DRAFT

Digitally signed by Rene Salazar
DN: CN = Rene Salazar, C = US, O = Intellectual Tech., Inc.
Reason: I attest to the accuracy and integrity of this document
Location: San Antonio, Texas
Date: 2004.07.30 16:14:21 -05'00'

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

LIST OF FIGURES

FIGURE **PAGE**

LIST OF TABLES

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[1].

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

---

1 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[2] (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

---

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

illustrated in figure 1.

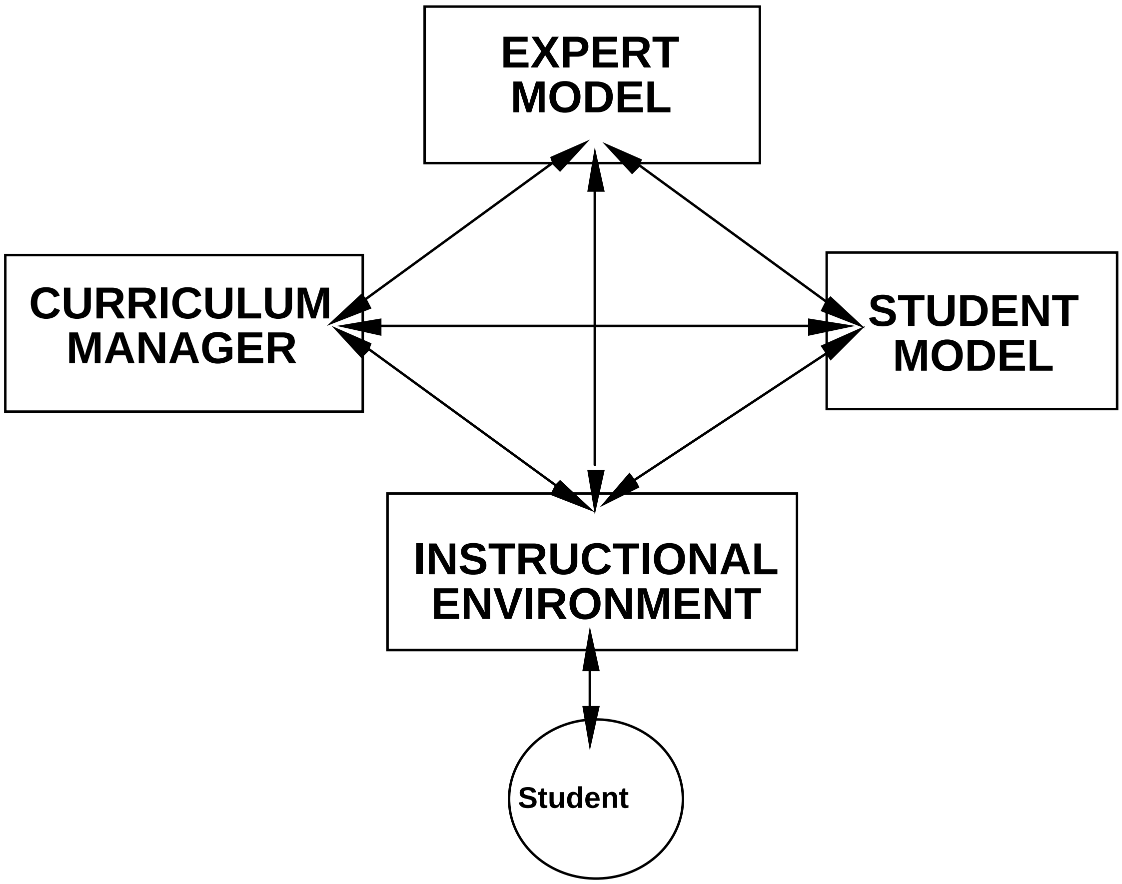

Figure 1. Components of an ITS

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[3] 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

3 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.