

Progress on Three Projects: Systems for Spacecraft, Rovers, and Station Crew Return Vehicles

by Albert M.K. Cheng

ABSTRACT—Three projects are closely related with the ultimate goal of building fully-verified space vehicles that are (1) reliable, (2) energy-efficient, and (3) schedule-optimized. Timing response is a key issue.

1. Building and Verifying Fault-tolerant Autonomous Real-time Systems for Space Applications

Premises of the project

AUTONOMOUS SYSTEMS AND THEIR EMBEDDED AUTONOMY software in many NASA missions must perform correctly for an extended period of time and make real-time decisions that meet both logical and timing requirements. Furthermore, the autonomy software must tolerate implementation and environment-induced faults. Developing and verifying these systems are especially difficult because of the large (and often infinite) state space and execution sequences as well as the uncertainties in the environment in which these systems operate.

One focus of the NASA's Automated Reasoning thrust is to "enhance the autonomous decision-making capabilities of robotic explorers, spacecraft, and mission management systems." The objectives of this project are to address two key technology areas in the category of Automated Reasoning: (1) adding scalable fault-tolerance in the decision-making autonomy software of a real-time autonomous system and (2) augmenting the capability of formal verification strategies by providing an alternative based on scalable rulebase analysis to model checking and theorem proving.

Results

Since our approach can transform subsets of the code into self-stabilizing equivalents with different code modifications depending on their syntactic/semantic forms, it can be scaled to increasingly large and complex autonomy software systems. Coupled with the compositional analysis/verification strategy that identifies these syntactic/semantic code subsets applies to our overall approach further scales to deriving and verifying highly fault-tolerant autonomy software systems. Ongoing work evaluates this strategy on the modified Garglia simulation platform.

2. Optimizing System Reward in Battery-Powered Spacecrafts and Rovers

Premises of the project

RECHARGEABLE BATTERIES ARE USED TO OPERATE MANY SPACECRAFTS and autonomous rovers; consequently, their operational periods

are limited by their battery supplies before the next recharge. How to use this battery-supplied energy efficiently is a critical issue. Most existing energy-conserving techniques are based on dynamic voltage scaling (DVS) and consider only timing or energy constraints. In a more realistic scenario, we should simultaneously consider three constraints: time, energy, and reward (quality-of-service).

This project investigates two static methods (Greedy and Dynamic Programming) and an On-Line method for selecting tasks to optimize system reward while meeting timing constraints and conserving energy. We use simulation experiments to compare the performance of these methods with existing techniques. We have three static methods for selecting tasks from the task sets: (1) Simplified REW-Pack, (2) Greedy and (3) REW-Pack. We have compared the reward gained by these three methods. We found that our Greedy method often yields a larger total reward value than REW-Pack's. Besides, the Greedy method is more efficient than REW-Pack.

Results

In this project, we have developed a static method to schedule an overloaded, battery-powered system. We compared our methods to a previous method. We compared the performance of these four methods in many situations. Our conclusion is that the Greedy method often has a better performance than REW-Pack, especially when the system has more of an energy limit, number of processor frequencies, and number of tasks. In a future project, we plan to develop a combined method which is more suitable in more situations. We also plan to investigate the best scheduling choice for each system environment.

This project implements power-saving methods and investigates their performance by simulation. In particular, we have compared our static methods with REW-Pack, the only existing technique that deals with all three constraints (time, energy, and reward) but that does not perform well in overloaded systems. We believe that our algorithms are more suitable when the energy limit is higher when there is a larger set of processor frequencies, or whenever there is a larger number of tasks.

3. Timing Analysis and Scheduling of the X-38 Space Station Crew Return Vehicle and Other Space Vehicles

Premises of the project

THIS PROJECT HAS PERFORMED TIMING ANALYSIS AND SCHEDULING of the X-38 autonomous spacecraft built by NASA as a prototype of the International Space Station (ISS) Crew Return Vehicle (CRV). The avionics hardware and software design phase for this spacecraft requires tools for representing, analyzing, and verifying the hard real-time timing aspects of the system. To verify the planned performance of the safety-critical system functions, researchers have modeled a high-level specification of the X-38 multi-processor system task structure in Real-Time Logic (RTL) and Presburger Arithmetic representations. This timing analysis methodology can be applied to other space vehicles.

Results



COMPUTER STUDENTS—Bin Lu (l.), master's student and Yingwei Kuo, a Ph.D. candidate in computer science, focus upon the complex problem of programming computers so that commands operate on a real-time basis. Many NASA missions rely on software that is able to make real-time decisions that meet both logical and timing requirements.

We have investigated the X-38 201 vehicle avionics system development through its requirements and design phases. Although the system is intentionally designed to reflect deterministic software timing relationships, one or more tools yet required for modeling and analyzing critical system performance throughout the life-cycle. A scheduling tool similar to those evaluated is believed necessary to provide a means of easily analyzing possibly fluctuating workloads, to ensure that deadlines are met, and to provide a pictorial representation of the system timeline for communication.

Although none of the tools met all evaluation requirements, it was decided that rather than building a custom tool for this project, it would be best to choose one or more of the commercially available tools and try to extend it for our particular needs. RAPID RMA and TimeWiz were both chosen for further development based upon their current state of development as well as maintenance support.

In addition to a scheduling tool, the RTL representation seems to be a promising mechanism for satisfying similar timing analysis and verification tool requirements. The RTL representation presented here represents one aspect, task loop timing, of a complex avionic system. The specification language itself seems well-suited for representing this as well as possibly a broader range of areas of the system specification. For example, only one of the four FCC's high-level task structure is modeled. It may be possible to model redundancy aspects of the system as well as actual hardware devices in order to verify system fault tolerance. The RTL representation combined with the graphical constraint analysis mechanism seems to be a powerful enough

tool to represent many aspects of the system which may aid in timing, schedulability, fault, and safety analysis as well as verification. This methodology can be readily applied to verify the timing properties of other space vehicles.

Publications

Cheng, A. M. K. and F. Shang. "Priority-Driven Coding of Progressive JPEG Images for Transmission in Real-Time Applications," *Proc. 11th IEEE-CS Intl. Conf. on Embedded and Real-Time Computing Systems and Applications*, Hong Kong, August 2005.

Andrei, S. W., N. Chin, A. M. K. Cheng, and Y. Zhu. "Runtime-Coordinated Scalable Incremental Checksum Testing of Combinational Circuits based on #SAT Problem," *Proc. 11th IEEE-CS Intl. Conf. on Embedded and Real-Time Computing Systems and Applications*, Hong Kong, August 2005.

Andrei, S. W., N., Chin, A. M. K. Cheng, and M. Lupu. "Incremental Automatic Debugging of Real-Time Systems Based on Satisfiability Counting," *IEEE-CS Real-Time and Embedded Technology and Applications Symposium*, San Francisco, March 2005.

Cheng, A. M. K. and F. Shang. "Priority-Driven Coding of Progressive JPEG Images for Transmission in Real-Time Applications," *Proc. 11th IEEE-CS International Conference on Embedded and Real-Time Computing Systems and Applications*, Hong Kong, August 2005.

Cheng, A. M. K. and L. E. P. Williams. "Timing Analysis and Scheduling of the X-38 Space Station Crew Return Vehicle Avionics," submitted to *IEEE Transactions on Aerospace and Electronics Systems*, 2004.

Rice, L. E. P. and A. M. K. Cheng. "Timing Analysis of the X-38 Space Station Crew Return Vehicle Avionics," *Proc. IEEE-CS Real-Time and Embedded Technology and Applications Symposium*, Vancouver, Canada, June 1999.

Zu, M. and A. M. K. Cheng. "Real-Time Scheduling of Hierarchical Reward-Based Tasks," *Proc. IEEE-CS Real-Time Technology and Applications Symposium*, Washington, D.C., May 27–30, 2003.

Presentations

Cheng, A. M. K. and S. Fang. "Study and Simulation of a Distributed Real-Time Fault-Tolerance Web Monitoring System," *Proc. IEEE-CS Real-Time Systems Symposium (RTSS) WIP Session*, Miami, FL, December 2005.