

**DRAFT**

MAINTENANCE SKILLS TUTOR (MST)  
SOFTWARE ENGINEERING REFERENCE GUIDE (SERG)

PREPARED BY:  
THE MITRE CORPORATION

PREPARED FOR:  
AIR FORCE SYSTEMS COMMAND HUMAN SYSTEMS DIVISION

**DRAFT**

## TABLE OF CONTENTS

SECTION	PAGE
1. INTRODUCTION	1
1.1 The Maintenance Skills Tutor Program	1
1.2 The Basic Job Skills Program	1
1.3 Contents	2
2. FUNCTIONAL OVERVIEW OF INTELLIGENT TUTORING SYSTEMS	3
2.1 Evolution of Computer Assisted Instruction	3
2.2 Software Components of ITSs	3
3. OVERVIEW OF THE MAINTENANCE SKILLS TUTOR	7
3.1 Definition of System Architecture	7
3.2 Goals of the MST Architecture	7
3.3 Means of Achieving the Goals	8
3.4 Architectural Overview	8
4. FUNCTIONAL DESCRIPTION OF THE MAINTENANCE SKILLS TUTOR	13
4.1 Executive	14
4.1.1 Inputs	15
4.1.2 Processing	15
4.1.2.1 Message Handler	16
4.1.2.2 MST Controller	19
4.1.2.3 Checkpoint / Bookmark	21
4.1.2.4 Executive Services Function	21
4.1.3 Outputs	22
4.2 Instructional Environment	22
4.2.1 Inputs	24
4.2.2 Processing	26
4.2.2.1 System Control Function	26
4.2.2.2 Checkpoint / Bookmark Function	26
4.2.2.3 Student Action Handling Function	27
4.2.2.4 Unsolicited Advice Determination Function	28
4.2.2.5 Coaching Coordination Function	29
4.2.2.6 Review Coordination Function	30
4.2.2.7 Reflective Follow-up Coordination Function	30
4.2.3 Outputs	31
4.3 Simulator	33
4.3.1 Inputs	34

4.3.2	Processing	34
4.3.2.1	System Control Function	35
4.3.2.2	Checkpoint / Bookmark Function	35
4.3.2.4	Initialization Function	35
4.3.2.3	State Maintenance Function	36
4.3.2.5	State Reporting Function	37
4.3.2.6	Simulation Control Function	37
4.3.3	Outputs	38
4.4	Expert Model	40
4.4.1	Inputs	40
4.4.2	Processing	41
4.4.2.1	System Control Function	42
4.4.2.2	Checkpoint / Bookmark Function	42
4.4.2.3	Student Action Interpretation Function	42
4.4.2.4	Troubleshooting Procedure and Goal Maintenance Function	43
4.4.2.5	Troubleshooting Action Selection Function	43
4.4.3	Outputs	44
4.5	Student Model	45
4.5.1	Inputs	45
4.5.2	Processing	46
4.5.2.1	System Control Function	46
4.5.2.2	Checkpoint / Bookmark Function	46
4.5.2.3	Problems Attempted Storage / Retrieval Function	46
4.5.2.4	Student Action Storage / Retrieval Function	47
4.5.2.5	Student Action / Performance Variable Mapping Function	47
4.5.3	Outputs	47
4.6	Explanation Generator	48
4.6.1	Inputs	49
4.6.2	Processing	49
4.6.2.1	System Control Function	49
4.6.2.2	Checkpoint / Bookmark Function	49
4.6.2.2	Explanation Generation Script Construction Function	50
4.6.3	Outputs	50
4.7	Presentation Controller	51
4.7.1	Inputs	52
4.7.2	Processing	52
4.7.2.1	System Control Function	52
4.7.2.2	Checkpoint / Bookmark Function	53
4.7.2.3	Explanation Generation Script Translation Function	53
4.7.2.4	Spatio-Temporal Resolution Function	53
4.7.2.5	Display Controller Function	53
4.7.2.6	Option Translation Function	54

4.7.2.7	Option Availability Maintenance Function	54
4.7.3	Outputs	54
4.8	Multimedia Interface	55
4.8.1	Inputs	55
4.8.2	Processing	56
4.8.2.1	Windows Control / Display Function	56
4.8.2.2	Text Display Function	58
4.8.2.3	Graphics Display Function	58
4.8.2.4	Video Display / Control Function	59
4.8.2.5	Audio Display / Control Function	59
4.8.2.6	Student Interaction Processing Function	60
4.8.3	Outputs	60
4.9	Curriculum Manager	61
4.9.1	Inputs	61
4.9.2	Processing	62
4.9.2.1	System Control Function	63
4.9.2.2	Checkpoint / Bookmark Function	63
4.9.2.3	Student Mastery Determination Function	63
4.9.2.4	Problem Selection Mode Determination Function	64
4.9.2.5	Problem Selection Function	64
4.9.2.6	Problem Creation Function	65
4.9.3	Outputs	66
4.10	Training Administrator	67
4.10.1	Inputs	67
4.10.2	Processing	68
4.10.2.1	System Control Function	68
4.10.2.2	Checkpoint / Bookmark Function	68
4.10.2.3	New Training Administrator Registration Function	68
4.10.2.4	Data Recording Function	68
4.10.2.5	Student Data Transfer Function	68
4.10.2.6	Recorded Data Analysis Function	69
4.10.3	Outputs	69
5.	BEHAVIORAL DESCRIPTION OF THE MAINTENANCE SKILLS TUTOR	71
5.1	Data Organization	71
5.2	Behavioral Notation	74
5.3	Executive	75
5.3.1	Inputs	76
5.3.1.1	Messages Received	76
5.3.1.2	Databases Accessed	77
5.3.1.3	Other Input Data Items	77
5.3.2	Algorithms	77

5.3.2.1	Message Handling Function	77
5.3.2.1.1	Software Component Registration	78
5.3.2.1.2	Publication Rights Control	81
5.3.2.1.3	Subscription Rights Control	85
5.3.2.1.4	Publication Port Control	90
5.3.2.1.5	Subscription Port Control	92
5.3.2.1.6	Sending Messages	95
5.3.2.1.7	Receiving Messages	99
5.3.2.1.8	Filtering Messages	105
5.3.2.2	MST Controller Function	110
5.3.2.3	Checkpoint / Bookmark Function	113
5.3.2.4	Configuration Parameter Processing Functions	114
5.3.2.4.1	Command Line Parameter Processing Support	115
5.3.2.4.2	Configuration File Parameter Processing	117
5.3.2.5	Executive Services Functions	121
5.3.3	Outputs	121
5.3.3.1	Messages Sent	121
5.3.3.2	Other Output Data Items	122
5.4	Instructional Environment	122
5.4.1	Inputs	122
5.4.1.1	Messages Received	122
5.4.1.2	Other Input Data Items	126
5.4.2	Algorithms	126
5.4.2.1	System Control Function	126
5.4.2.2	Checkpoint / Bookmark Function	127
5.4.2.3	Student Action Handling Function	127
5.4.2.3.1	Student Troubleshooting Action Handling Function	127
5.4.2.3.2	Student Request for Information Handling Function	128
5.4.2.4	Unsolicited Advice Determination Function	130
5.4.2.5	Coaching Coordination Function	132
5.4.2.6	Review Coordination Function	134
5.4.2.7	Reflective Follow-up Coordination Function	134
5.4.3	Outputs	136
5.4.3.1	Messages Sent	136
5.4.3.2	Other Output Data Items	138
5.5	Simulator	139
5.5.1	Inputs	139
5.5.1.1	Messages Received	139
5.5.1.2	Other Input Data Items	140
5.5.2	Algorithms	140
5.5.2.1	System Control Function	140
5.5.2.2	Checkpoint / Bookmark Function	141

5.5.2.3	State Maintenance Function	141
5.5.2.4	Initialization Function	143
5.5.2.5	State Reporting Function	145
5.5.2.6	Simulator Control Function	147
5.5.3	Outputs	149
5.5.3.1	Messages Sent	149
5.5.3.2	Other Output Data Items	150
5.6	Expert Model	151
5.6.1	Inputs	151
5.6.1.1	Messages Received	151
5.6.1.2	Other Input Data Items	153
5.6.2	Algorithms	156
5.6.2.1	System Control Function	156
5.6.2.2	Checkpoint / Bookmark Function	157
5.6.2.3	Student Action Interpretation Function	157
5.6.2.4	Troubleshooting Procedure and Goal Maintenance Function	159
5.6.2.5	Troubleshooting Action Selection Function	161
5.6.3	Outputs	162
5.6.3.1	Messages Sent	162
5.6.3.2	Other Output Data Items	164
5.7	Student Model	165
5.7.1	Inputs	165
5.7.1.1	Messages Received	165
5.7.1.2	Other Input Data Items	166
5.7.2	Algorithms	167
5.7.2.1	System Control Function	167
5.7.2.2	Checkpoint / Bookmark Function	168
5.7.2.3	Problems Attempted Storage / Retrieval Function	168
5.7.2.4	Student Actions Storage / Retrieval Function	168
5.7.2.5	Student Performance Measurement Storage / Retrieval Function	169
5.7.2.6	Student Action / Performance Variable Mapping Function	169
5.7.3	Outputs	170
5.7.3.1	Messages Sent	170
5.7.3.2	Other Output Data Items	172
5.8	Explanation Generator	173
5.8.1	Inputs	173
5.8.1.1	Messages Received	173
5.8.1.2	Other Input Data Items	173
5.8.2	Algorithms	173
5.8.2.1	System Control Function	174
5.8.2.2	Checkpoint / Bookmark Function	174
5.8.2.3	Explanation Generation Script Construction Function	175

5.8.3	Outputs	176
5.8.3.1	Messages Sent	176
5.8.3.2	Other Output Data Items	176
5.9	Presentation Controller	176
5.9.1	Inputs	176
5.9.1.1	Messages Received	176
5.9.1.2	Other Input Data Items	176
5.9.2	Algorithms	177
5.9.2.1	System Control Function	177
5.9.2.2	Checkpoint / Bookmark Function	178
5.9.2.3	Input Script Translation / Processing Function	178
5.9.2.4	Temporally and Spatially Related Media-bite Access Function	178
5.9.2.6	Option Translation Function	179
5.9.2.7	Option Availability Maintenance Function	179
5.9.3	Outputs	180
5.9.3.1	Messages Sent	180
5.9.3.2	Other Output Data Items	182
5.10	Multimedia Interface	182
5.10.1	Inputs	182
5.10.1.1	Messages Received	182
5.10.1.2	Other Input Data Items	182
5.10.2	Algorithms	182
5.10.2.1	Window Control / Display Function	182
5.10.2.2	Text Display Function	185
5.10.2.3	Graphics Display Function	186
5.10.2.4	Video Display / Control Function	186
5.10.2.5	Audio Display / Control Function	187
5.10.2.6	Student Interaction Processing Function	187
5.10.3	Outputs	187
5.10.3.1	Messages Sent	187
5.10.3.2	Other Output Data Items	187
5.11	Curriculum Manager	188
5.11.1	Inputs	188
5.11.1.1	Messages Received	188
5.11.1.2	Other Input Data Items	188
5.11.2	Algorithms	189
5.11.2.1	System Control Function	189
5.11.2.2	Checkpoint / Bookmark Function	189
5.11.2.3	Student Mastery Determination Function	190
5.11.2.4	Problem Selection Mode Determination Function	190
5.11.2.5	Problem Selection Function	190
5.11.2.6	Problem Creation Function	194

5.11.3	Outputs	195
5.11.3.1	Messages Sent	195
5.11.3.2	Other Output Data Items	195
5.12	Training Administrator	195
5.12.1	Inputs	195
5.12.1.1	Messages Received	195
5.12.1.2	Other Input Data Items	196
5.12.2	Algorithms	196
5.12.2.1	System Control Function	196
5.12.2.2	Checkpoint / Bookmark Function	196
5.12.2.3	New Training Administrator Registration Function	197
5.12.2.4	Data Recording Function	197
5.12.2.5	Student Data Transfer Function	197
5.12.2.6	Recorded Data Analysis Function	197
5.12.3	Outputs	197
5.12.3.1	Messages Sent	197
5.12.3.2	Other Output Data Items	197
List of References		199
Glossary		203
Program design language (PDL) standards		211
A.1	PDL Documentation Standards	211
A.1	TBD Package Specification	211
Common Software Component Architecture		213
B.1	MST Controller	213
B.1.1	Behavioral Description	213
B.1.2	MST Controller PDL	213
B.2	Software Component Architecture	224
B.2.1	Initialization Function	224
B.2.2	Termination Function	224
B.2.3	Checkpoint/Bookmark Function	224
B.2.4	Main Processing Loop	226
B.2.4.1	Primary Message Receipt and Dispatch	226
B.2.4.2	Secondary Sequencing	226
B.2.3	Common Software Component PDL	226
Message Dictionary		237
C.1	Standard Message Classes	237
C.1.1	System Message Classes	237

C.1.2	Tutor Message Classes	237
C.2	Standard Message Specifications	237
C.2.1	System Messages	237
C.2.2	Tutor Messages	238
	CONFIGURATION PARAMETER FILE FORMATS	239
D.1	Configuration Parameter File Purpose	239
D.2	Configuration Parameter File Structure	239
D.2.1	Configuration Segments	239
D.2.2	Segment Options	240
D.2.3	Supported Value Types	240
D.2.3	Extensibility	241
D.3	Configuration Parameter File Grammar	241
	MST package specifications	245
E.1	Package Specification Relationships	245
E.2	Executive Package Specifications	245
E.2.1	EXECUTIVE Package Specification	245
E.2.2	MESSAGING Package Specification	247
E.2.3	MSG_CLASSES Package Specification	273
E.2.4	MSG_FORMATS Package Specification	275
E.2.5	MSG_TYPES Package Specification	279
E.2.6	MST_CONTROL_VALUES Package Specification	283
E.2.7	MST_STATES Package Specification	283
E.2.8	SW_COMPONENTS Package Specification	284
E.2.9	TUTOR_MESSAGING Package Specification	285
E.3	Component-Specific Package Specifications	285
E.4	Utility Package Specifications	286

## LIST OF FIGURES

FIGURE	PAGE
Components of an ITS .....	4
Components of the MST .....	9
The Executive .....	14
The Instructional Environment .....	23
The Simulator.....	33
The Expert Model .....	40
The Student Model.....	45
The Explanation Generator .....	48
The Presentation Controller .....	51
The Multimedia Interface .....	55
The Curriculum Manager.....	61
The Training Administrator .....	67
Example Equipment Component Representation .....	72
Example Equipment Component Hierarchy .....	73
Example Active Path Representation .....	74
Minimal Startup Protocol.....	111
Minimal Restart Protocol.....	112
Minimal Shutdown Protocol.....	113
Minimal Checkpoint/Bookmark Protocol.....	114
Minimal Simulation Initialization Protocol .....	144
Minimal State Reporting Protocol .....	146
Minimal Simulation Control Protocol .....	148
Sample Troubleshooting Goal/Procedure/Action Dependency Network .....	156
Sample Student Modeling Variable Dependency Network.....	167
Example Checkpoint/Bookmark Data Classes .....	225

## LIST OF TABLES

<b>TABLE</b>	<b>PAGE</b>
Send Message Capabilities	17
Receive Message Capabilities	18
Simulation Control Commands	38
Sample Student Critiquing Strategies	131
Sample Explanation Content Strategies	133
Sample Window Manager Widgets	185

## **SECTION 1**

### **INTRODUCTION**

This section describes the purpose of the Maintenance Skills Tutors (MST) and provides a brief overview of the program and the cognitive research behind it. This section also describes the purpose and content of this document.

#### **1.1 The Maintenance Skills Tutor Program**

TAC has identified a need to improve troubleshooting skills for technicians performing flightline maintenance. These skills are normally taught through on-the-job training (OJT). This method is both time consuming and manpower intensive, requiring years of apprenticeship training under the tutoring of experienced maintenance technicians. Today's highly reliable systems fail less often and thus provide fewer opportunities to train technicians on the complex problems. At the same time, the Air Force has combined maintenance specialties and reduced the number of technicians per aircraft. Thus, technicians must have broader knowledge and more general skills.

In response to this need, the Human Resources Directorate (HRD) of Armstrong Laboratory is performing research under the Basic Job Skills (BJS) program in two areas: cognitive task analysis techniques to capture troubleshooting strategies used by experts and novices, and computer-based Intelligent Tutoring Systems to provide a practice environment for working authentic troubleshooting problems while coaching the student with hints and feedback. Prototype tutors were demonstrated at TAC fighter wings at Langley and Eglin Air Force Bases. The results showed significant gains in proficiency by novice technicians. TAC has identified this technology as a high priority for fielding.

The MST program will develop and field operational tutors for TAC based upon the BJS technology. These tutors will focus on teaching advanced troubleshooting skills for 5- and 7-level, F-15 and F-16 flightline maintenance technicians.

#### **1.2 The Basic Job Skills Program**

The BJS program has focussed on the cognitive skills involved in the execution of maintenance tasks requiring knowledge of and interaction with technologically complex systems. The BJS goals include the development of a skill analysis/instructional design technology that will promote the rapid transfer of skills necessary for the maintenance of complex systems. The instructional philosophy of the BJS program is to provide an apprenticeship training environment in which students are responsible for their own learning and coaching is provided to students primarily when requested by the student. The amount of information provided in

coaching is controlled to encourage student's to think things through on their own.

This research has produced methods for both expert knowledge acquisition (i.e., cognitive task analysis), problem scenario definition, and coaching strategies for complex maintenance tasks. As part of this research, several prototypes were developed. The Sherlock prototype, developed by The Learning Research and Development Center (LRDC) of the University of Pittsburgh, instructs maintenance technicians in troubleshooting a manual avionics test station. This prototype was used as the starting point for the functionality described in this document.

The products of the BJS program that this document references are the draft "A Procedural Guide to cognitive task analysis: the PARI Methodology" (hereafter referred to as the PARI Guide) and the draft "BJS Intelligent Tutor Development Guide" (hereafter referred to as the Tutor Development Guide).

### **1.3 Contents**

This document is meant to serve as a guide for a software engineer in designing and implementing the functionality required by the MST System Specification [MST Spec 92]. It specifies a generic software architecture for the Maintenance Skills Tutors. A distinction should be made at this time between the BJS products, their content, and this document. The PARI Guide provides a methodology for acquiring the domain-dependent information for incorporation into a tutoring system. The Tutor Development Guide will provide both an enumeration of BJS instructional design strategies and examples of their interaction with the domain-dependent instructional content acquired through a cognitive task analysis (such as the method described by the PARI Guide.) This Software Engineering Reference Guide will provide a software engineer with a software architecture description that can be instantiated with both the instructional content and instructional design strategies to form an intelligent tutoring system<sup>1</sup>.

The rest of this document is separated into four sections. Each section contains progressively more detail with respect to the functionality to be provided by the MST. Section 2 describes the components of an Intelligent Tutoring System. The purpose of this section is to define terms and organize the high level functionality required by an ITS. Section 3 describes the proposed software architecture for the MST. This section describes variations on the standard architecture described in section 2. Section 4 provides a functional description, in terms of the input, processing, and output required for implementation of each of the components outlined in section 3. Section 5 describes in detail the behavior of each component outlined in section 3. These behavioral descriptions

---

<sup>1</sup> It is worthy of note that, at this writing, the software architecture to be described herein has not been operationally prototyped.

include the enumeration of critical messages and algorithms that can be expressed independent of tutor subject matter (domain), that will implement the functionality described in section 4.



## SECTION 2

### FUNCTIONAL OVERVIEW OF INTELLIGENT TUTORING SYSTEMS

Intelligent Tutoring Systems (ITSs) are computer-based training systems that incorporate advanced techniques for communicating/transferring knowledge and skills to students. These systems emerged from the combination of Computer Aided Instruction<sup>2</sup> (CAI) and Artificial Intelligence (AI) technology. This section discusses the inception of ITSs and describes the software components of an ITS.

#### 2.1 Evolution of Computer Assisted Instruction

Computer Aided Instruction began in the 1950's as a means to provide automated individualized instruction to students. However, early systems did little more than incrementally replay canned text while waiting for a response from the student. As CAI evolved, instructional systems made use of a student's responses throughout an instructional session to determine the material to present during the remainder of the session. Advances in CAI technology allowed later systems to automatically generate teaching material. However, even with advances in CAI technology, sophisticated computer programs still did not approach the capability of human instructors. One reason for this shortcoming was that CAI systems lacked the means to accurately model a student's knowledge (i.e., they lacked a model of the student's cognitive processes, and therefore could not provide a robust individualized training environment).

ITSs emerged in the 1970's to address the deficiencies of CAI. By making use of the results of research work in AI, ITSs were able to employ knowledge representation strategies to model a student's cognitive processes. Using an accurate models of the student's and expert's knowledge, an ITS is able to provide instruction at the appropriate pace and level of abstraction for the student.

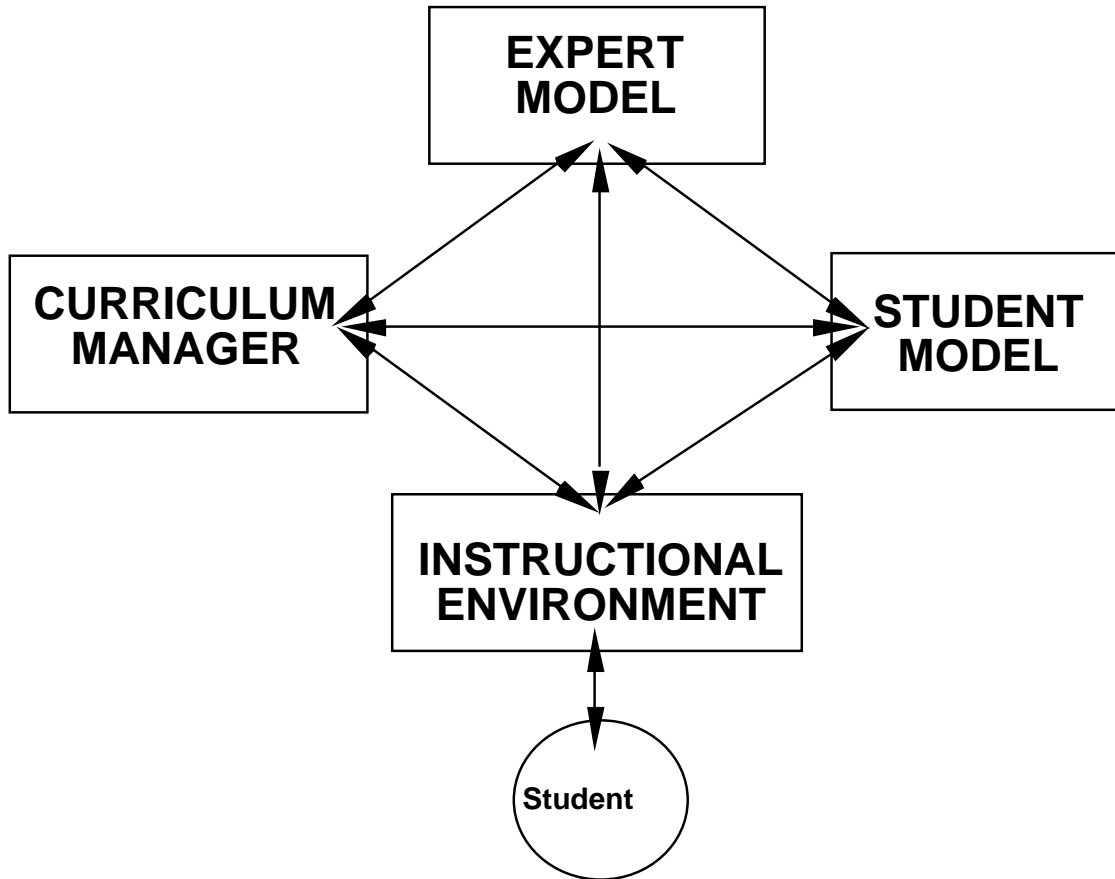
#### 2.2 Software Components of ITSs

Although there is no standard architecture for an ITS, four software components emerge from the literature as part of an ITS ([Burns & Capps 88], [Gott & Pokorny 87], [Massey 86]). These are the Expert Model, the Student Model, the Curriculum Manager, and the Instructional Environment. These four software components and their interactions are

---

<sup>2</sup> This document will refer to Computer Aided Instruction and Computer Based Training (CBT) as analogous technologies.

illustrated in figure 1.



**Figure 1. Components of an ITS**

Like a human expert, the Expert Model in an ITS has knowledge about a particular domain. The type of knowledge maintained by the Expert Model is referred to as domain or content knowledge. Typically, this knowledge is both factual and procedural, and is maintained in databases<sup>3</sup> by an expert system. A factual database stores pieces of information about the problem domain, while a procedural database contains knowledge of procedures and rules that an expert uses to solve problems within that domain. Although factual and procedural databases may adequately model knowledge in an expert system, a method of knowledge encoding known as cognitive, or qualitative, modeling provides for a closer simulation of the human expert's reasoning process. The Expert Model in an ITS may employ cognitive

---

<sup>3</sup> The term *database* is used throughout this document to refer to a collection of (usually persistent) information.

modeling by using structured knowledge of causality and human-like inference mechanisms.

While the Expert Model captures the knowledge that an expert uses in problem solving, the student model contains measurements of the student's knowledge of the problem area. Ideally, the Student Model can be thought of as containing an advanced profile of the student. The accuracy and detail of this profile are determined by the bandwidth of the Student Model (i.e., the quality and quantity of the input to the model). The bandwidth determines the granularity at which the student's actions can be tracked. Fine granularity provides more information to the Student Model to infer details about the student's understanding of the problem domain.

An ITS would not be a tutoring system if it did not contain facilities for teaching. Problems, or exercises, are the vehicle that an ITS uses to instruct the student. By solving problems, the student builds upon concepts already mastered. The facility in the ITS for sequencing and selecting problems is the Curriculum Manager. To select the appropriate problems for the student, the Curriculum Manager extracts performance measurements from the profile stored in the Student Model.

Teaching involves more than presenting material to the student. An effective instructor monitors a student's progress and provides coaching when the student requests assistance or is struggling. Like a human instructor, an ITS coaches the student through the use of an Instructional Environment. It is the Instructional Environment that provides the student with tools for proceeding through a tutorial session and obtaining help when needed. The Instructional Environment also determines when the student needs unsolicited advice and triggers its display.

These four components, the Expert Model, the Student Model, the Curriculum Manager, and the Instructional Environment interact to provide the individualized educational experience promised by ITS technology.

