Abstract. To fulfill the needs of its deep space exploration program, NASA is actively supporting research and development in autonomy software. However, the reliable and cost-effective development and validation of autonomy systems poses a tough challenge. Traditional scenario-based testing methods fall short because of the combinatorial explosion of possible situations to be analyzed, and formal verification techniques typically require a tedious, manual modelling by formal method experts. This paper presents the application of formal verification techniques in the development of autonomous controllers based on Livingstone, a model-based health-monitoring system that can detect and diagnose anomalies and suggest possible recovery actions. We present a translator that converts the models used by Livingstone into specifications that can be verified with the SMV model checker. The translation frees the Livingstone developer from the tedious conversion of his design to SMV, and isolates him from the technical details of the SMV program. We describe different aspects of the translation and briefly discuss its application to several NASA domains.

1 Introduction

As NASA’s missions continue to explore Mars and beyond, the great distances from Earth will require that they be able to perform many of their tasks with an increasing amount of autonomy, including navigation, self-diagnosis, and onboard science. For example, the Autonomous Controller for the In-Situ Propellant Production facility, designed to produce spacecraft fuel on Mars, must operate with infrequent and severely limited human intervention to control complex, real-time, and mission-critical processes over many months in poorly understood environments [4].

While autonomy offers promises of improved capabilities at a reduced operational cost, there are concerns about being able to design, implement and verify such autonomous systems in a reliable and cost-effective manner. Traditional scenario-based testing methods fall short of providing the desired confidence level, because of the combinatorial explosion of possible situations to be analyzed.
Often, formal verification techniques based on model checking\(^1\) are able to efficiently check all possible execution traces of a system in a fully automatic way. However, the system typically has to be manually converted beforehand into the syntax accepted by the model checker. This is a tedious and complex process, that requires a good knowledge of the model checker, and is therefore usually carried externally by a formal methods expert, rather than by the system designer himself.

This paper presents the application of formal verification techniques in the development of autonomous controllers based on Livingstone, a model-based health management and control system that helps to achieve this autonomy by detecting and diagnosing anomalies and suggesting possible recovery actions. We present a translator that converts the models used by Livingstone into specifications that can be verified with the SMV model checker from Carnegie Mellon University. The translator converts both the Livingstone model and the specification to be verified from Livingstone to SMV, and then converts any diagnostic trace from SMV back to Livingstone. It thereby shields the Livingstone application designer from the technicalities of the SMV model checker.

Sections 2 and 3 respectively present the Livingstone health management system and the SMV model checker. Section 4 introduces our translator and describes its different parts. Section 5 discusses its application to several NASA projects, Section 6 develops some comments on the nature of the verification problem for autonomy model, and Section 7 draws final conclusions.

2 Livingstone

Livingstone is a model-based health monitoring system developed at NASA Ames \[^9\]. It uses a symbolic, qualitative model of equipment to infer its state and diagnose failures. Livingstone is one of the three parts of the Remote Agent (RA), an autonomous spacecraft controller developed by NASA Ames Research Center conjointly with the Jet Propulsion Laboratory. The two other components are the Planner/Scheduler, which generates flexible sequences of tasks for achieving mission-level goals, and the Smart Executive, which commands spacecraft systems to achieve those tasks. Remote Agent was demonstrated in flight on the Deep Space One mission (DS-1) in May 1999, marking the first control of an operational spacecraft by AI software \[^6\]. Livingstone is also used in other applications such as the control of a propellant production plant for Mars missions and the monitoring of a mobile robot.

The functioning of Livingstone is depicted in Fig. 1. The \textit{Mode Identification} module (MI) estimates the current state of the system by tracking the commands issued to the device. It then compares the predicted state of the device against observations received from the actual sensors. If a discrepancy is noticed, Livingstone performs a diagnosis by searching for the most likely set of component

\(^1\) As opposed to those based on theorem proving, which can provide even more general results but require an even more involved and skilled guidance.