

Diagnosis of Communicating Systems: Dealing with Incompleteness and Uncertainty*

Marc Riese

Swiss Federal Institute of Technology, EPFL-LIT
CH-1015, Lausanne, Switzerland, riese@litsun.epfl.ch

Abstract

Systems with state, such as simple sequential circuits, can sometimes be modelled at a high-level of abstraction or encapsulated as a single component in order to diagnose their behaviour. However, for distributed systems and complex state machines, such encapsulation may not be appropriate and high-level abstraction may offer insufficient detail for diagnosis. Here state evolution must be dealt with more directly. This paper presents a novel algorithm, called HMDP (Heuristic Model-based Diagnosis of communication Protocols), for device-independent interpretation of observations and diagnosis of communicating agents¹ in distributed discrete systems, such as communication networks. A fundamental motivation of the HMDP is to deal with problems of incompleteness and uncertainty of observations, and tractability problems incurred by detailed diagnosis. These problems are addressed by: heuristic and approximate matching and interpretation techniques; an incremental propagation technique; use of temporal information; and reiteration on subsets of conflicts.

1 Introduction

Problem: One of Hamscher's modelling principles states, "A sequential circuit should be encapsulated into a single component to enable the description of its behavior in a temporally coarse way" [1991]. This principle addresses tractability and is especially appropriate for cases where encapsulated components: form a single replaceable unit; have a relatively simple external interface; or have input/output behaviour similar to a combinatorial device when viewed in a temporally coarse way (as in Hamscher's example). However, for distributed systems and complex state machines, such encapsulation may not be appropriate and high-level abstraction may offer insufficient detail for diagnosis. The approach proposed here deals with the internal behaviour of complex

agents allowing more detailed analysis, but must also address the tractability problems incurred. The approach is intended for distributed discrete systems where models of communication, or protocols, between agents are at least partially available, such as for developing (debugging, conformance testing) or managing (troubleshooting) tele- and data-communication networks.

Diagnosis of systems with state has been the subject of much recent investigation, but most attention has been devoted to simple sequential circuits or so-called physical systems. Although the *flexibility*, *reliability* and *device-independence* of model-based approaches are very desirable for the diagnosis of distributed discrete systems, this problem has been addressed by few researchers. Expanding size and power of communication networks and rising user dependence make network diagnosis an increasingly difficult yet important problem. Error manifestations may be distributed physically and temporally. For example, an agent may incorrectly interpret a request directed to itself and erroneously broadcast the request to its neighbours, possibly resulting in a fatal chain reaction. Protocol functioning can not be guaranteed to be error-free because design verification and implementation validation techniques are still limited, and exhaustive testing of non-trivial implementations is practically impossible. A protocol may pass conformance tests and yet only state "Reason not specified" in all error messages [CCI, 1989, 13.5.3, 13.12.3]. Furthermore, a specified reason may not be correct. New, previously unexperienced problems — for which model-based diagnosis is particularly useful — arise as components are added to the network, are reconfigured, or are revised, such as with new software releases.

Observation of such systems is often *incomplete* and *uncertain*. Incompleteness arises because: state may evolve too quickly for detailed real-time diagnosis; distributed agents may generate too many events for complete storage; internal probing may be inaccessible; and testing may be slow, difficult, or impossible. Uncertainty arises because: observers are distributed and may be erroneous; observed event types may be generated by several agents in various states; observed (external) and actual (internal) event sequences may differ; and universally synchronous timing cannot be assumed.

Approach: This paper attempts to address the above problems by proposing a model-based algorithm for interpreting observations and diagnosis. A set of observations OBS is collected from logs of sensors distributed amongst the agents of the system to be diagnosed. For

*The Swiss PTT provided funding as part of ESPRIT II project #2409 EQUATOR, Environment for Qualitative Temporal Reasoning.

¹Agents are protocol executing entities that, together with links, form a distributed system. This is more general than the network management meaning of "agents".

example, an agent could be a protocol executing entity of a network and an observation could be a message time-stamped by an external sensor, as in Figure 1. The model, called the system description *SD*, includes

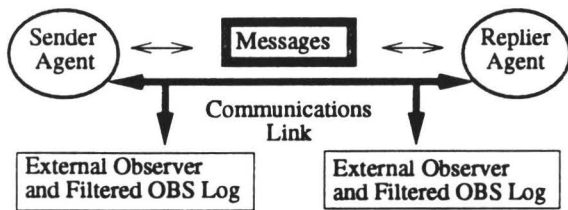


Figure 1: Two Agents and Their Observers

(possibly extended) finite-state machine (FSM) rules or constraints modelling agent communication behaviour. Initially, each observation is propagated forward in *SD* to determine the set of externally observable events, called *propagations*, that could have followed it in the next state of the model. Propagations of one observation are matched to other, temporally relevant observations and their propagations having similar message types and overlapping temporal intervals to find a possible explanation i.e. that one observation predicts the other. Matched observation pairs are used to determine the best possible interpretations of the sequence of transitions that explain OBS collectively. Weaknesses of explanation in the interpretations are used to suggest sets of constraints in *SD* that could be modified to form variants *SD_i*. The algorithm then iterates on these variants to find the best interpretations of OBS. Thus, iterative modification of the base model is performed in order to search for better matchings. The sets of constraints modified in order to form the models that generate the best interpretations are used to propose plausible diagnoses.

Contributions: The main contribution of this paper is a novel approach to interpreting and diagnosing distributed discrete agents of non-trivial complexity, in which observations may be incomplete and uncertain. Standard approaches, such as [de Kleer and Williams, 1987], process all conflicts together in a refinement loop and make strong assumptions about observation availability and certainty when resolving conflicts. The HMDP algorithm is derived from this, but addresses the problem domain as follows: internal access for observation is assumed to be limited; cost of testing or probing is taken into account; incremental and temporal propagation reduce search; intermediate propagations can fill substantial observation gaps; evaluation of matches and interpretations is based on probabilities and heuristics; the algorithm focuses on finding good interpretations of *OBS* as a whole in order to resolve conflicts and to find plausible candidates; reiteration allows focusing on a small number of conflicts at any time; and optional operator interaction improves focussing. As a second contribution, the paper proposes a flexible and device-independent technique applicable to the important task of diagnosing communication protocols.

The following section gives an example problem. The algorithm is then described in detail and applied to the example. Next, issues and related work are discussed. The conclusion briefly discusses an implementation of a TP4 model [CCI, 1989], current status, and

ideas for future investigation. This presentation is concise due to space limitations; a more detailed report is available from the author.

2 ABP Example

The Alternating Bit Protocol (ABP) is one of the simplest communication protocols [Bartlett *et al.*, 1969]. It describes how a *Sender* and a *Replier* agent interact to transfer data messages from the former to the latter. The *Sender* appends a 1-bit to send a data message, awaits an acknowledgement having the same bit from the *Replier*, sends the next message with a 0-bit, and so on. After transmitting a message with a 1-bit, the *Sender* must either: (1) receive an acknowledgement with a 1-bit; (2) time-out and resend the message with a 1-bit; or (3) receive and ignore redundant acknowledgements having a 0-bit. The *Replier* functions in a complementary manner. A simplification of the ABP is illustrated in Figure 2, where vertex labels correspond to state names, edges to transitions, and edge labels to input (1,0), output ($\bar{1}$,0), or internal events (time-outs). Initial states are in bold. In order to represent timing

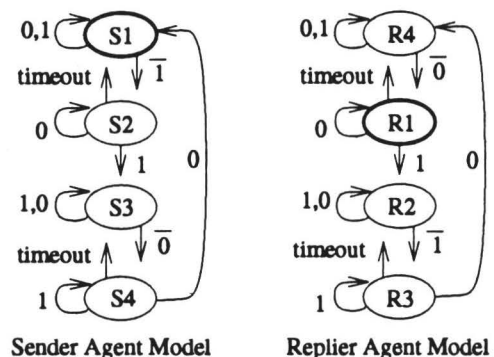


Figure 2: Alternating Bit Protocol State Machines

behaviour, some temporal specifications must be added. For the example, assume that each time-out is set to eight time units². The HMDP algorithm is applied below to the following example problem: given this model and a set of observations, find a specific implementation error, namely that the time-out transition between states *S3* and *S4*, denoted *S4* → *S3*, is set to one time unit instead of eight.

3 The HMDP Algorithm

Brief Overview: In practice, diagnosis is triggered by an operator or an automated monitor, optionally offering rough approximations of the relevant temporal interval and the physical locality of one or more error manifestations. An initial set *OBS* is collected from a log and possibly by probing the *Implementation Under Test* (IUT, the system diagnosed). The trigger and *OBS* are used to determine which entities in the IUT are included in *COMPS*, the set of components being diagnosed (agents

²More precisely, assume that: timing is synchronous; *Sender* and *Replier* timing parameters are equal; external observers time-stamp messages; total transmission and propagation time between the agents is one time unit; and processing time for any state is between zero (plus epsilon) and five time units.

and their components). Each observation O_i is initially propagated forward through SD . This generates a subset of the propagations that could have immediately followed O_i according to the model. Each propagation p_{O_i} is compared to temporally relevant observations and their propagations. If p_{O_i} and some observation O_j have similar message types and overlapping temporal intervals, then the pair (O_i, O_j) forms a *match*. This match indicates that the state transitions used for the propagation give a possible explanation that O_i predicts O_j . A likeliness value is evaluated for each match. Incremental deepening of propagations is performed selectively. Once all matches have been determined, they can be represented as a directed graph G_{SD} whose vertices are the members of OBS and whose edges are matches. *Interpretations* of OBS with respect to any agent can be evaluated by finding acyclic paths in G_{SD} .

If the IUT is faulty, the best interpretations of G_{SD} in the first iteration offer imperfect explanations. The algorithm must determine which constraints in SD may be associated with the weak areas of interpretations of G_{SD} , possibly due to a specific conflict between the model and the implementation. Sets of constraints are modified in order to generate variants of SD . The algorithm then loops, propagating OBS through one variant SD_i in each iteration to find better interpretations and hence better explanations. Sets of modified constraints corresponding to models having the best interpretations are used to propose plausible diagnoses.

Propagation: The goal of propagation is to find externally observable events that could have been received or transmitted by a given state machine before or after a given observation. This is done by searching transition paths extending to and from a starting transition corresponding to the given observation until an observable is found at each path searched. Observables immediately preceding or following a given observation are its *primary* propagations, having a depth of one. *Secondary* propagations go one level deeper and so on. Tables of propagations of transitions can be calculated in advance, allowing near-linear propagation time during diagnosis. Forward propagation corresponds to predicting observables occurring after the observation, assuming that the starting transition is correct; propagating backward corresponds to abductive hypothesis of observables occurring beforehand. Only forward primary propagations are found initially. Observations that subsequently receive poor matching evaluations have not predicted, or are not predicted by, other observations with sufficient plausibility. Propagation of these observations is deepened by incrementing propagation in the direction of matching weakness (forward or backward). Since observations may be aberrant, incremental deepening is only repeated a small number of times.

If not given, rough transition probabilities and transition times can often be estimated in advance and partially refined by statistical analysis of IUT performance or simulation. Probability estimates are based on heuristics (e.g. time-out are less probable than normal transitions) and statistical analysis. Timing estimates are based on the function of a transition (e.g. send internal message), on heuristics (e.g. normal processing times may be estimated from timing constraints), or on code execution estimates. Propagation requires estimating time intervals extended from the time associated with

the original observation. Time labels must be normalised with respect to the agent being diagnosed.

Consider the propagations in Figure 3, based on observing a *Sender* of the ABP protocol³. Timing is indicated relative to the *Sender*. Observation (S 0 20), a

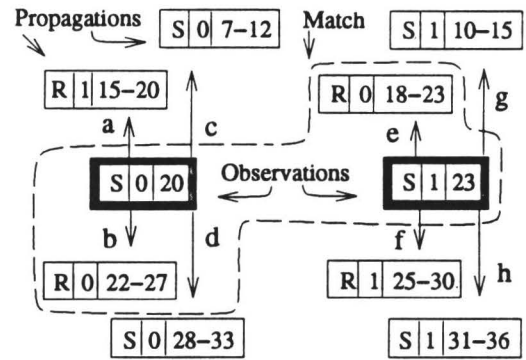


Figure 3: Propagating S:0:20 (left), S:1:23 (right)

Sender output message having a 0-bit observed at time 20, is propagated through the transition constraints of SD to derive primary propagations. Five transitions in Figure 2 correspond to a message with a 0-bit sequence number originating from the *Sender*: $S3 \rightarrow S4$, $R3 \rightarrow R4$, $R1 \rightarrow R1$, $R2 \rightarrow R2$, and $R4 \rightarrow R4$. This is the set of *agent-transition* pairs related to the observation (S 0 20). Consider propagation from transition $S3 \rightarrow S4$. The backward propagation (R 1 15-20), labelled *a* in Figure 3, indicates that, between times 15 and 20 relative to the *Sender*, the *Replier* could have generated a message having a 1-bit in acknowledgement to a previous *Sender* message. Forward propagation (R 0 22-27), labelled *b*, corresponds to a *Replier* acknowledgement of the observation.

For diagnosis of simple circuits, the signal observed at an input terminal can be precisely correlated to the input parameter of a constraint, and the result of propagation can be correlated to observation of output terminals. In the problem domain treated here, observers are external and cannot measure “terminals” of a transition, so the correspondence between observations and transitions must be hypothesised using agent-transition pairs, incremental propagation, and matching.

Matching: The goal of matching is to evaluate if, how, and how well each propagation of an observation O_i can be matched to members of $OBS'_T \subseteq OBS \setminus \{O_i\}$ and their propagations. With each observation is associated an overall interval T indicating the earliest and latest end-points of the intervals of all of its propagations. Before propagations are matched, intervals based on T are compared and only overlapping pairs are matched.

Matches are found by comparing messages and time intervals of propagations and observations. Match evaluation is initially based on the probabilities of the transition(s) of related propagations. Consider an example with observations (S 0 20) and (S 1 23) and their propagations as shown in Figure 3. Here, forward propagation (R 0 22-27), labelled *b*, and backward propagation

³For lack of space, some inessential details on probabilities, timing, and less likely propagations involving redundant acknowledgements are not given here.

(R 0 18-23), labelled e , have the same source agent (the *Replier*), the same sequence number (zero), and overlapping time intervals. (When data corruption is suspected, such as from a sensor or a transmission link, close but imperfect pattern matches can be retained with lower evaluations.) Because interval estimates widen rapidly as propagation depth increases, the evaluation is divided by a value based on the depth of propagation needed to relate one observation to another. Thus, intermediate missing observations can be hypothesised at the cost of lower evaluation.

Interpretation: The generated matches of SD can be represented as the weighted edges of a directed graph G_{SD} whose vertices are the members of OBS . An example G_{SD} is summarised on the left of Figure 4, where propagation (R 0 *) is shown explicitly, rather than as part of several edges. Interpretations of OBS are generated by finding distinct, acyclic paths in G_{SD} using a heuristic traversal algorithm. For example, if each member of OBS predicts exactly one member occurring after it and is predicted by exactly one occurring before it, the interpretation could represent a partial simulation of an agent. Interpretations are evaluated by several factors such as the sum of individual match evaluations, the number of members of OBS included, and the number and lengths of paths. Thus an interpretation can exclude erroneous observations and yet retain a good ranking.

Reiteration: For the first iteration of the main loop, which uses SD , even the best interpretations generated are likely to have imperfect evaluations because SD does not explain observations of the faulty IUT. Information on specific weaknesses of the interpretation of G_{SD} , such as excluded observations or low match evaluations, is used to determine a set C of sets of constraints C_i related to poor explanations. With each constraint $c_{i,j} \in C_i$ is associated a primitive modification operation $m_{i,j}$. Several primitive modification operations are related to each type of constraint (and parameters for extended models). For example, a state-transition constraint can be modified by a suspension operation (or other operations as in [Bouloutas, 1990]), and a parameter can be set to extrema or qualitatively distinct values (ex. buffer empty, buffer not-empty, buffer full). Once the set C has been initialised at the end of the first iteration, each subsequent iteration i uses C_i to determine the model SD_i used for propagation, and may add new elements to C . For example, if C_1 consists of one transition constraint $c_{1,1}$ whose modification primitive is SUSPEND, then $SD_1 = SD \setminus C_1$. High evaluation of an interpretation of G_{SD_1} means that conflicts have been reduced and that SD_i offers a plausible explanation of OBS , so C_i is used to propose a diagnosis.

The creation of set $C = \{C_1, C_2, \dots\}$ is a difficult problem. It could be generated: (1) by brute-force generation of permutations of relevant constraints; (2) by heuristics associating evaluations with constraints; (3) by use of functional models⁴ or fault models [Struss and Dressler, 1989]; or (4) by operator interaction, as visualisation of interpretations is often very informative for humans. The set C may be partially ordered (e.g. by interacting with the operator) according to the interpretation weaknesses addressed by each C_i and the potential

⁴OSI-style service primitives provide an excellent basis for protocol functional models.

Sender Output Bit,S-Time	Replier Output Bit,R-Time	Merged Output Log Origin, Bit,S/R-Time
...	...	...
1 : 12	1 : 16	S : 1 : 12
0 : 17	1 : 28	R : 1 : 16
0 : 18	...	S : 0 : 17
0 : 19	-	S : 0 : 18
0 : 20	-	S : 0 : 19
0 : 21	-	S : 0 : 20
0 : 22	-	S : 0 : 21
1 : 23	-	S : 0 : 22
...	-	S : 1 : 23
-	-	R : 1 : 28
-	-	...

Table 1: ABP Observation Logs Used to Form OBS

value of the corresponding modifications.

Flow Control

Step 1: Trigger. Initialise OBS , $COMPS$.

Step 2: Propagation, matching, interpretation.

Step 3: Costing Additional Observations Any IUT probes and tests requested for completion of an interpretation or for discrimination of candidates are evaluated for: potential cost (IUT response time and availability), potential informative value, and additional computation time.

Step 4: Termination Information has been accumulated about: candidates; potential value of reiteration; and potential cost and value of further observation to improve current interpretations. Termination occurs if candidates acceptable to operator as diagnoses, or if reiteration possibilities exhausted.

Step 5: Observing or Reiterating If no more observations requested or their cost is prohibitive, then reiterate with a new SD_i at Step 2. Otherwise, perform probing or testing, complete interpretations with new observations, and go to Step 4.

Example: Consider an implementation of the ABP as specified in Section 2. Given a model based on this specification and sample logs⁵ of the IUT as in Table 1, an example diagnosis problem is to find a specific implementation error, namely that the time-out transition $S4 \rightarrow S3$ is set to one instead of eight. This means that if the *Replier* uses its maximum normal processing time, the *Sender* will time-out often and resend many messages having a 0-bit.

The diagnosis session may have been triggered (Step 1) by a monitor indicating that an error manifestation (excessive traffic) was detected in a physical region including the two agents and between times 0 and 100. The set OBS is initialised by scanning logs for entries originating from or destined for the given region and occurring within the given time interval. $COMPS$ is determined using the triggering information and explicit (ex. destination) or implicit (ex. transmission path) component information in OBS . Propagation and matching (Step 2) are executed.

⁵Logs are limited to agent output here in order to facilitate comprehension of time labels.

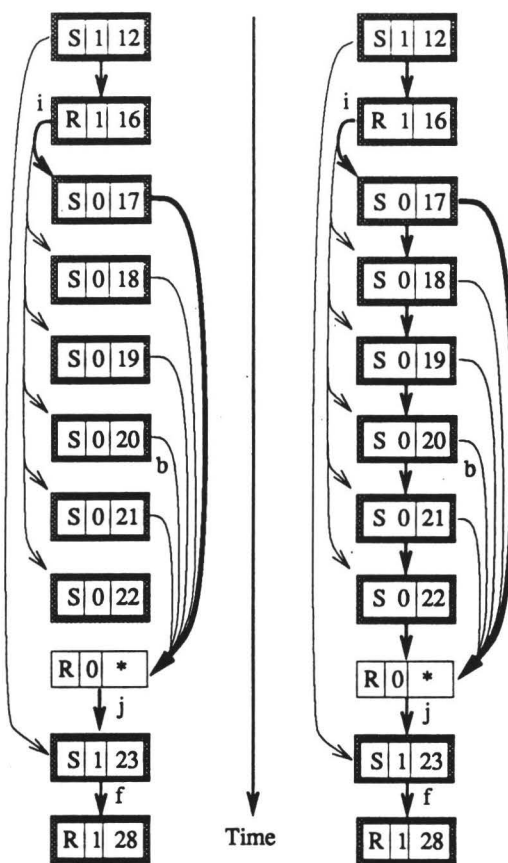


Figure 4: Summarised G_{SD} (left), G_{SDi} (right)

The left half of Figure 4 summarises G_{SD} . The outgoing transition of observation (S 1 23) in this figure, labelled f , corresponds to the transition with the same label leading to propagation (R 1 25-30) in Figure 3. This primary propagation matches to observation (R 1 28) in Figure 4. A forward propagation of observation (R 1 16), labelled i , matches to each of the (S 0 *) observations, but only one of these transitions is possible. Observation (S 0 20) has a transition labelled b in Figures 3 and 4, corresponding to the primary forward propagation (R 0 22-27). This is marked (R 0 *) in Figure 4 to indicate that other (S 0 *) observations generate similar propagations with shifted intervals. After initial propagation, none of the (S 0 *) observations have forward matches, so their forward propagations are incremented to depth two. This allows the secondary forward propagations corresponding to the transition labelled j to match to observation (S 1 23).

Interpretations are evaluated and set C is initialised. Constraints related to the generation of (S 0 *) messages are modified, because no interpretation path of G_{SD} includes more than one of the (S 0 *). An example interpretation is indicated as a transition path in bold on the left half of Figure 4. No additional observations are required (Step 3), and of course no candidates are proposed in the first iteration (Step 4), so the algorithm reiterates (Step 5) on the variant models.

Consider SD_i , where the time-out value of transition $S4 \rightarrow S3$ is set to one i.e. the absolute value of the time constraint (8 units after) is changed, but the rel-

ative time constraint (after) continues to hold. G_{SDi} is summarised on the right half of Figure 4. Here, the (S 0 *) observations match by use of transition $S4 \rightarrow S3$. One interpretation of SD_i (in bold) scores a high evaluation because it includes all observations in one transition sequence. Thus the model SD_i offers a better explanation of OBS than SD . Interpretation has provided useful information for diagnosis: that the problem is related to 0-bit messages generated by *Sender*, and that plausible explanations are generated when the time constraint related to $S4 \rightarrow S3$ is modified. Note that a missing observation (R 0 22) has been predicted and included.

One interpretation may include multiple transition sequence paths, where each path explains a distinct subset of OBS ordered over time. Gaps between any two paths have weak explanations, and provide information for potential relaxations. One heuristic for ranking different interpretations is to favour those for which the path starting with the earliest members of OBS is long. This heuristic is similar to the following: "The longer a hypothesis survives changing observations, the stronger the evidence supporting that hypothesis. This age-ranking also focuses attention on hypotheses that account for the earliest manifestations of a fault, before other manifestations and corresponding hypotheses appear." [Dvorak and Kuipers, 1991].

4 Issues and Related Work

Simulation and Tractability Because of the problem characteristics described, naïve simulation would very quickly become intractable. The HMDP algorithm attempts to address tractability by: limiting OBS and agent-transitions pairs used; incrementing propagation; using of temporal information; constraining traversal of G_{SD} ; and using multiple iterations to focus on small numbers of conflicts.

OBS Density When OBS is dense, i.e. when it represents a high proportion of observable events, observations tend to require only one forward propagation. Incremental propagation is comparable to opportunistic reasoning techniques that exploit "islands" of information. This is generally more efficient under incomplete and uncertain conditions than propagation to a fixed depth in a forward direction alone, which creates a larger number of irrelevant propagations. If observations are too scarce, then evaluations are low, reiteration is poorly constrained, and likely diagnoses may not be found.

Distributed Diagnosis The approach presented is capable of finding candidates involving constraints of any of the communicating agents in *COMPS*. However, it must perform matching and interpretation of a given model one agent at a time in order to normalise time-stamps and to correlate messages. A more efficient approach would be desirable for practical global diagnosis.

Communications: An approach to diagnosis of FSMs for troubleshooting based on strongly-sequenced structure estimation is presented in [Bouloutas, 1990]. *A priori* knowledge of possible faults is required, and calculation is substantial even for simple machines (pp85). Ghedamsi, Bochmann and Dssouli propose a model-based approach to diagnosis of FSMs for conformance

testing in [1993]. This approach offers more certain diagnoses than the HMDP algorithm at the cost of dependence on systematic testing. It diagnoses faults involving incorrect outputs or state changes. Zinky and Etkin implemented a system that uses model-based techniques to diagnose network performance using abstract behaviour models of traffic and queues [Zinky and Etkin, 1992].

Model-Based Reasoning The Mimic system for process diagnosis [Dvorak and Kuipers, 1991], the DATMI system for measurement interpretation [DeCoste, 1990], and the HMDP algorithm are related because all three involve reasoning about multiple possible states with incomplete or uncertain observations. DATMI and Mimic were designed for continuous physical systems where state does not usually evolve as rapidly as communicating systems. Mimic attempts to propagate observations forward in one model for each currently plausible fault hypothesis during system operation. Tractability problems would arise for such an approach in the described problem domain. The notion of "iterative search based diagnosis" using information about the degree of mismatch between observations and predictions was first proposed by in [Leitch and Shen, 1991]. This and the HMDP approach share "the important advantage of reducing the sensitivity to modelling errors and to approximate fault models." Both approaches use temporal durations to reduce search. However, the HMDP algorithm is applied to distributed discrete systems, concentrates on incomplete and uncertain conditions, and does not assume that state can be tracked synchronously.

5 Conclusion

Status: Propagation, matching, and interpretation procedures have been implemented and tested on the ABP prototype model. A prototype model of the OSI Transport Protocol Class 4 (TP4 [CCI, 1989]), considered to be one of the largest and most complex communication protocols, has been implemented to determine the scalability and applicability of the HMDP algorithm and for troubleshooting scenarios based on a real implementation of the entire OSI stack. Automated reiteration and practical tractability with the TP4 have not yet been tested at the time of writing.

For implementation independence, the basis for our TP4 model is the standard specification. Such specifications allow multiple correct implementations, each of which must be allowed by the diagnosis program. This issue, which is related to behaviour modes [de Kleer, 1989], increased modelling effort considerably. For example, an event may cause multiple optional outputs in an unspecified order. For efficiency, such output sequences must be modelled abstractly rather than extensionally.

Current and Future Work: Practical considerations and preliminary analysis are reported in [Riese, 1993b]. This includes a reformulation of the problem addressed by the HMDP algorithm as a constraint satisfaction problem. We are currently working on using this formulation to extend the algorithm in order to take advantage of constraints related to variables, timers, guard conditions, and side-effect procedures. Such rich information is ignored by most FSM-based approaches, but is available in communication models, and is very useful for reducing search and for corroboration. We plan to experiment on the hypothesis that discrepancy metrics

between correct and incorrect interpretations (similar to [Leitch and Shen, 1991]) can be used to guide model modification. Other potential areas for research include the use of reason maintenance, fault models, and hierarchical abstraction to improve performance, and the development of distributed diagnosis and multiple model techniques (including physical and queueing models) to increase general practicality. More extensive results and analysis will be available in [Riese, 1993a].

Acknowledgements Thanks to Sailesh Chutani, Boi Faltings, Roy Leitch, Beat Liver, Henri Nussbaumer, Pearl Pu, Prasad Raja, Peter Struss, and the referees for helpful discussions or suggestions on an earlier draft.

References

- [Bartlett et al., 1969] K. Bartlett, R. Scantlebury, and W. Wilkinson. A note on reliable full-duplex transmission over half-duplex links. *CACM*, 12(5), 1969.
- [Bouloutas, 1990] A. Bouloutas. *Modeling Fault Management in Communication Networks*. PhD thesis, Columbia University, 1990.
- [CCI, 1989] CCITT. *OSI Protocol Specifications, Conformance Testing. X.220-X.290*, 1989.
- [de Kleer and Williams, 1987] J. de Kleer and B. Williams. Diagnosing multiple faults. *A.I.*, 32, 1987.
- [de Kleer, 1989] J. de Kleer. Diagnosis with Behavioral Modes. In *IJCAI*, 1989.
- [DeCoste, 1990] D. DeCoste. Dynamic across-time measurement interpretation. In *AAAI*, 1990.
- [Dvorak and Kuipers, 1991] D. Dvorak and B. Kuipers. Process monitoring and diagnosis. *IEEE Expert*, June 1991.
- [Ghedamsi et al., 1993] A. Ghedamsi, G. Bochmann, and R. Dssouli. Diagnosing multiple faults in finite state machines. Technical Report 859, Université de Montréal, Canada, Jan. 1993. Accepted for IFIP-93.
- [Hamscher, 1991] W. Hamscher. Modeling digital circuits for troubleshooting. *A.I.*, 51, 1991.
- [Leitch and Shen, 1991] R. Leitch and Q. Shen. Finding faults with model based diagnosis. In *Second Intl. Workshop on Principles of Diagnosis*, Oct. 1991.
- [Riese, 1993a] M. Riese. *Model-Based Diagnosis of Communication Protocols*. PhD thesis, Swiss Federal Institute of Technology, Lausanne (EPFL-LIT), 1993. In preparation.
- [Riese, 1993b] M. Riese. Réalisation d'un système de diagnostic basé sur modèle pour les protocoles de communication. Technical Report EPFL-DI 93/24, March 1993. Submitted to Colloque francophone sur l'ingénierie de protocole, Montréal, Canada, 1993.
- [Struss and Dressler, 1989] P. Struss and O. Dressler. 'Physical negation' - integrating fault models into the general diagnostic engine. In *IJCAI*, 1989.
- [Zinky and Etkin, 1992] J. Zinky and J. Etkin. Troubleshooting throughput bottlenecks using executable models. *Computer Networks ISDN Systems*, 24, 1992.