Model-Based Programming of Intelligent Embedded Systems and a Robotic Messenger to Mercury

Brian C. Williams, Seung Chung, Mitch Ingham & John Van Eepoel

Artificial Intelligence Laboratory
Massachusetts Institue Of Technology
Cambridge, Massachusetts 02139

http://www.ai.mit.edu

The Problem: Can we develop languages for robots and embedded systems that adapt quickly to novel events, by reasoning on the fly from models of themselves and their environment? How do we make deductive algorithms fast enough to to plan, schedule, diagnose, reconfigure and plan contingencies in real-time? How do we seamlessly integrate these powerful deductive capabilities into traditional languages?

Motivation: Current robotic explorers are demonstrating incredible successes (e.g., the Mars Sojourner rover) and incredible failures (e.g., Mars Polar Lander). The number of embedded systems that are brittle, or have demonstrated catastrophic failure is quickly mounting. For example, Mars Polar Lander was programmed to believe that it had leapt from 50 Meters above Mars down to the surface in milliseconds, causing it to turn off its engine prematurely and plunge to the ground. Other spacecraft have been lost by asking them to point away from Earth without telling them how to restore communication. Similar but less dramatic failures occur within everyday embedded systems. Many of these failures could be avoided if the programming language interpreters or compilers had a little common sense.

Previous Work: Our project builds upon an extensive body of work in reactive and constraint programming languages, automated deduction, common sense modeling, diagnosis and planning. To extend traditional languages with deductive abilities we look to work on the Esterel Language[Berry,Halbwachs] and Concurrent Constraint languages [Saraswat, Gupta]. To achieve real-time inference we look to work on propositional satisfiability [Davis, Putnam, Logeman, Loveland] incremental truth maintenance[Doyle, McAllester, deKleer, Nayak, Williams] and phase transition phenomena in constraint problems [Selman, Cheeseman, Huberman]. For work on common sense modeling we look to research in qualitative physics [de Kleer, Forbus, Kuipers, Williams], and work on representing time [Boddy, Dechter, Kautz, Villain]. For work on diagnosis and monitoring we look to research in model-based diagnosis and belief state update using Hidden Markov Models [Davis, Shrobe, Williams, de Kleer, Nayak, Kaelbling]. For work on fast planning we look to work in classical planning [Fikes, Weld], reactive planning [Williams, Jensen & Veloso] and related work in symbolic verification [McMillian, Bryant].

Approach: A new generation of sensor rich, massively distributed systems is emerging, including building energy systems, deep space probes and sensor webs that monitor the earth ecosystem. These robotic webs have the richness that comes from interacting with physical environments, together with the complexity of networked software systems. They must be efficient, capable and long lived, that is, able to survive decades of autonomous operation within unforgiving environments.

To achieve this level of performance software regulatory and immune systems must be developed that robustly coordinate sensor and actuator activities internal to individual robotic systems, and that coordinate activities between robotic system participating in a robotic web. Developing the regulatory and immune systems of these robotic systems offers an overwhelming programming challenge. Traditionally programmers must reason through system wide interactions along lengthy paths between the sensors, control processors and control actuators. The resulting code typically lacks modularity, is fraught with error and makes severe simplifying assumptions.

Model-based programming meets this challenge through two ideas. First, we note that programmers generate the desired function based on their common sense knowledge of how the software and hardware modules behave. The idea of model-based programming is to exploit this modularity by having engineers program reactive systems by simply articulating and plugging together these common sense models. The second challenge is the infeasibility of synthesizing a set of codes at compile time that envision all likely failure situations and responses. Our solution is to develop real time systems, called model-based executives, that respond to novel situations on the order of hundreds of milliseconds, while performing extensive deduction, diagnosis and planning within their reactive control loop.

In this research we formulate a model-based executive as a deductive form of an optimal, model-based controller, in which models are specified through a combination of hierarchical automata, probabilistic transition systems and propositional logic. This framework allows us to unify a diverse set of research results from model-based reasoning, planning, search, real-time propositional inference, Markov decision processes and the theory of reactive languages. Reactivity is achieved using a high performance deductive kernel, called OPSAT, that solves combinatorial optimization problems with constraints encoded in propositional logic.

A first generation executive, called Livingstone, was demonstrated on NASA's first intelligent autonomous space probe, called Deep Space One, shortly before its asteroid encounter in May 1999. We are currently developing a much more capable executive, called MARS, that is able to reason about and coordinate the behavior of complex components, whose behaviors are modeled by programs. It is our hope that MARS will soon be charting its way aboard the Messenger space probe to Mercury.

Difficulty: In the 80's it was believed that robots couldn't think in real-time, hence until recently planning and deduction have played little role in robotics. The rapid increase in computational power and our deeper understanding of the real difficulties of NP Hard problems has allowed us to revisit this belief. Our challenge is to develop algorithms for performing extensive deductions in real time, as an enabler to explorers that live long lives in unknown environments.

Impact: The future of space exploration depends on cheap, agile explorers that cannot only fly by planets, but are clever enough to roam asteroids or plumb the oceans of Europa, and then come back alive. Our focus is on developing an autonomous control systems for NASA's Messenger mission, targeted to launch towards Mercury within a few years.

On Earth more than 90% of all computers are used to control continuously operating, embedded systems. Hence a programming paradigm for making these embedded systems intelligent and robust will have wide impact on Earth and beyond.

Future Work: We plan on developing algorithms that reason from models of complex behaviors involving very large state space. We will be incorporating into our model-based programming language algorithms for planning and inferring hidden state, based on both symbolic encoding and graph-based planning. We will be testing out the model-based programming paradigm on the Messenger space mission, and the intelligent office.

Research Support: This work is funded by DARPA under contract number F33615-00-C-1702, administered by the Air Force Research Laboratory at Wright Patterson AFB.