In the Ninth International Conference on Machine Learning, Workshop on Integrated Learning in Real-world Domains, Aberdeen, Scotland, July 1992.

Learning to Troubleshoot in Electronics Assembly Manufacturing

S. Narayanan
School of Industrial & Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205
sn@chmsr.gatech.edu

Ashwin Ram
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
ashwin@cc.gatech.edu

Abstract

This paper presents a case study on the application of multistrategy learning methods through introspective reasoning in the complex task of troubleshooting in electronics assembly manufacturing. The paper describes a computational model of problem solving and learning which was built based on observations of troubleshooting operators and protocol analysis of the data gathered in the test area of an operational manufacturing plant. The learning model is based on the theory of *Meta-XPs*, which are meta-explanation structures that help the system identify reasoning failures, formulate learning goals, and choose appropriate learning strategies in coping effectively with the complexity of the domain.

1. INTRODUCTION

There has been a significant growth of research on machine learning (see, e.g., Shavlik & Dietterich [1990] for a broad introduction to the methods developed in this field). However, most of this research has not been performed in the context of complex real-world problem solving tasks. As a result, the issues of scalability and robustness of these methods, as they are applied to real-world problems, are still unresolved in many cases. To promote the applicability and usability of research methods, it is important to ground theories of reasoning, knowledge representation, and learning in the context of real-world tasks and domains.

We present a computational model of problem solving and learning, which was built based on observations of troubleshooting operators and protocol analysis of the data gathered in the test area of an operational electronics assembly manufacturing plant. The model is implemented in a computer system, which uses multiple types of knowledge to troubleshoot printed-circuit boards that fail in the test area. The system then analyzes its reasoning process introspectively, and uses multiple learning strategies to learn the associative and heuristic knowledge that human operators use for this problem solving task. This model is based on the theory of *Meta-XPs*, which are explanation structures that help the system identify types of reasoning failures, formulate learning goals, and choose appropriate learning strategies in coping effectively with the complexity of the domain [Cox & Ram, 1991, 1992; Ram & Cox, 1992].

We begin with a description of the case study. We briefly discuss the problem solving model designed for the task. We then provide a rationale for incorporating a learning module in the system and describe the architecture of this module. Finally, we outline the various components in the learning module, discuss the implementation, and conclude with a summary of related research.

2. CASE STUDY DESCRIPTION

NCR's electronic assembly manufacturing plant located near Atlanta has state-of-the-art facilities in electronics assembly manufacturing with a newly installed Surface Mount Technology (SMT) line. Our project began when the plant became operational in January 1990 [Cohen, 1990]. At that time, the plant was facing typical start-up problems experienced by most new facilities. There were a high number of printed-circuit boards (PCBs) in the test and repair region waiting to undergo troubleshooting, resulting in high work-in-process (WIP) inventories. Our analysis of the system revealed that developing a model of the troubleshooting operator would provide a structure for designing and implementing computer-based tools for this task. The effort would also facilitate in formalization of the task, which could then be used in the design of instructional systems [Clancey, 1987], thereby facilitating the development of a flexible work force that is complementary to the policy of the company.

A schematic of the manufacturing plant is shown in Figure 1. An unpopulated board enters the SMT line where it is populated with components, soldered, cleaned, and sheared. The populated board then enters the test and repair area. Boards are shipped only after they pass both the in-circuit test (ICT) area and the functional test area. Our research focuses on the troubleshooting process when a board fails in the ICT area.

In the ICT area, the board is mounted on an ICT machine. The ICT machine checks individual components as well

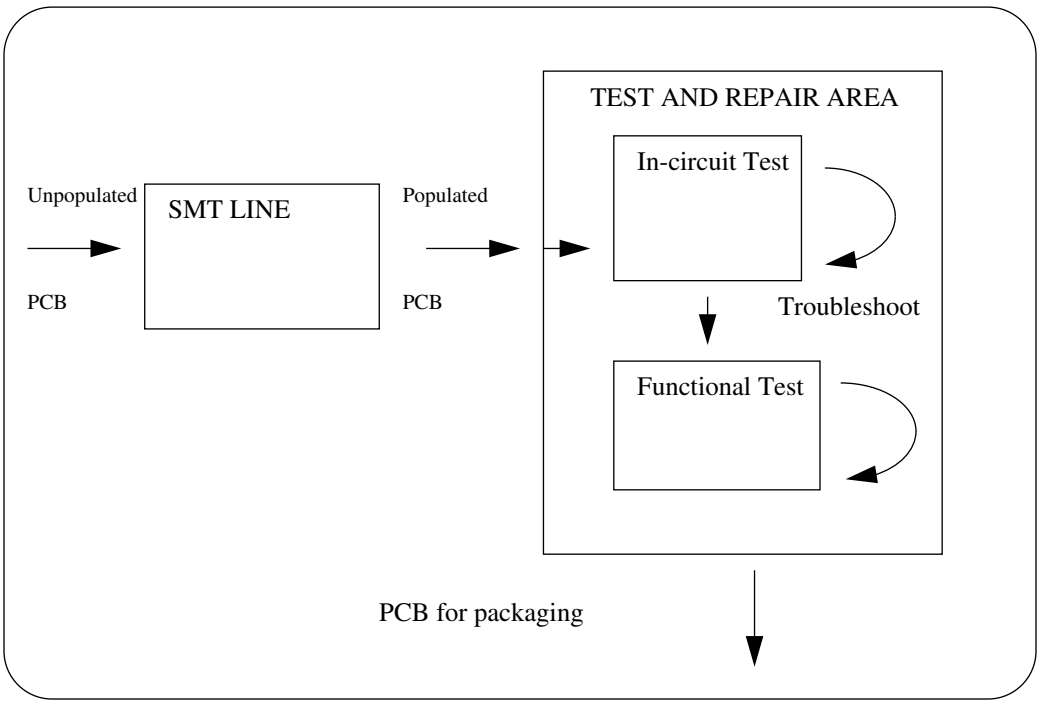


Figure 1. A schematic of the electronics assembly manufacturing plant.

as connections between components for proper functioning. If the board passes the tests, an appropriate message appears on the console of the ICT machine. When the board fails any of the tests, the ICT machine produces a ticket listing the detected failure(s). The operator uses information in the ticket and troubleshoots the board. Troubleshooting is a complex task which can be broken into two components: diagnosis and repair. Diagnosis is a problem solving task where the operator arrives at an appropriate set of repair actions to fix the faulty PCB and repair is the manual action performed by the operator on the board.

3. PROBLEM SOLVING MODEL

A schematic of the problem solving model is shown in Figure 2 [Narayanan et al., 1992]. The model takes as input the ICT ticket information and the PCB information. The output of the system is a set of diagnostic repair actions. The knowledge used by human operators can be broadly categorized into *associations* and *heuristics*. Associations are symptomatic rules using which the operator forms a solution based directly on the symptom cue. S/he may perform an intermediate action to confirm her/his hypothesis, but does not perform a series of search sequences. This type of knowledge is context-sensitive and is indexed by board type. Heuristic knowledge is a set of standard rules of thumb. They are not context-sensitive and are applicable across board types. Heuristics are used by the operator for troubleshooting when there is no known association for a given problem situation. This knowledge determines the series of standard operating procedures performed in troubleshooting a faulty PCB.

Actions are intermediate subtasks performed by the system to gather the board information. These correspond to explicit operator actions used in confirming intermediate hypothesis. The *control methods* in the model are a collection of procedures that enable the system to look at the symptoms, utilize the appropriate type of knowledge, invoke proper intermediate actions, and finally arrive at the diagnosis result. The *diagnostic* is the repair action to be taken by the operator in fixing the faulty PCB. Example of the major components in the model is shown in Figure 3.

The computational model of problem solving is implemented in a prototype system that runs on a SUN workstation under the UNIX operating system. The system was validated using about 75% of the data available from the verbal protocol analysis for a major category of board failures. 84% of the time, the model arrived at the same diagnostic result as an operator did in the real world for the same input information; and 68% of the time, apparently similar actions were performed in the solution process. Analysis of the results revealed the need for a learning component in the system, discussed further in the

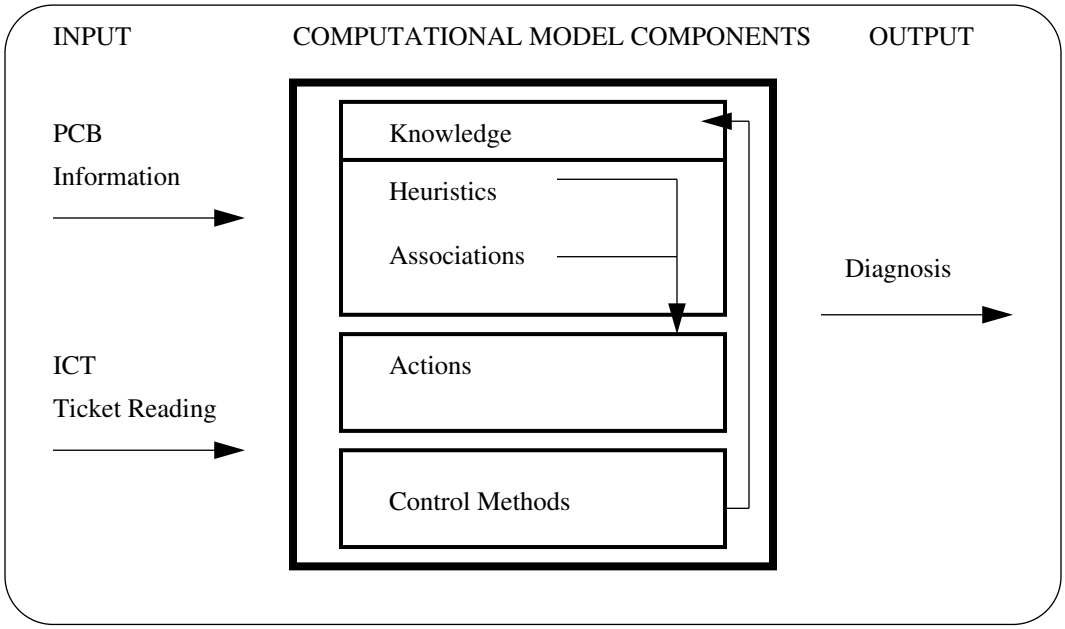


Figure 2. A schematic of the diagnostic problem solving system.

ASSOCIATION
Part r254 is often damaged. Visually inspect the part. If it is damaged, replace the part.

HEURISTIC
If the measured reading of the resistor is slightly higher or much lower than the nominal value on the ICT ticket, perform visual inspection. If the defect is found, terminate the search. Else, terminate with no inference.

ACTION
Ohm out on a resistor.

CONTROL METHOD
Find symptom information from the ICT ticket. If there is an association for the given symptom, invoke it, and terminate the search.

DIAGNOSTIC

Figure 3. Example of the major components in the problem solving model.

next section.

4. LEARNING MODEL

RATIONALE

Although the system was a fair representation of a skilled troubleshooting operator, it had some limitations. First, the system assumed that its knowledge was correct and complete during the reasoning process. It is difficult to hand code all the knowledge required for this task. Furthermore, even if this could be done, the system would still be faced with the "brittleness" problem. In the electronics manufacturing domain, due to the dynamics of the system state changes, there is need for a computational model with a high degree of robustness. For example, one of the associations in the system is "r254 is often damaged". Suppose this occurred due to a process problem in the plant. When the process problem is fixed, the association is no longer valid. The system must have the capabil-

world. Second, the system did not capture improvement in the problem solving skills of the operator. Thus, there is the problem of the plausibility of our model as a cognitive model of human troubleshooting.

These considerations motivated our research towards incorporation of a learning module in the system. We have been developing a computational model of learning based on observations of troubleshooting operators in the plant, protocol analysis of the problem solving process, and critical examination of the computational model of the troubleshooting operator. Our model is intended both as a cognitive model of human learning during the troubleshooting task, and as a computational solution to the brittleness problem which is faced by all systems in a dynamic world.

WHAT IS TO BE LEARNED?

Analysis of the computational model of troubleshooting revealed that associative knowledge improves the system performance in two ways. First, it improves the speed of the problem solving process. Using associative knowledge typically results in the reduction of some intermediate steps in the reasoning process, thus resulting in some savings in the time required to troubleshoot. Second, associative knowledge can provide solutions in cases where heuristic knowledge requires information about the board that is not easy to obtain. For resistor failures, associative knowledge contributed to either the quality or correctness of solution for over 25% of the problem solving situations. This indicates that an important type of learning is one in which the operator learns associations through experience. We have found evidence for this type of learning in our observations of human operators. For example, we noticed that operators in the plant paste a yellow tag on their table when they notice recurring occurrences of a problem, thereby acquiring an association. Both supervised learning and unsupervised learning mechanisms appear to play a role in the learning process. Supervised learning corresponds to situations in which a novice troubleshooter gets an explicit input from a skilled troubleshooter. Unsupervised learning corresponds to situations in which a troubleshooter alters his/her knowledge based on his/her problem solving experience.

ARCHITECTURE OF THE LEARNING MODULE

Analysis of the learning situations for this task also reveals the need for methods for blame assignment and repair. Since a particular problem solving episode may involve several pieces of knowledge, potentially even of different types, the system must be able to examine the reasons for its successes and failures. When the system fails, it needs to determine which knowledge was missing or incorrect. Depending on the type of failure, the system then needs to invoke the learning strategy appropriate for acquiring the required type of knowledge. The system uses introspection to analyze what went wrong, determine what needs to be learned, and select and integrate the learning strategies to be used. Thus, learning is viewed as a deliberative, planful process in which the system makes explicit decisions about what to learn and how to learn it [Hunter, 1990; Ram, 1991; Ram & Cox, 1992; Ram & Hunter, 1992]. Introspection requires declarative representations of meta-knowledge about the reasoning performed by the system during the performance task, about the system's knowledge, and about the organization of this knowledge.

The key representational entity in our architecture is a meta-explanation pattern (Meta-XP), which is a causal, introspective explanation structure that explains how and why an agent reasons, and that helps the system in the learning task [Cox & Ram, 1991, 1992; Ram & Cox, 1992]. There are two broad classes of Meta-XPs. *Trace Meta-XPs* record a trace of the reasoning performed by a system, along with causal links that explain the decisions taken. *Introspective Meta-XPs* are structures used to explain and learn from a reasoning failure. They associate a failure type with a particular set of learning strategies and point to likely sources of the failure within the Trace Meta-XP.

In our multistrategy learning approach, the reasoning system records a declarative trace of its own reasoning process using a Trace Meta-XP. The data structure holds explicit information concerning the manner in which knowledge gaps are identified, the reasons why particular hypotheses are generated, the strategies chosen for verifying candidate hypotheses, and the basis for choosing particular reasoning methods for each of these. If the system encounters a reasoning failure, it then uses Introspective Meta-XPs to examine the declarative reasoning chain. An Introspective Meta-XP performs three functions: it aids in blame assignment (determining which knowledge structures are missing, incorrect or inappropriately applied); it aids in the formulation of appropriate learning goals to pursue; and it aids in the selection of appropriate learning

algorithms to recover and learn from the reasoning error. Such meta-explanations augment a system's ability to introspectively reason about its own knowledge, about gaps within this knowledge, and about the reasoning pro-

Meta-XP structures allow direct inspection of the reasons by which learning goals are posted and processed, thus enabling a system to improve its ability to reason and learn.

The modules and the control flow in the learning system are shown in Figure 4. The problem solving module has a declarative representation of the associative knowledge used in troubleshooting. The introspective learning module can add, delete, or modify associative knowledge in the problem solving module. During a troubleshooting episode, a trace of the reasoning performed by the system along with causal links that explain the intermediate decisions taken is recorded in an instance of a Trace Meta-XP. The *introspector* takes the reasoning trace as input. It also has methods to perform tests on the world. These methods access the ICT information and the PCB information which is input to the problem solving system. The introspector has a set of verification actions that includes methods to obtain statistical information and to gather information from an expert troubleshooter. There is also a

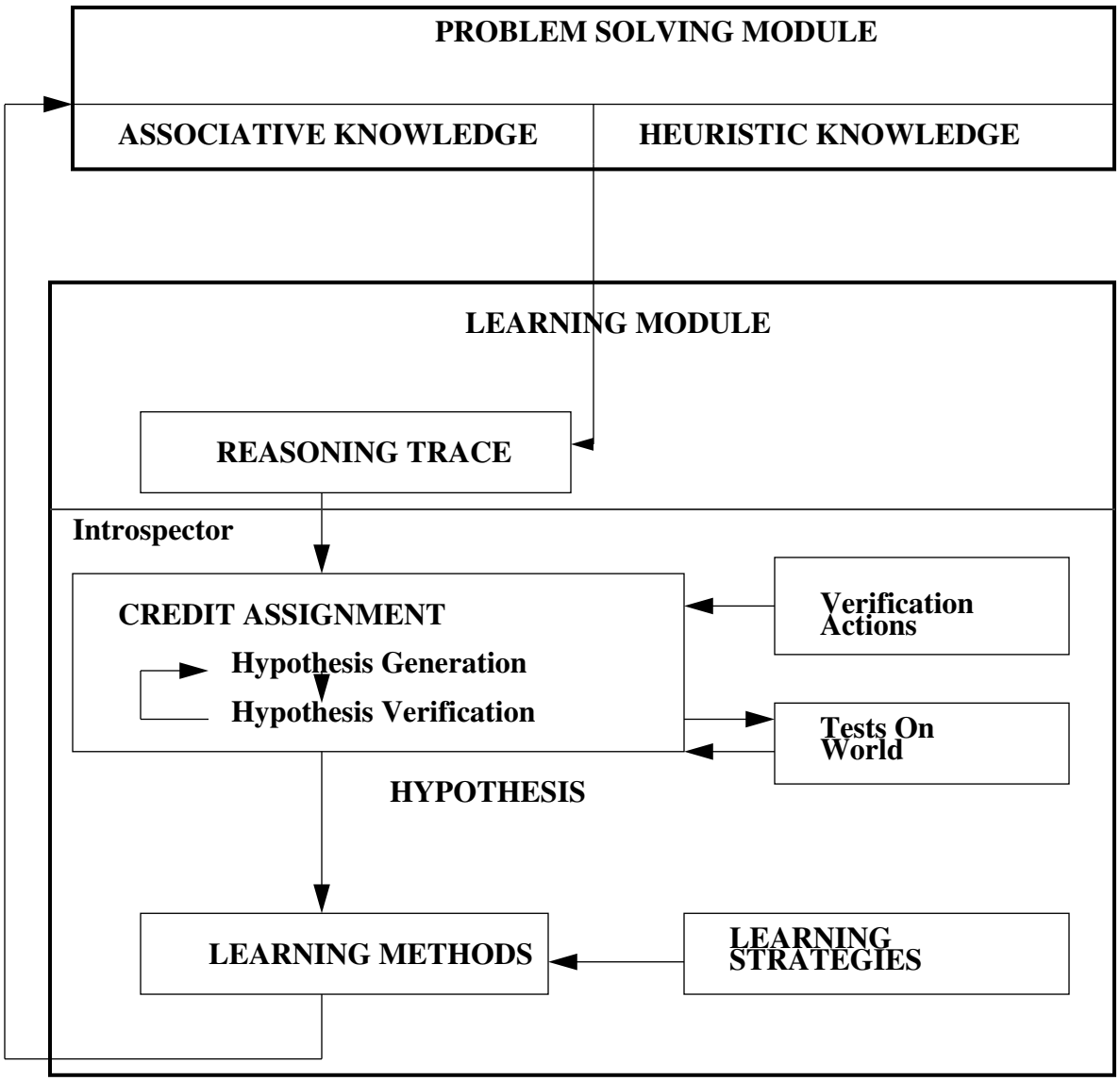


Figure 4. Modules and control flow in the learning system.

set of learning strategies represented declaratively along with information for strategy selection.

After every problem solving episode, the introspector looks at the reasoning trace and uses information gathered from the tests on the world to determine if the system can learn something from this experience. This occurs when the system fails to make the correct diagnosis (due to missing or incorrect knowledge) or the system can learn new knowledge that would allow it to arrive at the correct diagnosis more efficiently. When the introspector determines that something is to learned, it performs the task of hypothesis generation which results in a hypothesis or explanation of the system's reasoning process. This identifies the knowledge structure to be learned or improved. Hypothesis generation is interleaved with the hypothesis verification process. During this process, the introspector uses information gathered from both the verification actions and tests on the world to test and verify candidate hypotheses. Finally, based on what needs to be learned, an appropriate learning strategy is triggered, which results in the modification of existing knowledge in the problem solving system.

IMPLEMENTATION AND RESULTS

The system is implemented using AT&T 2.0 C++ and runs on a SUN workstation under the UNIX operating system. Currently, the system has two strategies for learning associative knowledge in the task. One strategy is that of incremental inductive learning where the introspector forms an association when the problem solving model arrives at the right solution using heuristic knowledge. The introspector compiles the heuristic knowledge into an association. This is similar to "knowledge compilation" presented in Anderson [1989]. The motivation for this type of learning in the introspector is performance gain through reduction of the number of intermediate steps when the system encounters a similar problem in the future. The second strategy is learning of an association through supervisory input. This strategy is triggered when the system arrives at an incorrect solution using heuristic and/or associative knowledge. The system attempts to acquire correct associative knowledge from a skilled troubleshooter. This mechanism is similar to the interactive transfer of expertise in TEIRESIAS [Davis, 1979]. However, the knowledge learned in our system is not in the form of production rules, but in the form of frames and slots for association records. Example program transcripts illustrating these strategies are shown in figures 5 and 6.

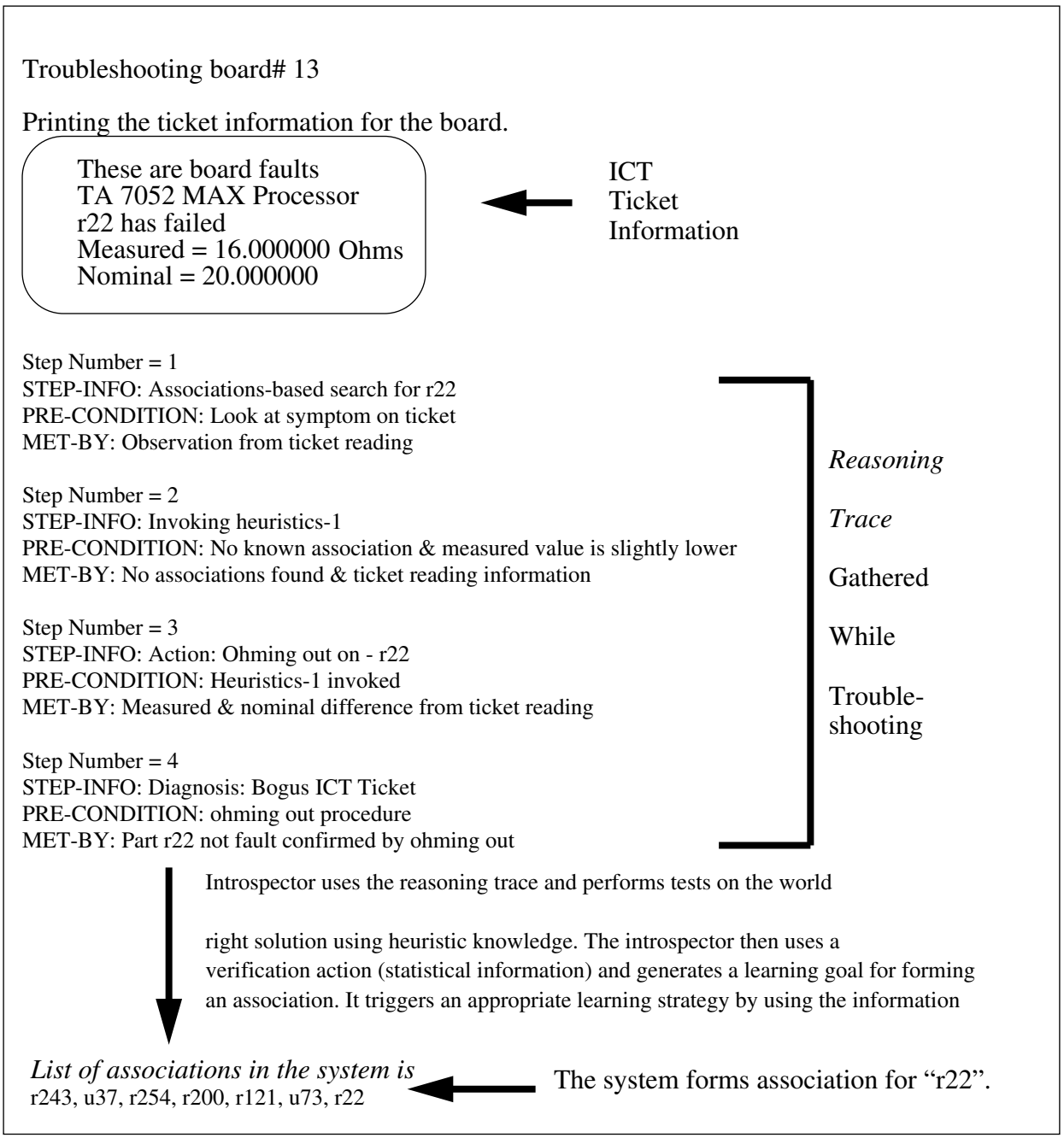
Preliminary results on the learning strategies are encouraging. We removed all the associative knowledge in the system, ran the system with the learning module for 30 problem solving cases taken from the plant data and compared the performance of the system with and without learning. The first strategy enabled the system learn 30% of the hand-coded associations. The contribution of the second learning strategy is dependent on the type of input given by a skilled troubleshooter. Analysis of the results also revealed the need for other learning strategies. We are currently implementing the strategy of postponement [Ram, 1991]. This strategy is triggered when the system is unable to get an input from a skilled troubleshooter right away. The system posts a "learning goal", keeps track of the reasoning trace for the particular problem solving episode and asks questions at a later time to gather appropriate associative knowledge. Given the learning architecture in our system, it is relatively straightforward to incorporate additional strategies, both supervised and unsupervised, for learning both associations and heuristics.

5. DISCUSSION AND RELATED RESEARCH

Diagnostic problem solving has received considerable attention in the AI research community since the time of MYCIN [Buchanan et al., 1984]. Many recent research efforts in diagnostic problem solving have proposed the use of "deep reasoning methods" [Richardson et al., 1985] or integration of "deep" and "shallow" reasoning methods in knowledge-based systems [Fink et al., 1987] and in tutoring systems [Lesgold et al., 1988]. Our observations reveal that operators rely predominantly on "shallow" reasoning methods using heuristic and context-sensitive associative knowledge during problem solving. This is reflected in our computational model. This may be due to the fact that the ICT machine filters out most of the topographic knowledge of the PCB and causal knowledge of the components in the board through a series of tests. The observation by Barr and Feigenbaum [1981], that humans often solve a problem by finding a way to think about the problem that facilitates the search for a solution, was clearly evident in our study. This clearly influences what needs to be learned by the system.

There has been a significant growth of research on machine learning. However, most of this research has not been performed in the context of real-world complex problem solving tasks, and has focused on the use of a single learning strategy in a well-defined learning situation. We are developing a model of learning that can deal with the complexities of real-world troubleshooting, in which the system must deal with an incomplete world model, dynamic changes in the world which renders part of the world model obsolete, and multiple forms of knowledge (much of it shallow). Our model integrates multiple learning methods (inductive, analytical, and interactive) in both supervised and unsupervised situations using introspection. Furthermore, the architecture of introspective reasoning for strategy selection in multistrategy learning provides a general framework for integrating multiple learning methods. The learning strategies are not dependent on the domain, but are however dependent on the types of knowledge used in the performance task.

Recently, there has been much interest in multistrategy learning systems. However, such systems simply consist of a cascade of more than one learning algorithm (e.g.,



Ahmad et al., 1991; Flann & Dietterich, 1989; Hunter, 1991; Shavlik & Towell, 1989). The control is always the same for every input. Usually an initial learning technique is applied, and its output used as the input to the next algorithm. Some non-cascaded multistrategy learning systems can select particular algorithms to apply to different situations. Most such systems, however, try learning algorithms in a predefined order (e.g., Genest et al., 1990; Pazzani, 1991). If the first fails, the next strategy is tried, and so forth. One of the greatest benefits of using Meta-XPs in our framework is their ability to select and apply learning methods that are appropriate in a given situation in a flexible manner without resorting to exhaustive search ([Ram & Cox, 1992]; see also [Hunter, 1990; Ram & Hunter, 1992]).

From the system design point of view, the use of Meta-XPs in reasoning about learning goals during problem-

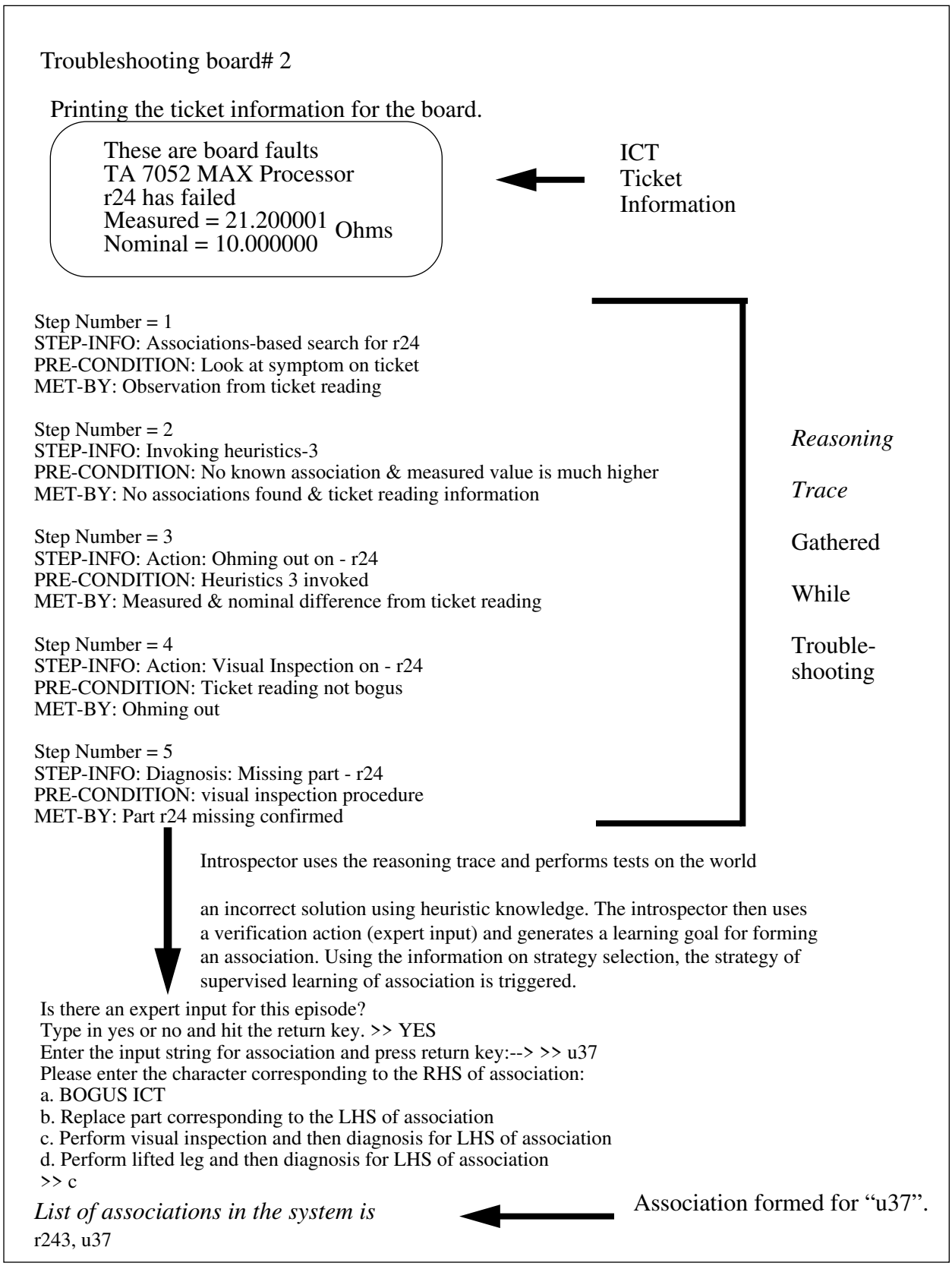


Figure 6. An example of the strategy of supervised learning of associations.

solving provides a number of benefits. Because Meta-XPs make the trace of reasoning explicit, an intelligent system can directly inspect the reasons supporting specific conclusions. With these reasons enumerated, a system can explain how it produced a given failure and can retrieve an introspective explanation of the failure. This approach provides a robust framework for integrated learning approaches to real-world problems in which reasons for processing and learning decisions are made explicitly by the system rather than existing only in the minds of the system's designers.

Acknowledgments

The problem solving model presented in this paper benefited from discussions with Sally Cohen, Christine Mitchell, T. Govindaraj, Ashok Goel, and Leon McGinnis at Georgia Tech. The ideas on integrating multiple learning strategies through introspection benefited from discussions with Mike Cox in the College of Computing at Georgia Tech. We thank Jim Wallace, Don Haywood, Mike Bunger, and Kenneth Mitchell of the manufacturing plant at Atlanta for their cooperation during this study.

References

A. Ahmad, S. Matwin, & H. Ould-Brahim, "Acquiring the second tier: An experiment in learning two-tiered concepts," In R.S. Michalski & G. Tecuci (eds.), *Proceedings of the First International Workshop on Multistrategy Learning*, pages 419--426, Harpers Ferry, WV, 1991.

J. Anderson, "A Theory of the Origins of Human Knowledge", *Artificial Intelligence*, Vol. 30, pages 313-351, 1989.

A. Barr & E. A. Feigenbaum (Eds.), *The Handbook of Artificial Intelligence*, Vol. 1, Reading, MA: Addison-Wesley, 1981.

B. G. Buchanan & E. H. Shortliffe, *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Menlo Park, CA: Addison-Wesley, 1984.

M. Cox & A. Ram, "Using introspective reasoning to select learning strategies," In R.S. Michalski and G. Tecuci (eds.), *Proceedings of the First International Workshop on Multi-Strategy Learning*, pages 217--230, Harpers Ferry, WV, 1991.

M. Cox & A. Ram, "Multistrategy Learning with Introspective Meta-Explanations," *Machine Learning: Proceedings of the Ninth International Conference*, Aberdeen, Scotland, 1992.

W. J. Clancey, *Knowledge-based tutoring: The GUIDON program*. Cambridge, MA: The MIT Press, 1986.

S. M. Cohen, "A Model of Troubleshooting in Electronics Assembly Manufacturing," M.S. thesis, Center for Human-Machine Systems Research, Georgia Institute of Technology, *CHMSR 90-3*, 1990.

R. Davis, "Interactive transfer of expertise: Acquisition of new inference rules," *Artificial Intelligence*, 12:121--157, 1979.

P. K. Fink, J. C. Lusth, "Expert Systems and Diagnostic Expertise in the Mechanical and Electrical Domains," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC - 17, No. 3, pages 340-347, May/June 1987.

N.S. Flann & T.G. Dietterich, "A study of explanation-based methods for inductive learning," *Machine Learning*, 4:187--226, 1989.

J. Genest, S. Matwin, & B. Plante, "Explanation-based learning with incomplete theories: A three-step approach" In B.W. Porter & R.J. Mooney (eds.), *Proceedings of the Seventh International Conference on Machine Learning*, pages 286--294, Austin, TX, 1990.

L. Hunter, "Classifying for prediction: A multistrategy approach to predicting protein structure," In R.S. Michalski & G. Tecuci (eds.), *Proceedings of the First International Workshop on Multistrategy Learning*, pages 394--402, Harpers Ferry, WV, 1991.

L. Hunter, "Planning to Learn," *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, pages 26--34, Boston, MA, 1990.

A. Lesgold, S. Lajoie, M. Bunzo, & G. Eggan, "SHERLOCK: A Coached Practice Environment for an Electronics Troubleshooting Job," in J. Larkin, R. Chabay, & C. Scheftic (Eds.), *Computer Assisted Instruction and Intelligent Tutoring Systems: Establishing Communication and Collaboration*, Hillsdale, NJ: Lawrence Erlbaum Associates, 1988.

M.J. Pazzani, "Learning to predict and explain: An integration of similarity-based, theory-driven and explanation-based learning," *The Journal of the Learning Sciences*, 1(2):153--199, 1991.

S. Narayanan, A. Ram, S. M. Cohen, C. M. Mitchell, & T. Govindaraj, "Knowledge-based diagnostic problem solving and learning in the test area of electronics assembly manufacturing," *Proceedings of SPIE's Applications of Artificial Intelligence X* conference, Orlando, Fl, 1992.

A. Ram, "A theory of questions and question asking," *The Journal of the Learning Sciences*, 1(3 & 4), pages 273--318, 1991.

A. Ram & M. Cox, "Introspective reasoning using meta-explanations for multistrategy learning", To appear in R.S. Michalski & G. Tecuci (eds.), *Machine Learning IV: A Multistrategy Approach*, Morgan Kaufman Publishers, San Mateo, CA., 1992.

A. Ram & L. Hunter, "The use of explicit goals for knowledge to guide inference and learning," *Applied Intelligence*, to appear. Also available as Technical

Report *GIT-CC-92/04*, College of Computing, Georgia Institute of Technology, Atlanta, GA, 1992.

J. J. Richardson, R. A. Keller, R. A. Maxion, & P. G. Polson, "Artificial Intelligence in Maintenance: Synthesis of Technical Issues," Technical Report *F33615-82-C-0013*, Air Force Human Resources Laboratory, October, 1985.

J. W. Shavlik & T. G. Dietterich (Eds.), *Readings in Machine Learning*, San Manteo, CA: Morgan Kaufmann, 1990.

J.W. Shavlik & G.G. Towell. "An approach to combining explanation-based and neural learning algorithms", *Connection Science*, 1(3), 1989.