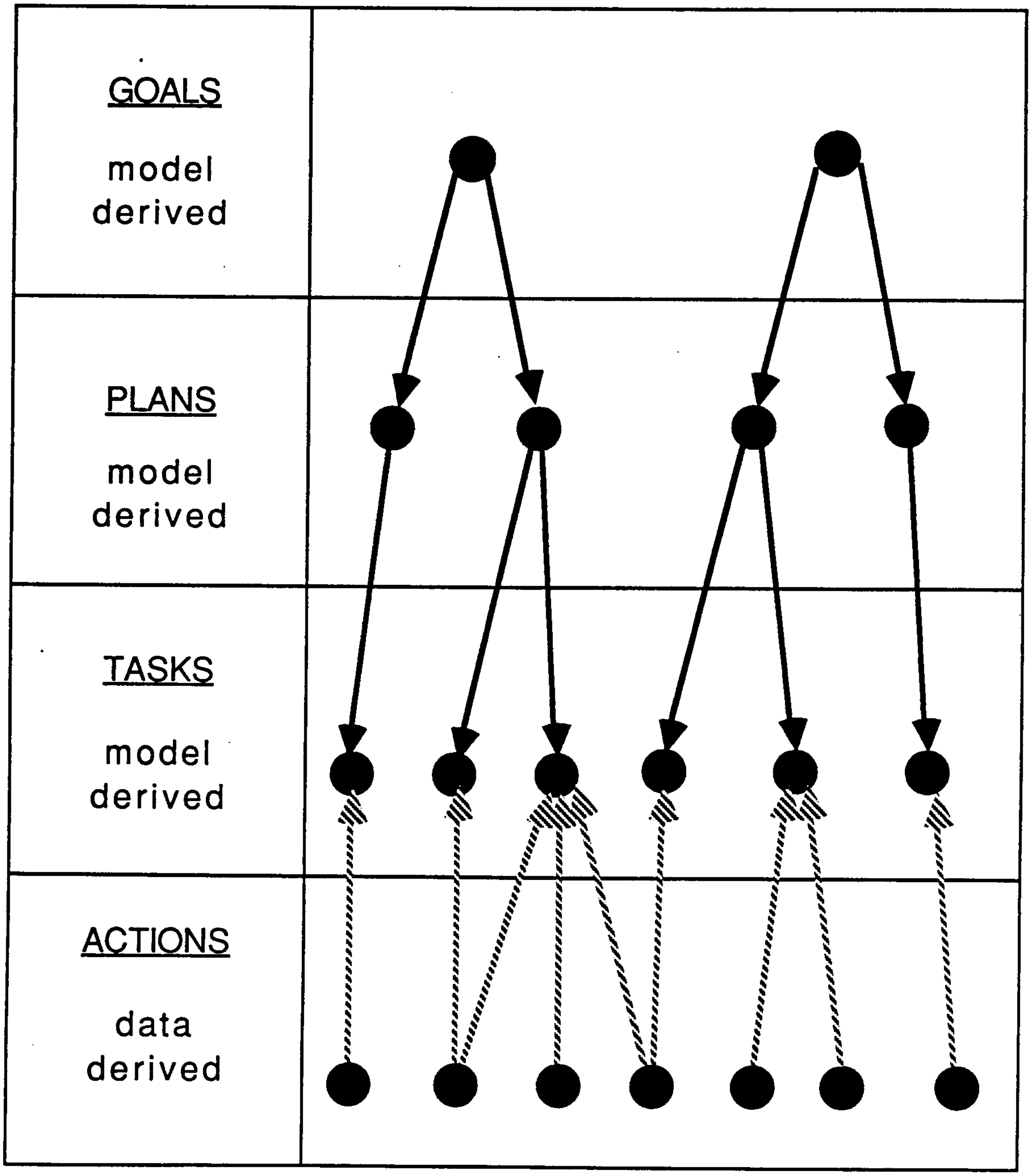


Figure 2. ACTIN's Architecture

Blackboard Model of Interactions

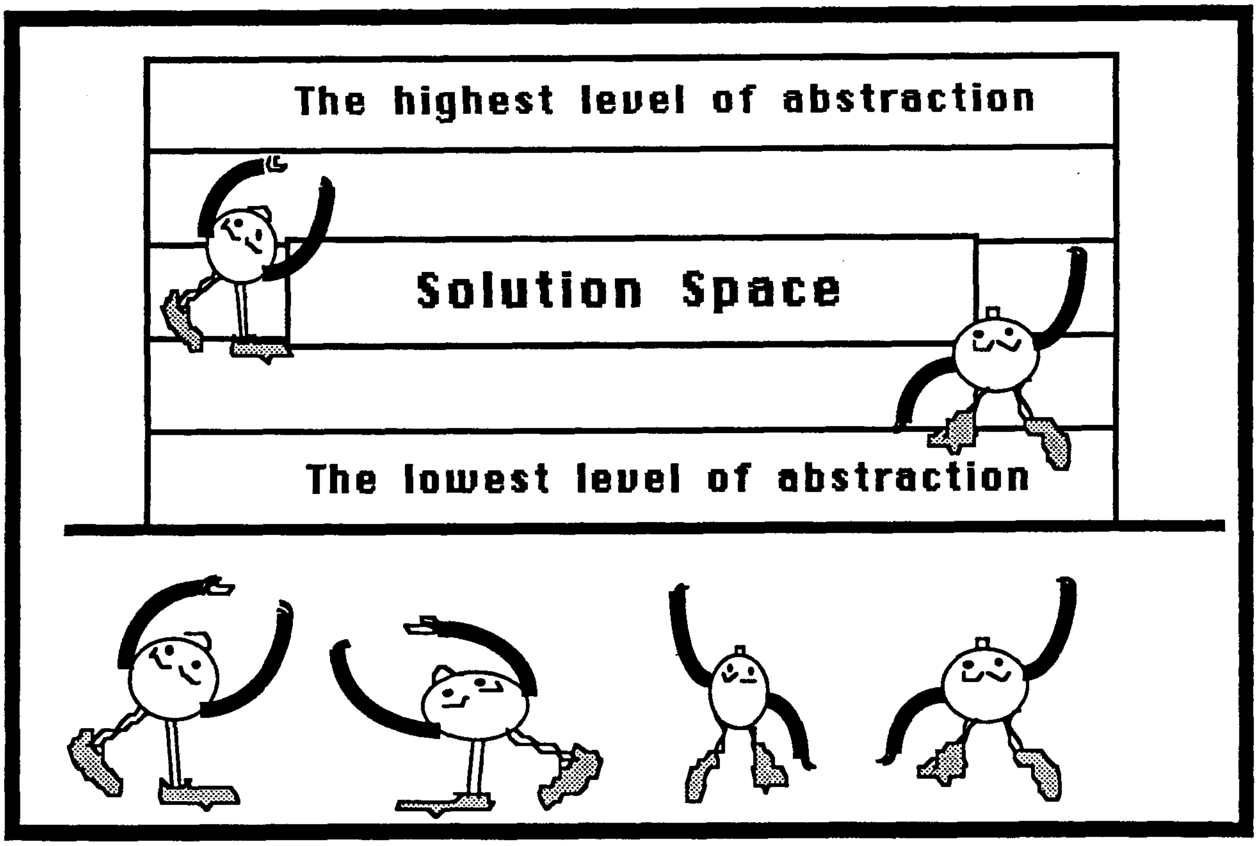


COMPONENTS OF THE BLACKBOARD MODEL:

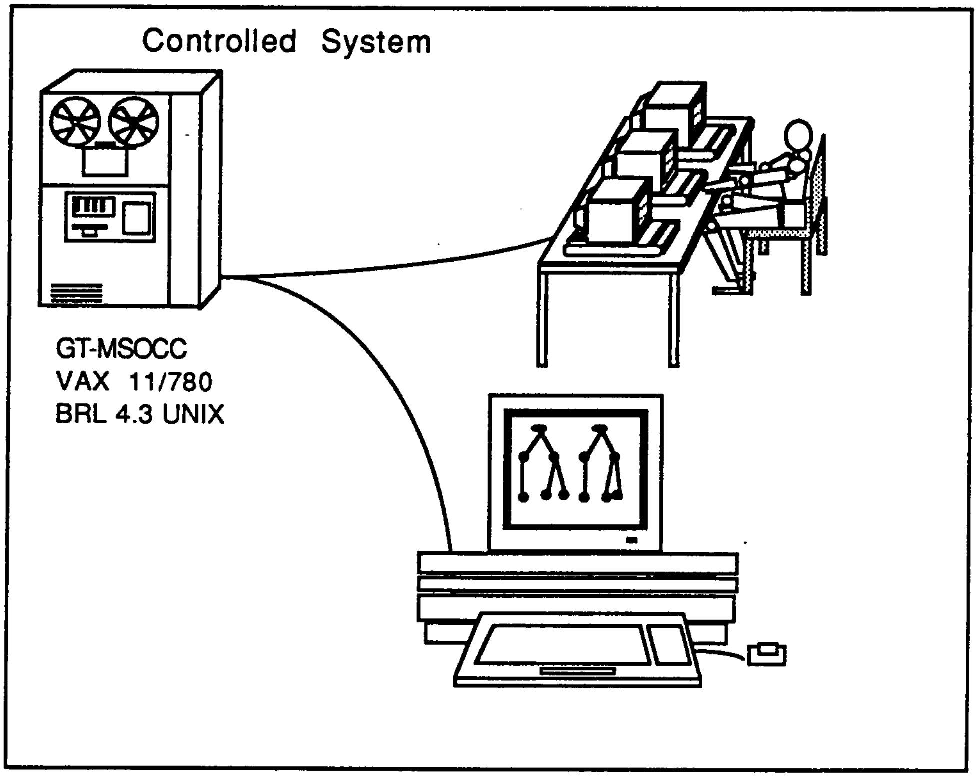
Blackboard data structure

Contains the set of partial and complete solutions known as the solution space.

Divided into levels of information where each level represents a distinct level of abstraction in the solution space.



Blackboard data structure


Controlled System
GT-MSOCC
VAX 11/780
BRL 4.3 UNIX

Experimental Validation

of OFMspert's Intent Inferencing

- Compare a domain expert's interpretations of operator actions to OFMspert's interpretation of those actions.

- Compare verbal protocols of subjects verbalizing their intentions for each action to OFMspert's interpretations of those actions.

Table 3. Experiment 1: Average Percentage of Equivalent Interpretations between ACTIN and a Human Domain Expert. Ordered by rank.

Configure	100%
Endpoint telemetry page requests	100
Deconfigure	97.1
Telemetry page requests	96.3
Answer	91.4
Reconfigure	91.2
Interior telemetry page requests	87.1
Replace	75.3
Mission schedule page requests	66.7
MSOCC schedule page requests	50.3
Equipment schedule page requests	21.8
Events page request	17.7
Pending page request	16.7

Table 5. Experiment 2: Average Percentage of actions matched by OFMspert

Configure	100%
Deconfigure	100
Answer	96.2
Replace	94.8
Equipment schedule page requests	90.3
Mission schedule page requests	85.7
Interior telemetry page requests	84.3
Endpoint telemetry page requests	76.5
MSOCC schedule page requests	75.5
Telemetry page requests	70.2
Reconfigure	60.8
Events page request	53.9
Pending page request	33.3

Table 2. Experiment 1: Proportions of Equivalent Interpretations between ACTIN and a Human Domain Expert

	Subject 1	2	3	4	5	6	7	8	9	10
Telem	7/7 *	32/36 *	11/12 *	22/23 *	40/41 *	11/11 *	39/40 *	18/19 *	28/29 *	29/29 *
Endpoint Telem	2/2	25/25 *	2/2	13/13 *	27/27 *	4/4	24/24 *	8/8 *	1/1	19/19 *
Interior Telem	14/14 *	29/36 *	27/30 *	20/23 *	17/24	20/22 *	25/29 *	18/18 *	42/50 *	22/27 *
MSOCC Sched	14/24	7/11	14/26	10/33	2/27 #	7/26 #	18/37	10/11 *	13/19	6/11
Equip Sched	5/12	1/12 #	10/21	7/87 #	0/13 #	0/7 #	3/11	24/48	10/39 #	2/21 #
Mission Sched	--	--	--	--	--	--	--	--	6/9	--
Events	1/23 #	1/11 #	2/16 #	0/6 #	2/7	3/21 #	5/10	1/17 #	5/26 #	3/9
Pending	--	--	--	0/1	--	0/1	0/1	1/2	1/3	--
Deconfig	12/12 *	16/16 *	11/11 *	15/15 *	12/17	12/12 *	19/19 *	12/12 *	16/16 *	16/16 *
Reconfig	4/4	6/7	1/1	2/3	6/6 *	6/6 *	8/8 *	2/2	3/5	5/5 *
Config	3/3	4/4	3/3	4/4	3/3	--	4/4	3/3	5/5 *	4/4
Replace	12/12 *	13/18	12/17	14/17 *	14/21	12/18	14/22	12/14 *	11/15	15/21
Answer	5/5 *	3/7	5/6	8/8 *	7/8 *	9/9 *	8/8 *	8/8 *	9/9 *	9/9 *

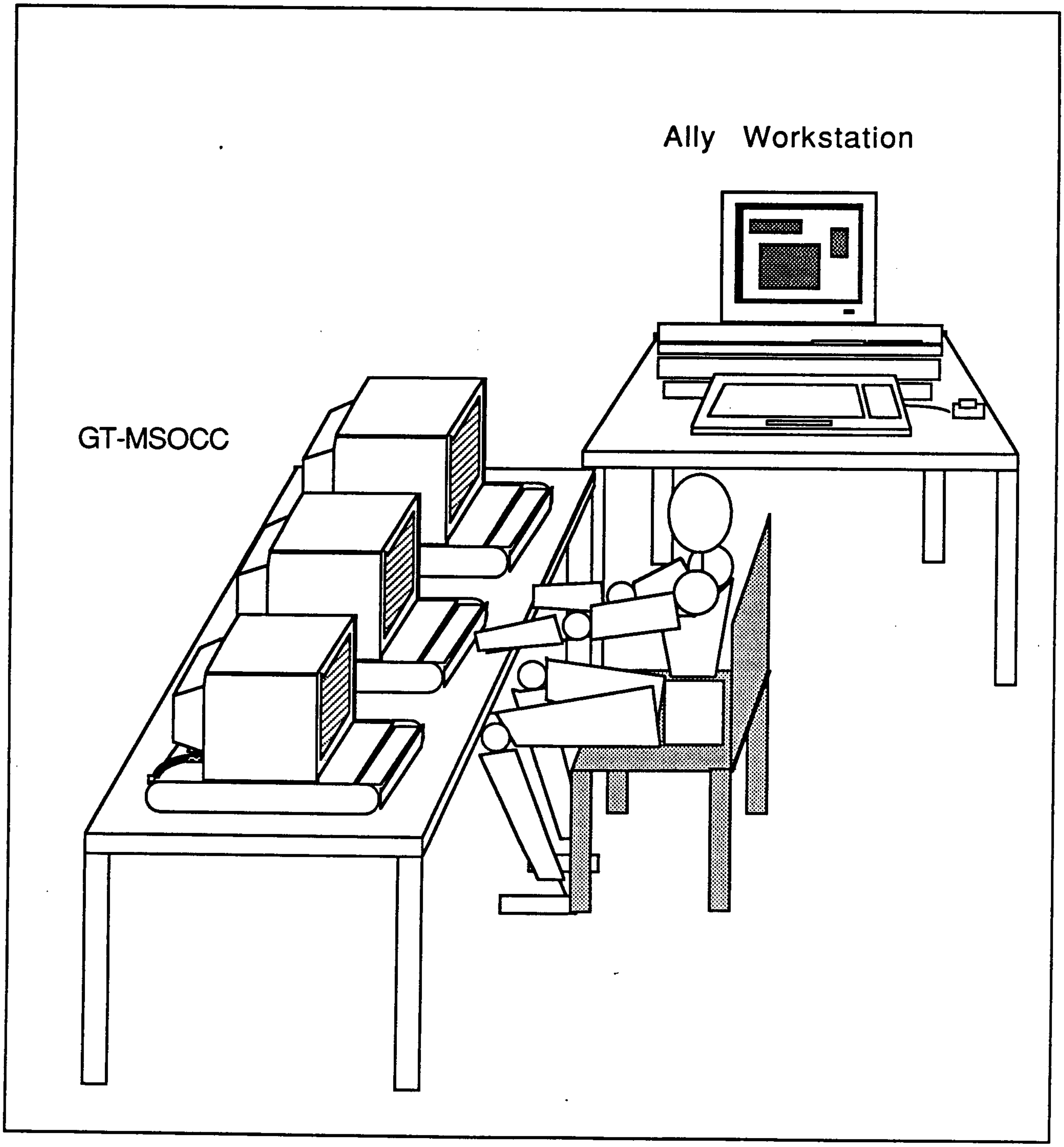
* Significantly good match
\# Significantly poor match

Table 4. Experiment 2: Proportions of Equivalent Interpretations between ACTIN and Verbal Reports

	Subject 21	Subject 22
Telem	30/42 *	40/58 *
Endpoint Telem	33/39 *	26/38 *
Interior Telem	15/19 *	26/29 *
MSOCC Sched	36/45 *	22/31 *
Equip Sched	4/4	25/31 *
Mission Sched	8/8 *	5/7
Events	11/18	7/15
Pending	0/3	4/6
Deconfig	31/31 *	30/30 *
Reconfig	7/15	6/8
Config	5/5 *	3/3
Replace	23/23 *	26/29 *
Answer	12/12 *	12/13 *

* Significantly good match (B> b(0.025,n,0.5)

\# Significantly poor match (B< n-b(0.025,n,0.5)


Ally Workstation
GT-MSOCC

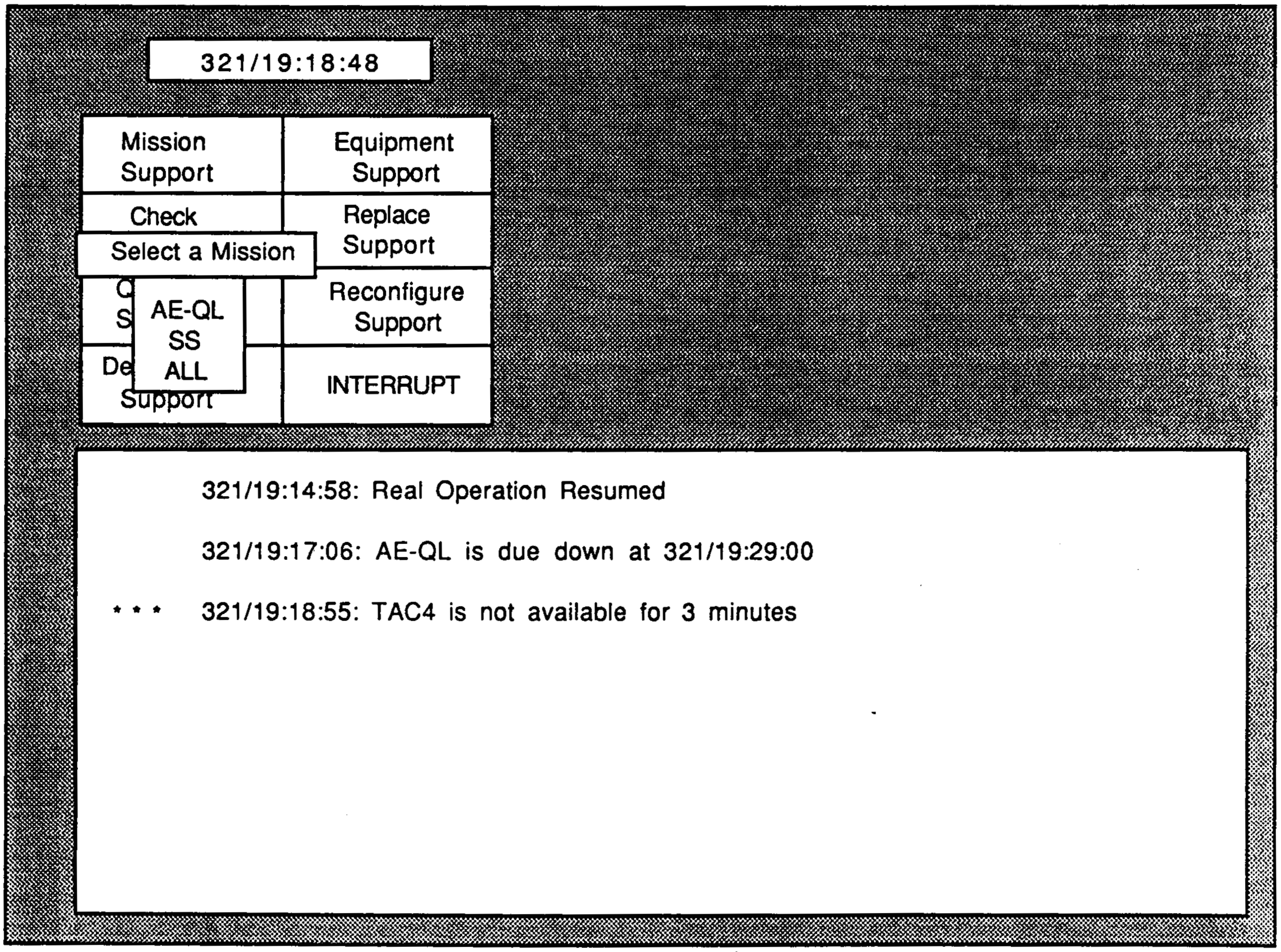
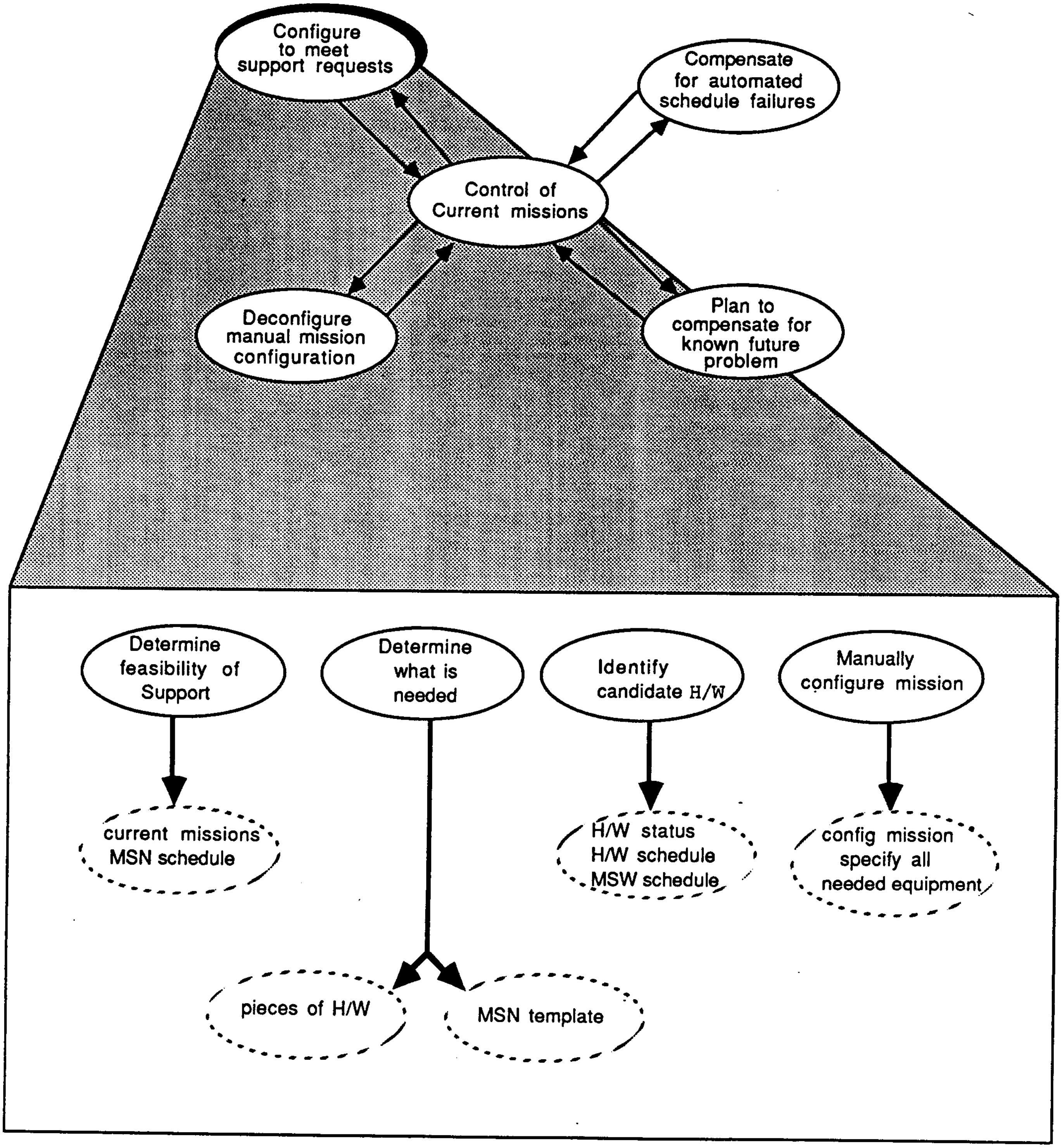
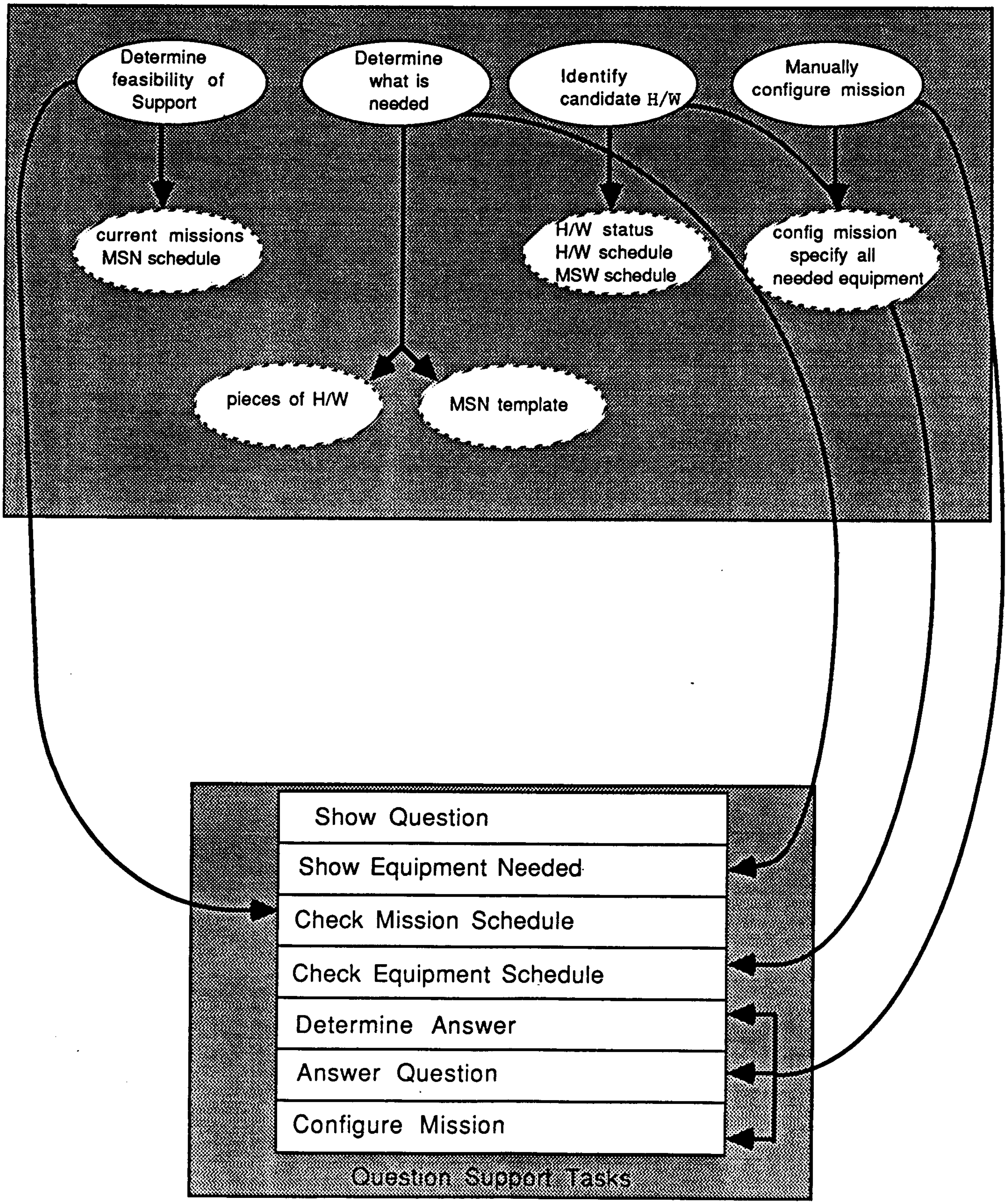


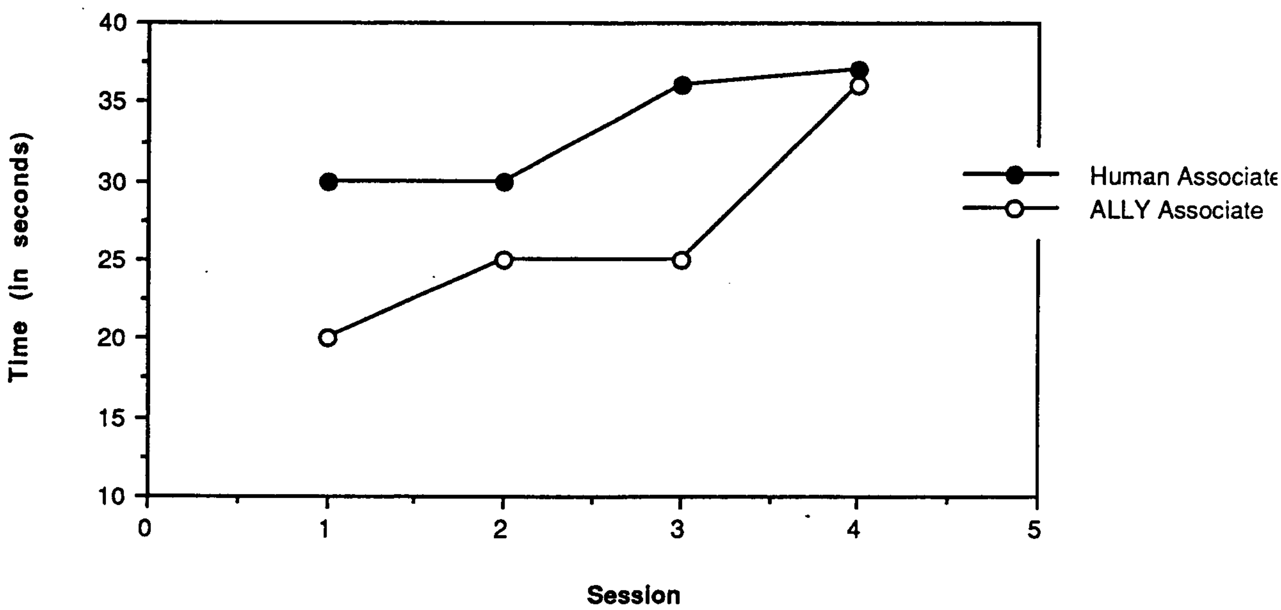
Figure 10. Example of Ally's User Interface



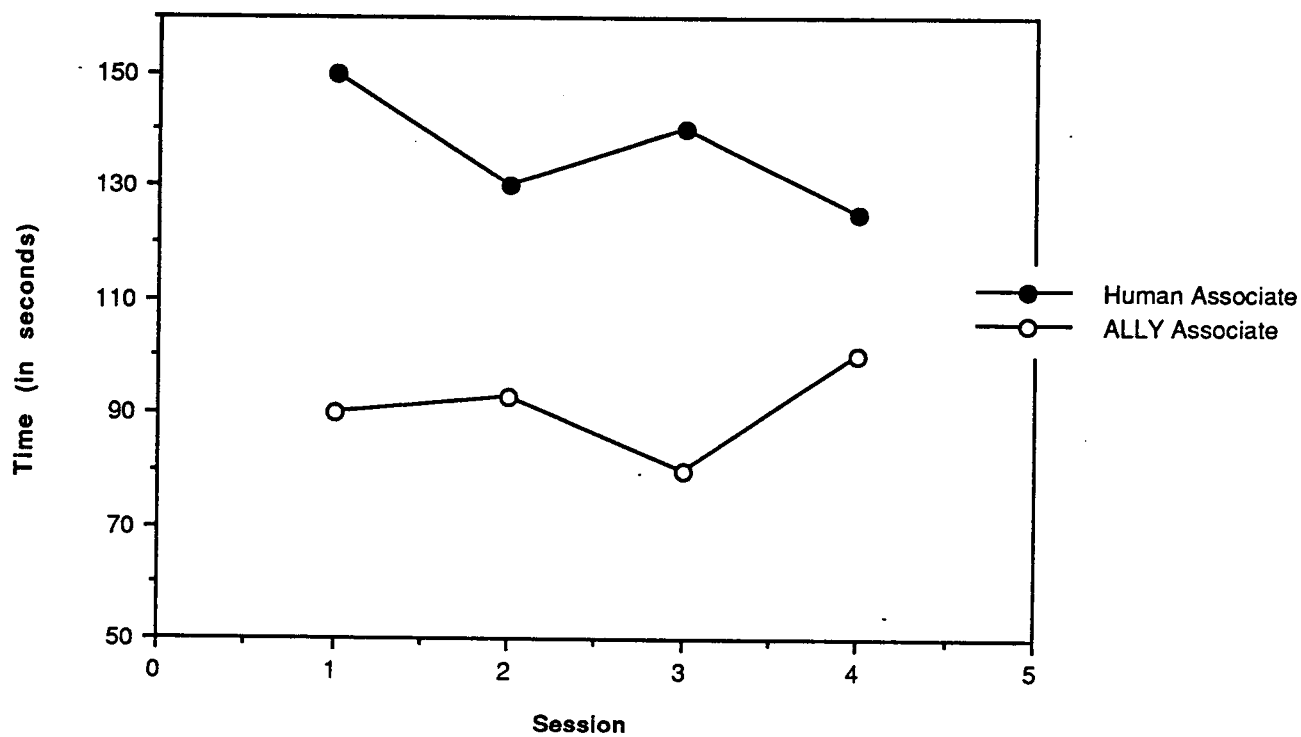
Configure to meet support requests
Compensate for automated schedule failures
Control of Current missions
Deconfigure manual mission configuration
Plan to compensate for known future problem
Determine feasibility of Support
Determine what is needed
Identify candidate H/W
Manually configure mission
current missions MSN schedule
H/W status H/W schedule MSW schedule
config mission specify all needed equipment
pieces of H/W
MSN template


Determine feasibility of Support
Determine what is needed
Identify candidate H/W
Manually configure mission
current missions MSN schedule
H/W status H/W schedule MSW schedule
config mission specify all needed equipment
pieces of H/W
MSN template
Show Question
Show Equipment Needed
Check Mission Schedule
Check Equipment Schedule
Determine Answer
Answer Question
Configure Mission
Question Support Tasks

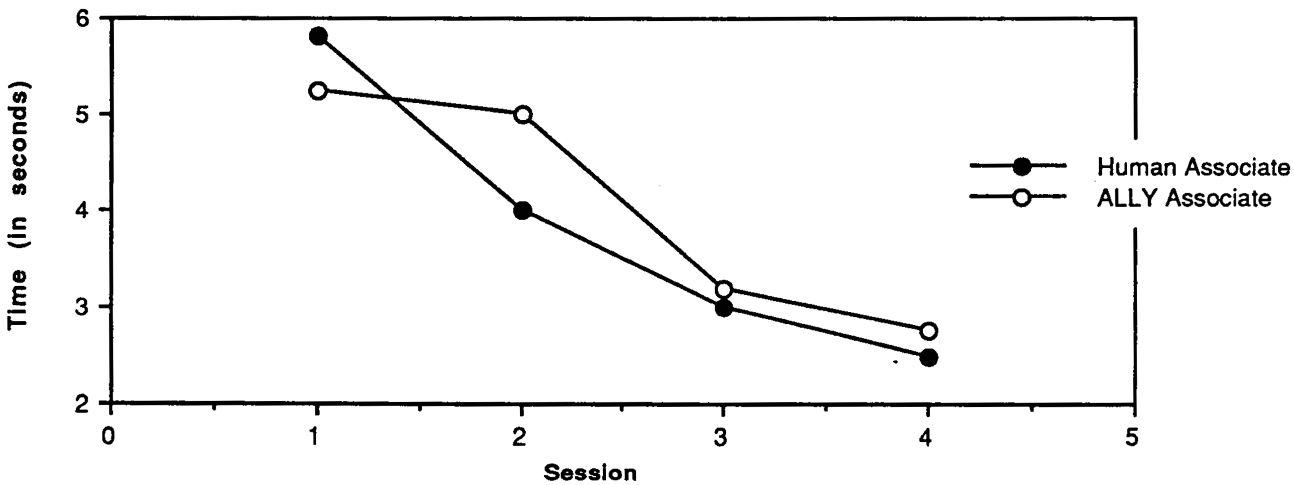
Mean Time to Compensate for Hardware Failure by Session



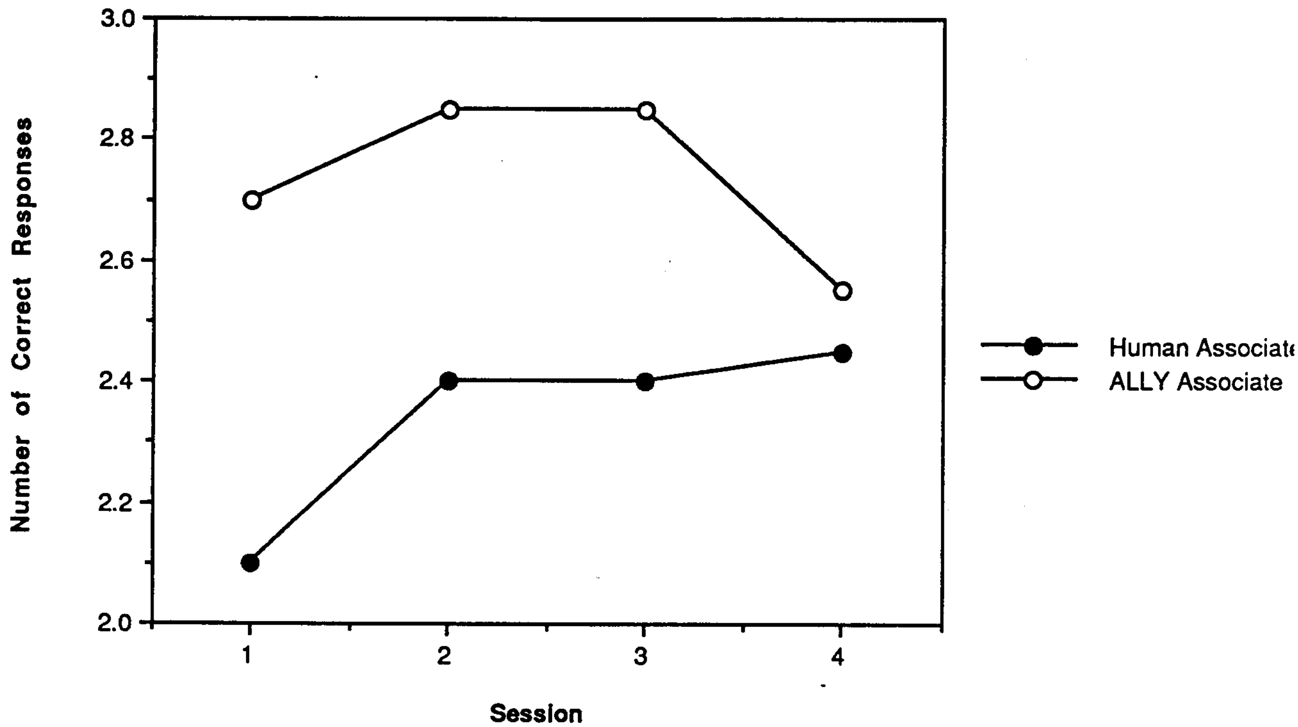
Mean Time to Compensate for Software Type 1 Failures by Session



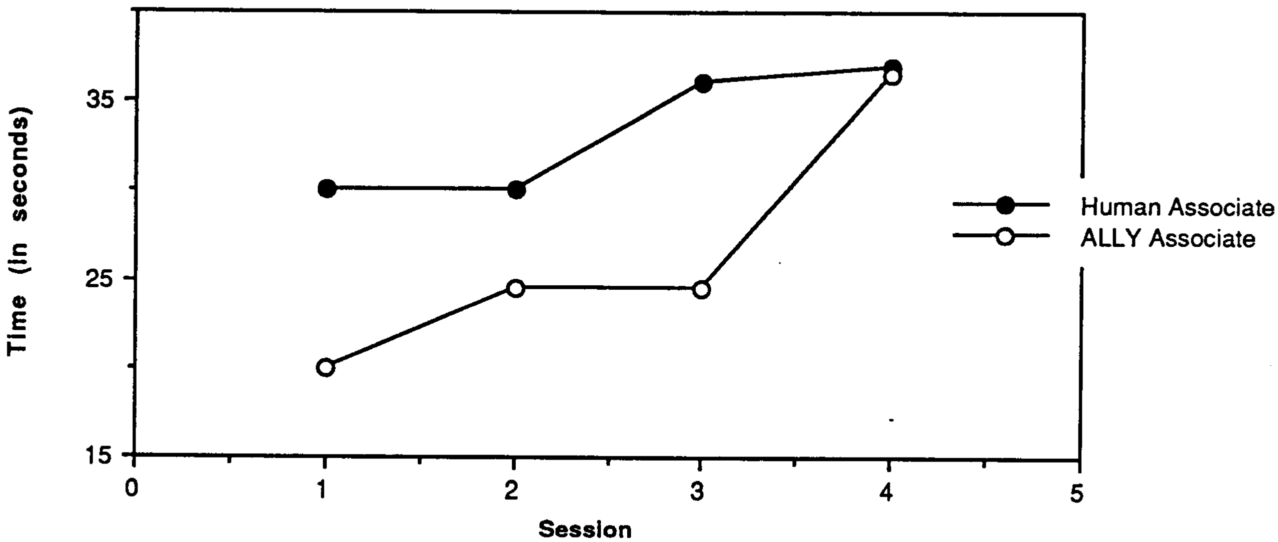
Mean Time to Compensate for Schedule Conflicts by Session



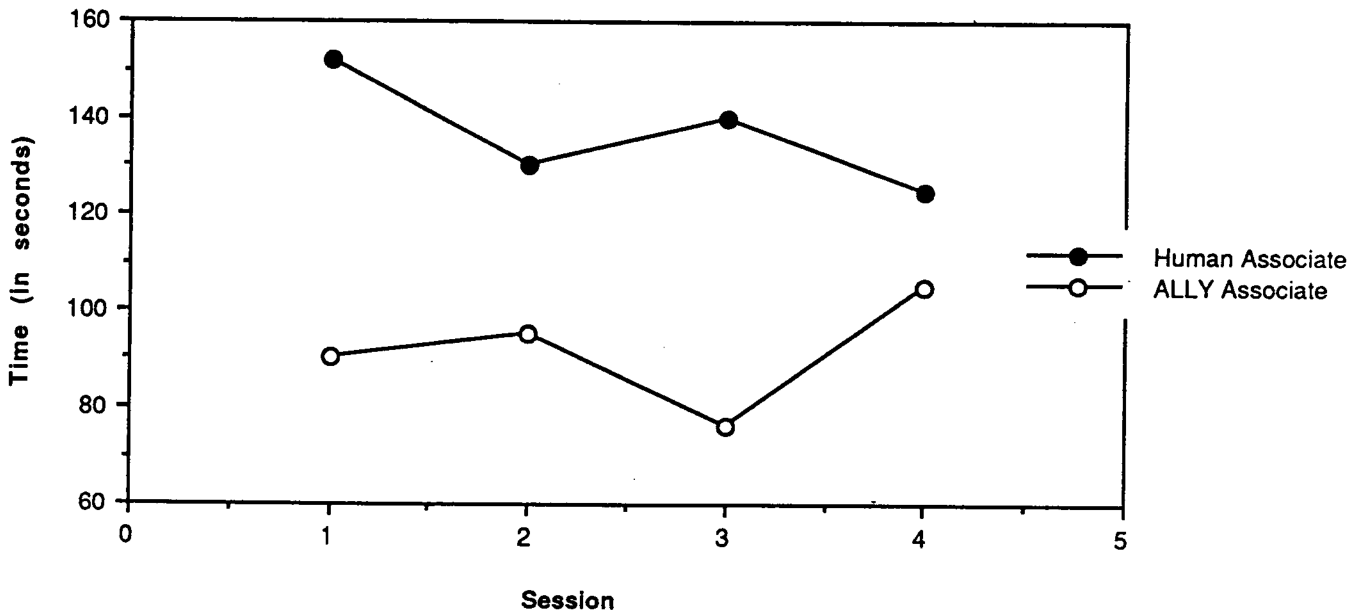
Mean Number of Correct Responses to Support Requests by Session



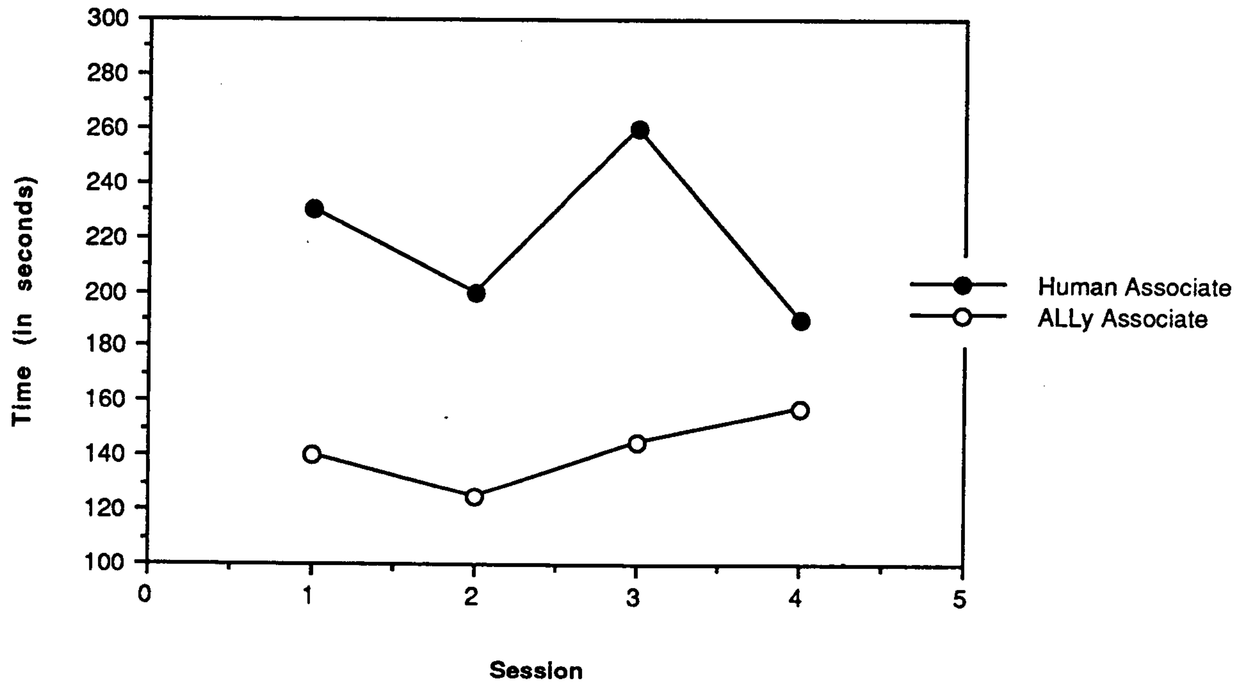
Mean Time to Compensate for Hardware Failure by Session



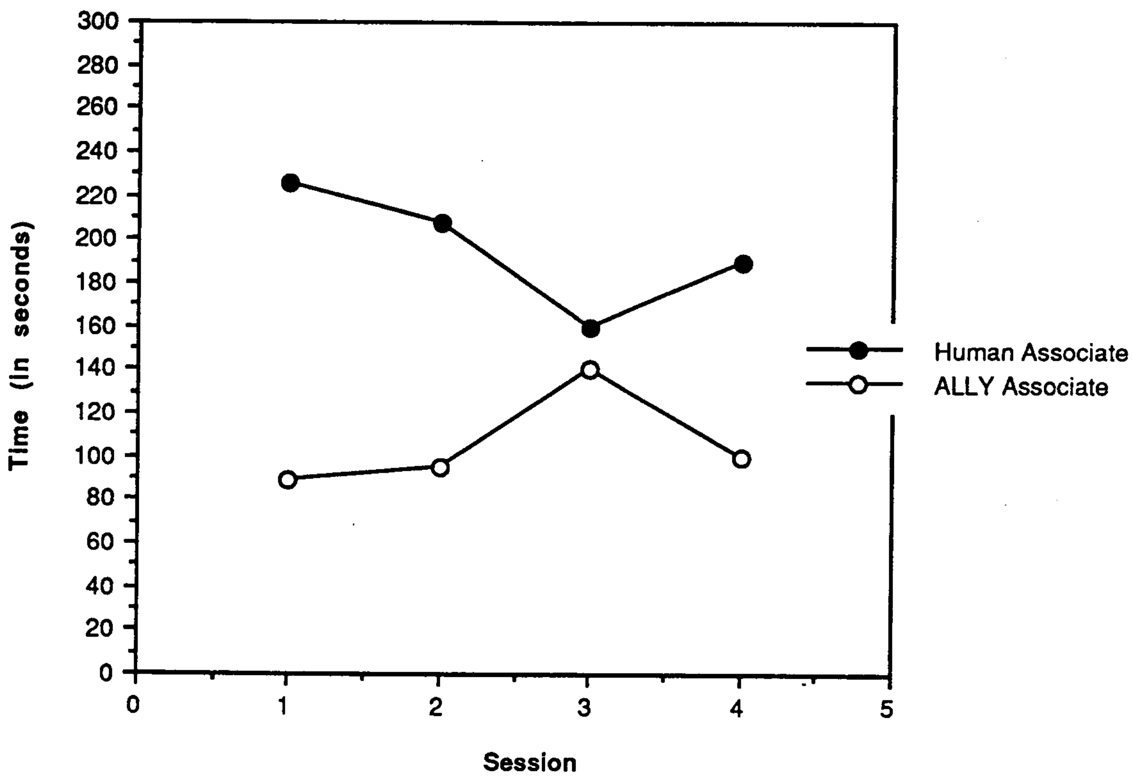
Mean Time to Compensate for Software Type 1 Failures by Session



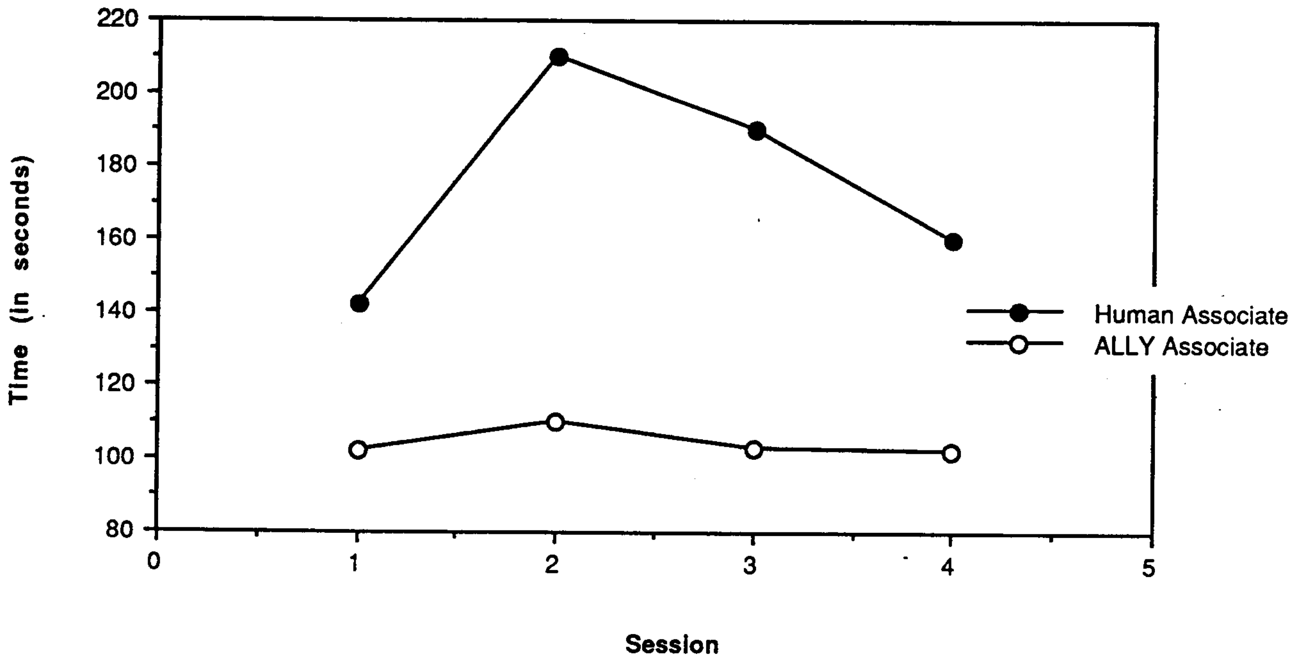
Mean Time to Compensate for Software Type 2 Failures by Session



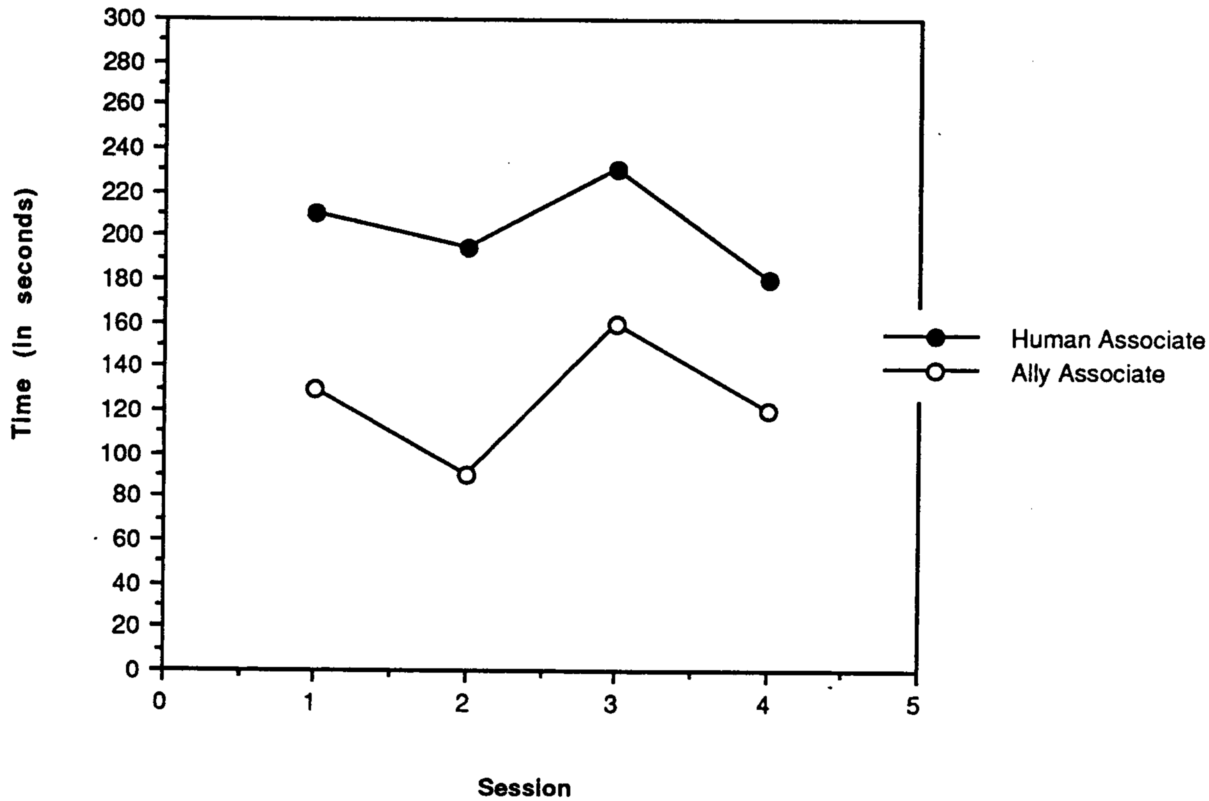
Mean Time to Compensate for Software Type 3 Failures by Session



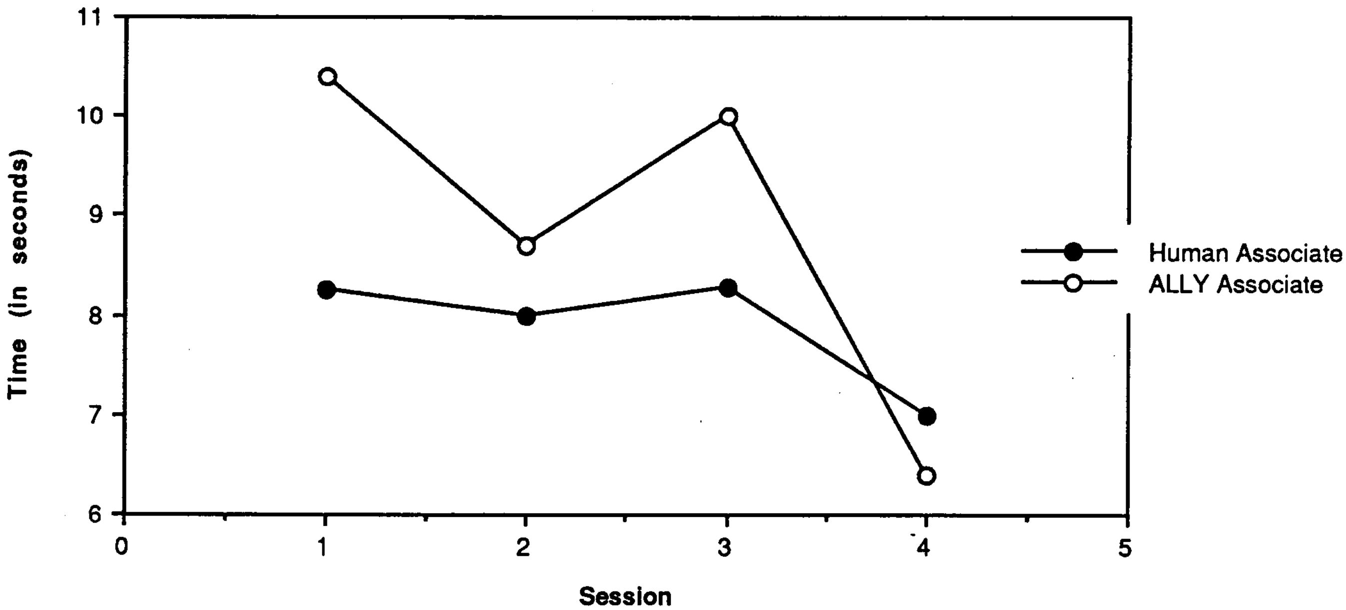
Mean Time to Respond to Support Requests by Session

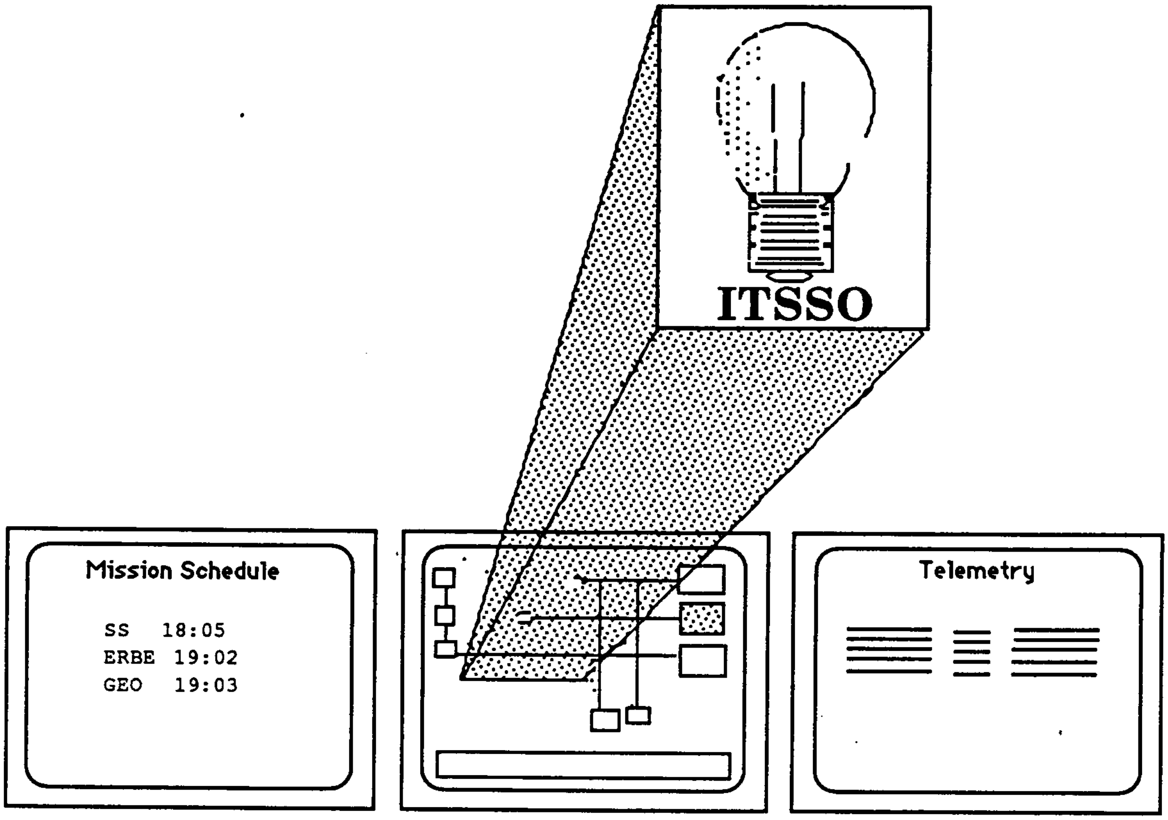


Mean Time to Configure Unscheduled Support Contacts by Session

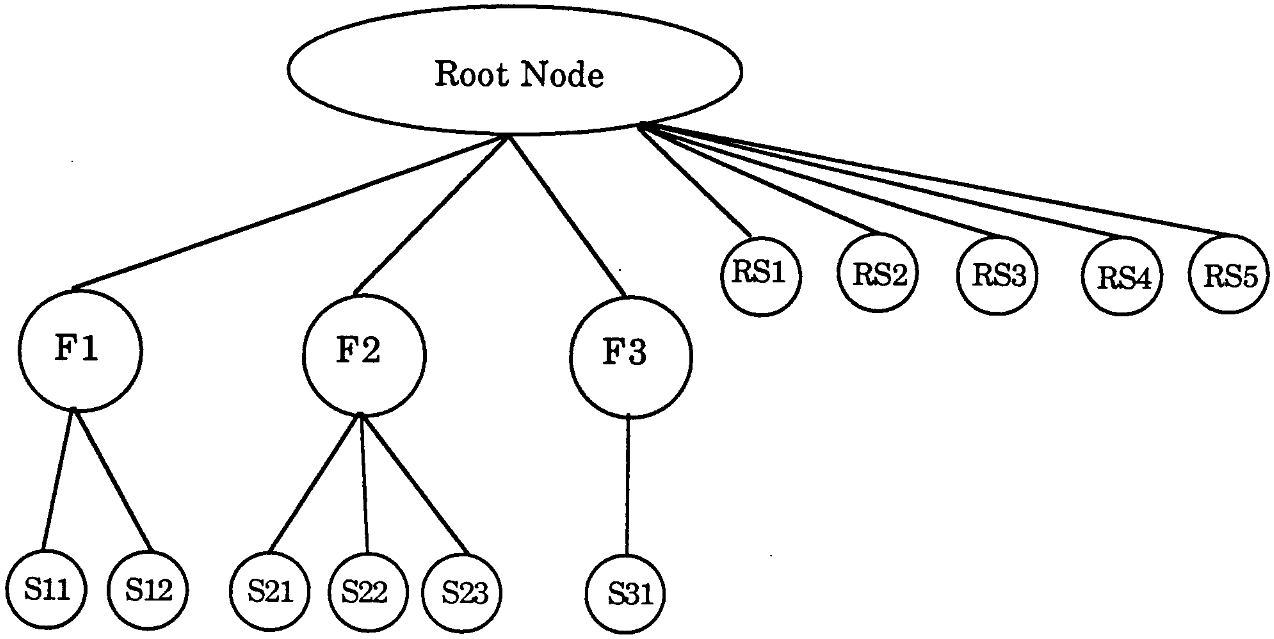


Mean Time to Respond to Deconfigure Requests by Session

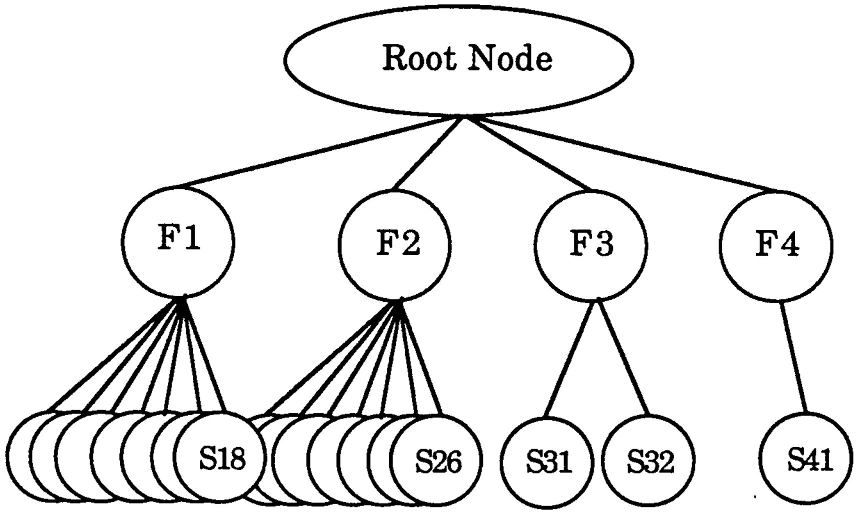




Intelligent Tutoring System for Satellite Operators



A Task Model



Root Node: Supervisory Control of GT-MSOCC

F1: Control of Current Missions

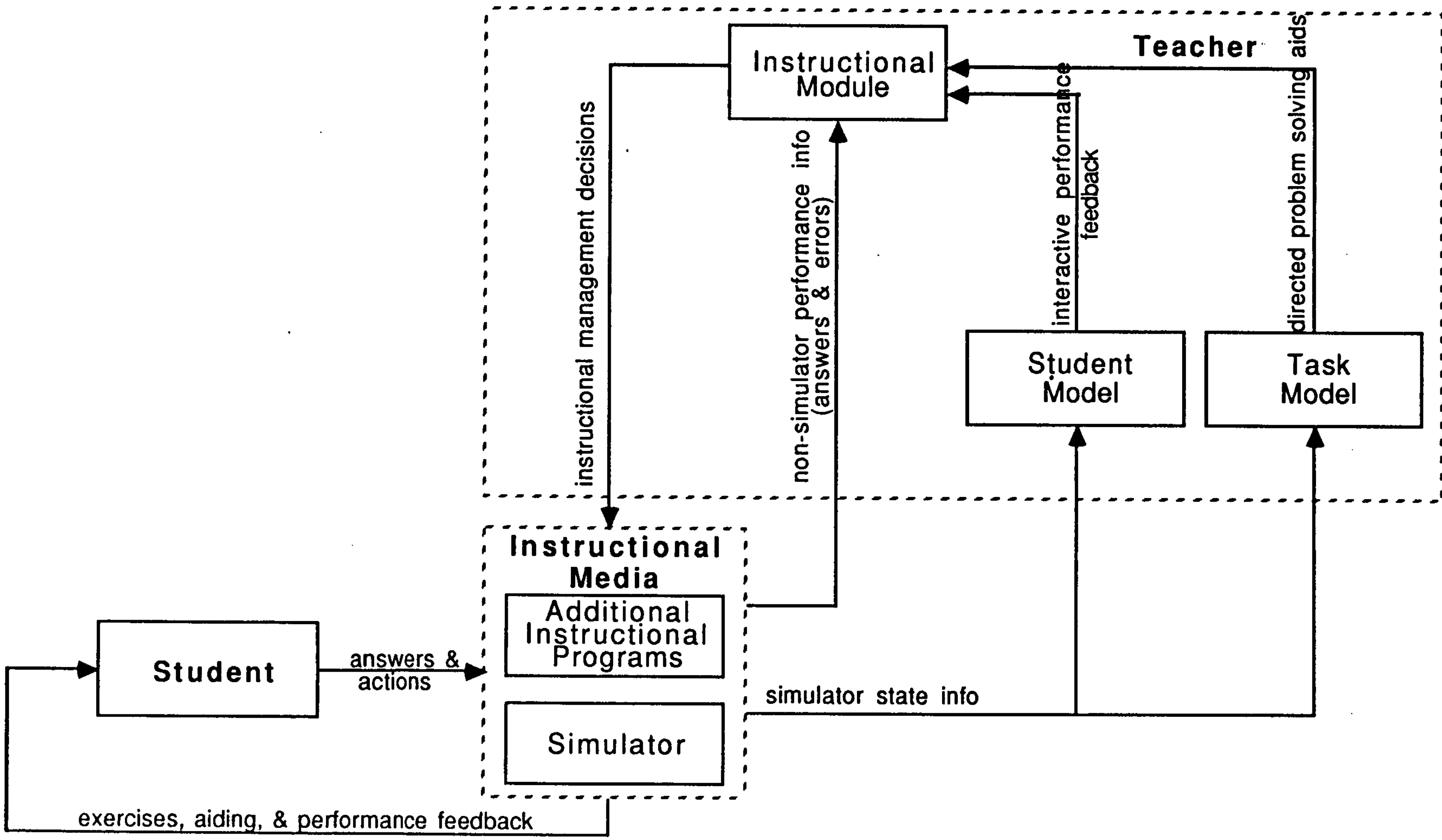
Type of Failure	Replaceable	NonReplaceable
Hardware Failure	S11	S15
No Data Relayed	S12	S16
Half Normal Data	S13	S17
Triple Normal Errors	S14	S18

F2: Configure to Meet Support Request

F3: Compensate for Automated Schedule Failures

F4: Manually Deconfigure a Mission

Figure 3 A Task Model


Teacher
Instructional Module
instructional management decisions
non-simulator performance info (answers & errors)
interactive performance feedback
directed problem solving aids
Student Model
Task Model
Instructional Media
Additional Instructional Programs
Simulator
simulator state info
Student
answers & actions
exercises, aiding, & performance feedback

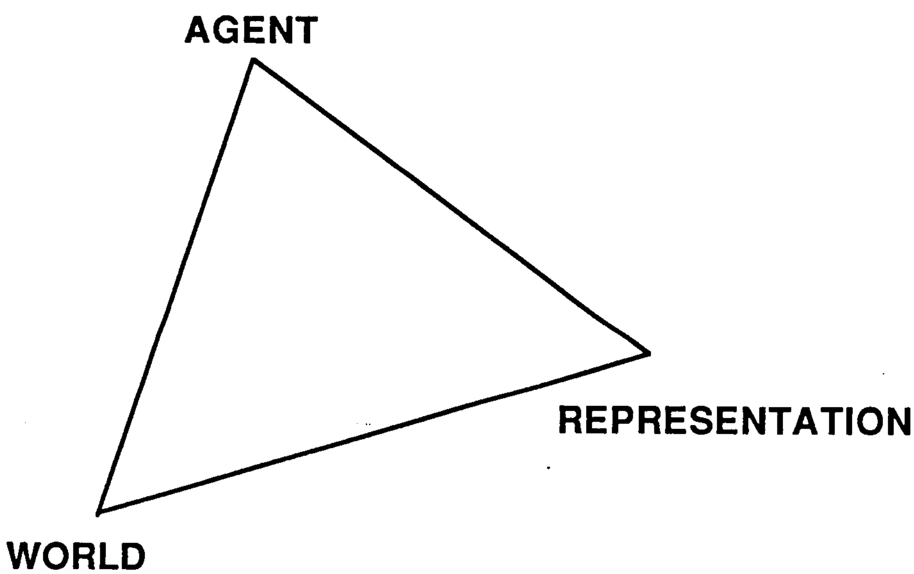
COMPLEXITY AND PROBLEM SOLVING

Three basic elements in problem solving situations:

The **World** to be acted on,

The **Agent** who acts on the world,

The external **representation** of the world utilized by the problem solving agent.



WHAT MAKES PROBLEM SOLVING COMPLEX:

DIMENSIONS OF COMPLEXITY

- Dynamism
- Number of parts and extensiveness of interconnections between parts
- Uncertainty
- Risk

DOMAIN OF INTEREST: COMPLEX DYNAMIC SYSTEMS

- Human operator as a supervisory controller
  - monitoring task
  - troubleshooting task

QUESTIONS TO BE ADDRESSED

What are the specific skills with respect to the four dimensions of complexity that are necessary to carry out the tasks involved in a CDS?

What are the goals of an ITS designed for a CDS? What do we want the operator to learn? Are the goals attainable?

What approaches in each module of an ITS seem appropriate to a CDS and why? How do we translate an approach in the context of a CDS?

What about implementation issues and "do-ability"? How much of the CDS world should be represented in the ITS?

How do we evaluate the ITS (if implemented) to test if the goals are attained?

COMPONENTS OF AN INTELLIGENT TUTORIAL SYSTEM

- Domain Expertise
- Student Model
- Pedagogical Expertise
- Interface

Figure 3. ACTIN's Intent Inferencing Structure

REVIEW OF APPROACHES

Domain Expertise:

- Information-structure-oriented paradigm (SCHOLAR, 1970)
- Hierarchical scripts (WHY, 1977)
- Finite state automata (METEOROLOGY, 1973)
- Multiple representations of procedural and declarative knowledge (SOPHIE I, 1975; RBT, 1986)
- Qualitative modelling (STEAMER, 1984)
- Probabilistic model (INTEGRATION, 1973)
- D-rules (MYCIN/GUIDON, 1979)
- Procedural networks (BUGGY, 1975)
- Generalized AND/OR graph (REPAIR theory, 1980)
- Problem-solving models:
    - Active structural networks (FLOW, 1974)
    - Linguistics theory (SPADE, 1976)
    - Dependency graphs (MACSYMA ADVISOR 1977)
    - Intention-based knowledge structure (PROUST, 1984)
    - Operator function model (AHAB, 1987)

STUDENT MODEL

- Differential model (WEST, 1976)
- Overlay model (WUSOR-II, 1977; GUIDON, 1979)
- Buggy model (BUGGY, 1978; MENO-II; PROUST, 1984)
- Limited bug model (AHAB, 1987)

INTERFACE

- Textual (SCHOLAR, 1970; SOPHIE, 1975; WEST, 1976; GUIDON, 1979; PROUST, 1984; etc)
- Graphical (ALGEBRALAND, 1983; STEAMER, 1984; IMTS, 1986; RBT, 1986; AHAB, 1987)

REVIEW OF APPROACHES (cont'd)

Pedagogical Expertise:

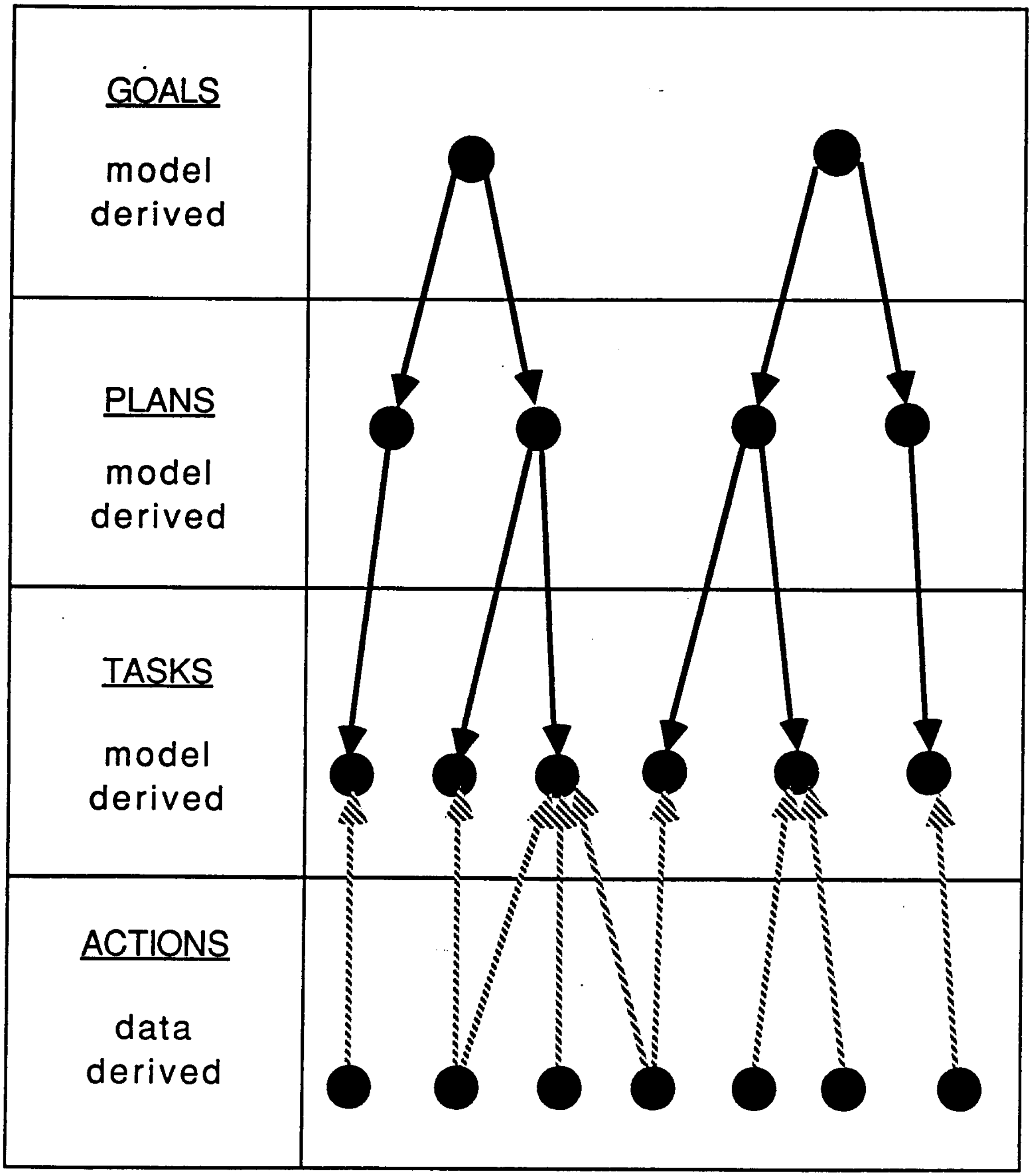
- Socratic method (WHY, 1977)
- Reactive learning environment (SOPHIE I, 1975; MACSYMA ADVISOR, 1977)
- Conceptual fidelity (STEAMER, 1984; AHAB, 1987)
- Progression of O-order qualitative models (QUEST, 1986)
- Curriculum Information Network (BIP, 1976)
- Exploratory learning (LOGO, 1980)
- Issues and examples paradigm (WEST, 1976)
- Increasingly complex microworlds paradigm (Fischer, et al., 1978)
- Expert-based coaching (WUSOR-I, 1976)
- Bite-sized architecture (SMITHTOWN, 1986)
- Layered curriculum and steering test concept (MHO, 1987)
- Discourse management networks (MENO-TUTOR, 1984)
- T-rules (GUIDON, 1979)
- ACT theory (GEOMETRY and LISP tutors, 1984)

OFMTutor

- Intelligent tutoring system for operators of complex dynamic systems

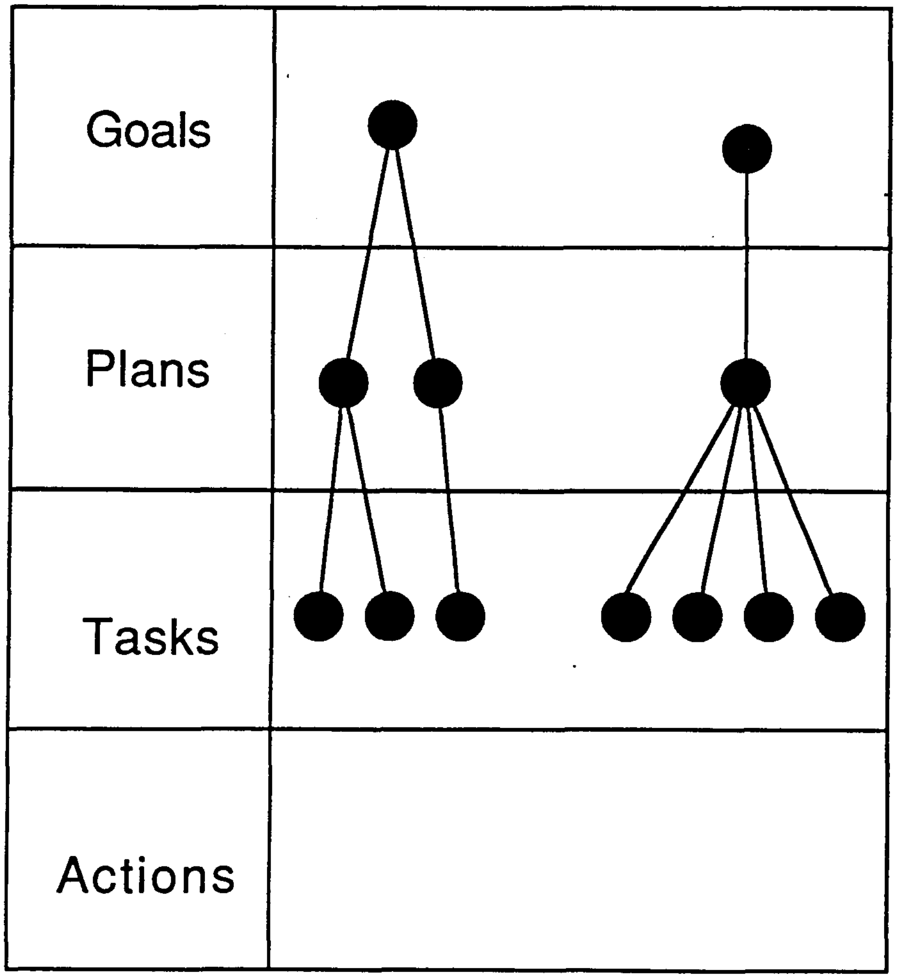
- Based on the Operator Function Model (OFM)

Blackboard model of Interactions



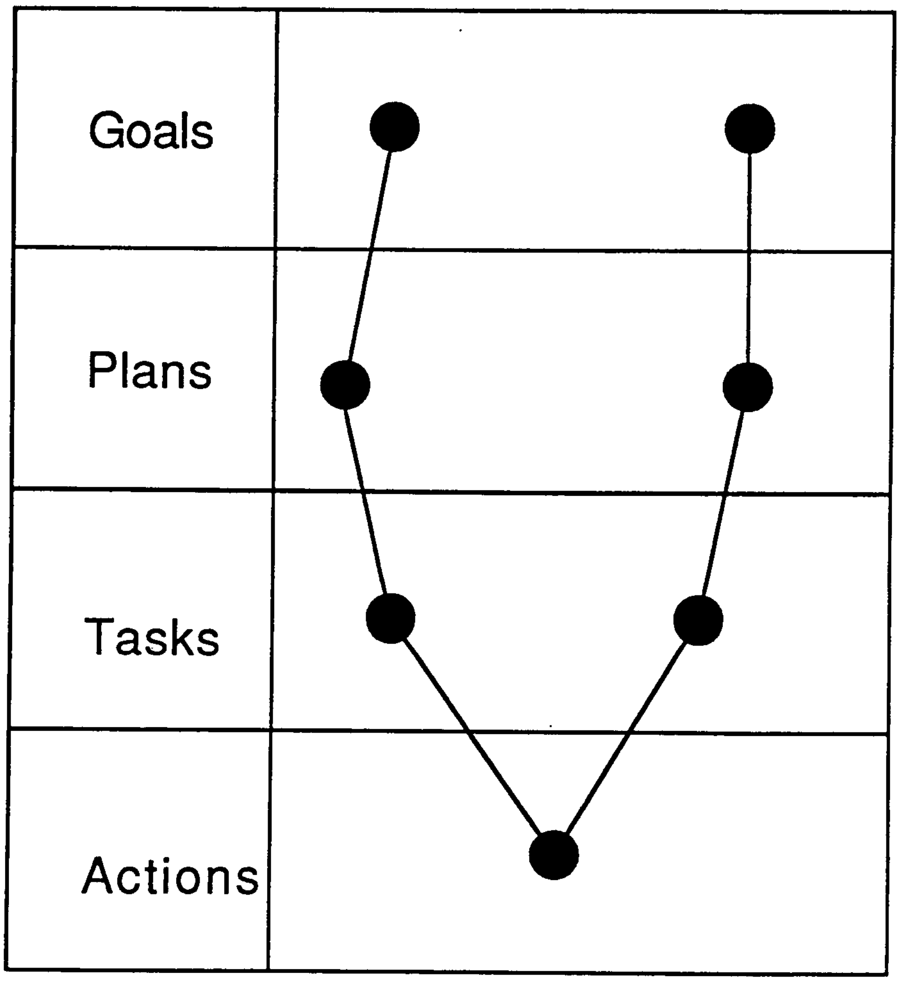
OFMTutor's Model of Expertise

Model derived representative of Goals, Plans, and Tasks



OFMTutor's Student Model

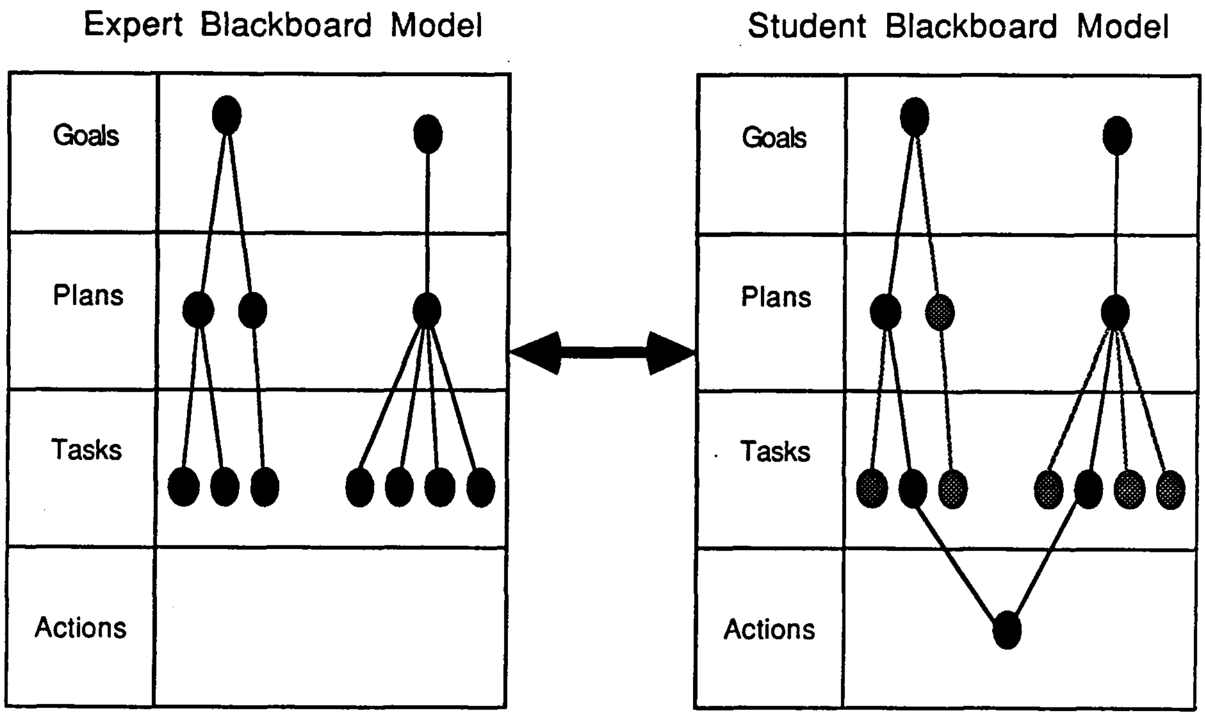
Data derived representation of goals, plans, and tasks based on student's actions



OFMTutor's Pedagogical Strategy and Diagnosis

Guided discovery/coaching in context of system operation

Differential modeling techniques that compare expert and student blackboard models



Work in Process

* Design of a computer-based operator associate that evolves from tutor to assistant as the skills of the human operator change from novice to expert.

* The refinement of the Ally interaction to allow cooperative problem solving and repair of hypothesis formation.

* Evolution of a broader theory of 'good' architectures utilizing human and computer decision makers in interactive control.

OFMTutor's Interface

Supports graphical, inspectable representation of joint hypotheses (expert and student)

Model of discourse enables conversational capabilities and supports repair