Diagnosis Framework for MDS
Proposal for 632-07

Task Leaders

Daniel Dvorak, JPL, dldvorak@pop.jpl.nasa.gov, 818-393-1986, PI.
Dennis DeCoste, JPL, decoste@aig.jpl.nasa.gov, 818-393-5366, co-PI.
Pandu Nayak, ARC, nayak@ptolemy.arc.nasa.gov, 650-604-4756, co-PI.
James Kurien, ARC, kurien@ptolemy.arc.nasa.gov, 650-604-4745, co-PI.

Product Description

This project will demonstrate and deliver a software framework for onboard fault diagnosis that plugs into the JPL Mission Data System (MDS) state determination framework. This framework will define software classes for modeling device behavior and will include a model-based domain-independent diagnostic algorithm that produces probability-ranked hypotheses. First-year work will deliver a basic capability to MDS by adapting suitable technology from the Livingstone state identification system [Williams & Nayak 1996], as demonstrated during DS1's Remote Agent experiment. Second and third year work will explore extensions that allow diagnostic reasoning about uncertain evidence. This is a ***new, push*** *task, with about 25% pull from MDS Program.*

Benefits

The proposed technology will make it easier for missions to achieve robust diagnostic performance. Left to their own devices and budgets, most missions will opt for a seductively simpler approach to diagnosis—such as hard-coded logic or rule-based heuristics—where diagnostic knowledge is not clearly expressed and structured, has potentially serious errors of omission, and is brittle with respect to hardware design changes. Such approaches co-mingle knowledge about device behavior with diagnostic techniques, making it hard for missions to make informed tradeoffs between the fidelity of their models and the thoroughness of their diagnostic reasoning. In contrast, the proposed model-based approach provides a strong organizing principle for device knowledge that is isomorphic to the hardware design, providing models that are easily inspectable, individually testable, readily modifiable, and cleanly separated from the diagnostic reasoning algorithms. The benefits of such better organization not only pays dividends throughout the life of a mission but also facilitates reuse. This proposal serves a basic goal of MDS and DSMS (Deep Space Mission System): to reduce mission costs and risks by providing high-quality reusable technology and services.

As with many advances in technology, it is difficult to make a credible quantitative claim for cost reduction or reliability improvement without a controlled experiment. However, we can make an argument based on many years of research by many people that highlights the value of separating knowledge (in the form of device behavior models) from reasoning (in the form of probability-ordered diagnostic search). Such separation has been an important theme in AI research in diverse areas such as diagnosis, planning, design, and instruction

Technical Approach

This task has two distinct phases, each with its own challenges. The first phase, largely in the first 13 months, will focus on producing a basic diagnostic framework that first customers can begin to use, to be delivered as part of a November 2000 MDS delivery. This phase will begin with well-known technology for conflict-driven model-based diagnosis—the Livingstone state identification system—and will adapt suitable parts of it (possibly most of it) to the MDS state determination framework. This work will be guided by MDS requirements and focused on a spacecraft domain that is rich in potentially conflicting evidence from multiple sources, specifically GNC (guidance, navigation and control). All designs and prototypes will be tested on the "MDS Mission Spacecraft", an MDS-defined generic spacecraft design, including high-level and sensor-level simulators, that is highly relevant to the specific spacecraft of future MDS Space Science mission customers (e.g. Europa Orbiter, Pluto-Kuiper Express, and Solar Probe).

There are differences in state determination between Livingstone and MDS that will present technical challenges in the first phase that will shape research in the second phase. For example, MDS uses both discrete and continuous behavior models, MDS takes evidential uncertainty into account when deciding if a conflict exists, and MDS must synchronize trailing diagnostic conclusions with real-time estimation. The phase-one work will focus on pragmatic approaches that make a more principled diagnostic capability available to MDS customers.

The second phase will investigate the ill-defined and/or unsolved issues by extending the framework, by September 2002, to use a unified representation for modeling uncertainty and performing probabilistic inferences, possibly based on Bayesian networks. This second phase will be more exploratory than the first phase. In particular, we propose using a framework based on probabilistic graphical models (e.g. Bayesian networks [Pearl 1988]). Using an underlying graphical model formalism will allow a principled and unified representation for modeling uncertainty and performing probabilistic inferences. A key objective is to use an underlying graphical model formalism while not imposing undue constraints on the surface representations that customers use to develop their specific device models. Thus, we will explore compilation methods that automatically translate various model representations—such as qualitative symbolic constraints, engineering models (e.g. Kalman filters), and rules—into underlying graphical model representations.

This proposed work consists of several planned activities, driven by requirements and feedback from the MDS Design Team:

1) *Design the API for device behavior models in MDS.*
2) *Define a sample state determination problem using a subset of the MDS spacecraft and implement the device models.*
3) *Design and implement estimators that decide when two or more pieces of evidence are in conflict.*
4) *Implement a simple conflict-driven diagnostic engine that tests single-fault hypotheses in probability order (available in Livingstone).*
5) *Integrate all the pieces into a working system.*
6) *Demonstrate the system as an intermediate milestone.*
7) *Evaluate performance, memory usage, and scalability for larger, more typical problems.*
8) *Review design and evaluation results with MDS Design Team.*
9) *Implement a more fully featured diagnostic engine that supports probability-ordered multiple-fault diagnosis (available in Livingstone).*
10) *Test the diagnostic engine on synthetic problems to check its accuracy and robustness.*
11) *Document the framework and how to use it. (Note: The decision of whether to use Java or C++ will depend on the results of MDS's Java evaluation in August 1999).*

In addition to developing a trailing diagnosis technology per se, this effort will also study methods for state estimation that best take advantage and support that capability. This will involve joint estimation of *multiple states, as well as hierarchical estimation of decomposed partial states (including multiple time scales).*

Partnering

JPL and NASA Ames will work jointly on this proposal. JPL will be responsible for the bulk of the effort, particularly in working with MDS to understand both the near-term and longer-term requirements for a diagnosis framework, and in delivering products that meet those requirements. NASA Ames will supply technical expertise on the “Mode Identification” technology demonstrated in the Remote Agent Experiment of Deep Space 1 [Williams & Nayak1996] and consulting on adaptation issues. Given the differences in the state determination frameworks between Remote Agent and MDS, it will require some study to decide how much existing technology can be used directly and how much will be adapted.

Status and Milestones

This is a new task. However, the PI’s have extensive experience with the underlying technologies (diagnosis, state estimation, Bayesian networks) and have conducted some preliminary discussions with MDS Design Team during FY99.

Status for FY 1999:

Currently preparing and reviewing API for models with MDS Execution & Planning team (the team that defines the state determination architecture).

FY 2000 Milestones:

- *Demonstrate initial working system on sample problem (GNC of MDS spacecraft).*
- *Review performance and scalability results with MDS*
- *Basic diagnosis framework implemented, testing and documentation begun.*

FY 2001 Milestones:

- *Begin testing of fully featured diagnostic engine.*
- *Present preliminary documentation and training; get feedback for improvement.*

FY 2002 Milestones:

- *Deliver final tested product, documentation, and training.*

Customer Relevance

MDS exists to provide a flight/ground/test software foundation for many future missions to use, so the software deliverables from this task will become available to an indeterminate number of customers. The most likely first user of this diagnostic technology will be one of the Outer Planets/Solar Probe (OP/SP) missions, probably Pluto-Kuiper Express, scheduled for a 2004 launch. See letter of support from Anne Elson, OP/SP Project Software Engineer, JPL. Some of the Mars missions may also adopt the X2000 hardware and MDS software, so other customers are expected. This diagnosis framework may also be a candidate for future rovers, subject to rover processor and memory limitations.

MDS itself is also a customer since this task contributes directly to MDS’s state determination framework, as described at the 5/20/99 Peer Review for MDS Execution & Planning. See letter of support from Robert Rasmussen, MDS Chief Architect, JPL. Two major architectural themes in MDS are the separation of state determination from state control and the explicit use of models to express mission-specific and spacecraft-specific knowledge. The proposed framework supports both themes.

Qualifications

- Dr. Daniel Dvorak is a member of the Information and Computing Technologies Research Section at JPL and is the technical lead for state determination in MDS, with applied research interests in fault monitoring and diagnosis, qualitative reasoning, and software architecture.
- Dr. Dennis DeCoste is Senior Member of Technical Staff / Technical Group Leader in the Machine Learning Systems Group at JPL. He has been technical lead of several projects in fault detection and time series data mining at JPL for the last five years. He has served on the AAAI program committee and as reviewer for PAMI, AAAI, and IJCAI. He has published several papers on monitoring and data-mining, including AAAI, AI Journal, AI Magazine, and KDD.
- Dr. Pandurang Nayak is a Senior Computer Scientist and area lead of the Autonomy and Robotics Area at NASA Ames Research Center. He was deputy Project Element Manager of Remote Agent during the Remote Agent Experiment and was task leader for the Livingstone model-based diagnosis and control element. His interests include model-based autonomous systems and abstractions and approximations in reasoning, diagnosis and recovery.
- James Kurien is a computer scientist in the Computational Sciences Division of NASA Ames and a doctoral candidate at Brown University, focusing on improving model-based diagnosis and control under uncertainty. He has experience providing model-based diagnosis and control for the DS1 Remote Agent experiment, the JPL Interferometry testbed, and an in-situ propellant production simulation at KSC, among other applications.

Technical References

[Williams & Nayak 1996] B. C. Williams and P. P. Nayak. "Livingstone: A Model-based Approach to Reactive Self-Configuring Systems." In Proceedings of AAAI-96.

[Pearl 1988] J. Pearl. "Probabilistic Reasoning in Intelligent Systems". Morgan Kaufmann, San Mateo, CA, 1988.

Z. Manna and A. Pnueli (1992). The temporal logic of reactive and concurrent systems: Specification. Springer-Verlag.

W. S. Lovejoy (1991). A survey of algorithmic methods for partially observable Markov decision processes. Annals of Operations Research, 28.

JET PROPULSION LABORATORY **INTEROFFICE MEMORANDUM**

June 24, 1999

To: UPN 632-07 Review Board

Re: Diagnosis Framework for MDS

From: Anne Elson
Outer Planets/Solar Probe Project Software Engineer, Outer Planets/Solar Probe Project

Onboard fault diagnosis has typically been handled in very domain-specific ways, often in an *ad hoc* way that makes it difficult to see what's covered, and requires extensive testing to achieve confidence in fault detection and diagnosis.

I endorse the proposal by DeCoste, Dvorak, Kurien, and Nayak to build a diagnostic framework for MDS because it will benefit mission customers in two important ways. First, the framework will provide a way for us to organize and express knowledge about device behavior in inspectable models. That alone will be an improvement over the current situation where such knowledge often gets distributed—and effectively hidden—throughout the flight code, making it hard to validate and hard to update. Second, this framework will generate probability-ranked fault hypotheses for us, considerably simplifying the amount of diagnostic code that we would otherwise have to develop. Speaking as a customer of MDS, this is a kind of capability that will be quite valuable.

Anne Elson

JET PROPULSION LABORATORY **INTEROFFICE MEMORANDUM**

June 24, 1999

To: UPN 632-07 Review Board

Re: Diagnosis Framework for MDS

From: Robert D. Rasmussen
Chief Architect, Mission Data System

An essential part of the Mission Data System (MDS) is a set of software frameworks for expressing mission-specific state knowledge in a well-structured form that can then be processed with sound techniques. Part of this should be a good framework for fault diagnosis to help customers achieve better coverage and more accurate diagnosis of faults.

I endorse the proposal by DeCoste, Dvorak, Kurien, and Nayak to study, design, and build a diagnostic framework for MDS that extends and improves upon the initial design. Knowing the work of Pandu Nayak and James Kurien on the DS-1 Remote Agent Experiment, I'm confident that the framework will employ good modeling and sound inference procedures. Also, since Dan Dvorak has been an active member of the MDS design team from the beginning, I'm confident that the resulting framework will fit well within the MDS architecture.

Robert D. Rasmussen