Model Checking the Remote Agent Planner
Lina Khatib, Kestrel Technology; Nicola Muscettola, NASA; Klaus Havelund, Kestrel Technology

NASA Ames Research Center
Mail Stop 269-2, Moffett Field, CA, U.S.A.
{lina, mus, havelund}@ptolemy.arc.nasa.gov

Abstract
Verification and validation of autonomous spacecraft software is a critical problem. With more missions being designed and flown in a shorter span of time from mission conception to flight, traditional methods of validation and testing are struggling to remain effective. Model-based autonomous system software is potentially more complex since it strives to provide increased system robustness by correctly addressing a wider range of possible system behaviors. The model-based approach facilitates the elicitation and automatic manipulation of system level constraints. However, the models used in this approach still need to be verified, i.e., it is necessary to guarantee that no unintended consequences will arise.
This work tackles the problem of using Model Checking for the purpose of verifying the HSTS planning and scheduling framework of the Remote Agent autonomous control system that flew on Deep Space One [1]. Model checking is one of the most advanced verification techniques currently available for traditional software. Our approach consists in translating HSTS models into a system specification that is amenable to the application of a model-checking algorithm.
The most used model checking formalisms cannot easily represent constraints that are naturally represented by HSTS, namely continuous time and other continuous parameters. To allow model-checking algorithms to operate on HSTS models we provide a mapping between a subset of the HSTS Domain Description Language and a model checking formalism.
The model checker we use is UPPAAL, a system for the verification of timed automata whose formalism is close to the temporal planning framework of HSTS. UPPAL has been successfully applied to verify several real-time systems of industrial interest [2, 3].
The translation process identifies each separate state variable tracked by an HSTS model into a separate UPPAAL timed automaton. Each possible task type that can be scheduled on an HSTS state variable is translated into a main node in the corresponding UPPAAL timed automata. Moreover the main node is preceded and followed by two chains of auxiliary nodes that transition instantaneously. These nodes are a direct translation of temporal constraints that must apply respectively to the start and end of an HSTS task. A goal in HSTS corresponds to a property in UPPAAL. Similarly, a plan in HSTS corresponds to an execution trace in UPPAAL.
A preliminary application of the method to a sample problem of a rover moving between locations to pick up rocks has shown its usefulness and success.
We are currently working on identifying a set of verification properties that guarantee a certain degree of coverage for HSTS models and the underlying HSTS planning engine.

References