Chapter XVIII
High-Quality Software Models of the Mid-Infrared Instrument for the James Webb Space Telescope

Jane M. C. Oh
Jet Propulsion Laboratory, California Institute of Technology, USA

Martin S. Feather
Jet Propulsion Laboratory, California Institute of Technology, USA

Mori A. Khorrami
Jet Propulsion Laboratory, California Institute of Technology, USA

ABSTRACT

This chapter examines the experience of using model-based design in the context of development of critical software. The software is being developed to control a science instrument that it to fly as part of NASA’s James Web Space Telescope. The chapter discusses the context and nature of this software development effort, and why they motivated the choice of a model-based development approach. Illustrations are provided of the elements of model-based design that are proving to be beneficial. The chapter also considers how software assurance practices are being adapted to work with this approach.

1. INTRODUCTION

The work described in this chapter covers a software development effort being conducted at the Jet Propulsion Laboratory, software for control of an instrument that is to be a part of a deep-space observatory. While at first glance this may appear to be far removed (in not only physical distance) from concerns that permeate conventional software development, in fact many of the key factors
will be familiar concerns to many developers of terrestrial software. As we will discuss, it was consideration of these factors that led to the choice of model based development as the software development approach to follow. We will describe how our experiences to date suggest this was a good choice. We will also describe some of the ramifications of this choice – changes to existing software development practices that are occurring to accommodate the changed nature of software development stemming from adoption of model based development.

The software development effort discussed in this chapter is the ongoing development of “Flight Software” (i.e., software that will be executed by computers on board the spacecraft) for a science instrument, the Mid Infra-Red Instrument (MIRI). As its name suggests, it will measure light in the mid-infrared spectrum, for astronomy purposes. MIRI is one of four science instruments that together form the Integrated Science Instrument Module (ISIM) element. ISIM is in turn one of three elements that comprise the Observatory Segment, and that in turn is one of the three segments that comprise the James Webb Space Telescope (JWST). This is a large space telescope, scheduled for launch in 2013. It is being developed jointly by the United States’ National Aeronautics and Space Administration (NASA) and the European Space Agency; for details, see http://www.jwst.nasa.gov/.

This chapter will focus on the main challenges that pervade the development of this mission-critical software.

2. BACKGROUND AND CHALLENGES

NASA’s spacecraft operate across the solar system performing a wide range of missions, including planetary surface exploration using rovers, planetary observation using orbiters, and solar and astronomical observations using observing platforms at a variety of locations. As this fleet continues to expand in both number and complexity, there is a pressing need for increasing the role that on-board spacecraft software plays in controlling those spacecraft. This will relieve the growing burden levied on Earth-based ground control, and alleviate the bottleneck of communication back to earth. Furthermore, increased autonomy on the spacecraft themselves will enhance their capabilities – responding autonomously to investigate interesting but short-lived phenomena, increasing the capabilities of surface rovers to traverse terrain and to perform more intricate science experiments, etc. This all requires more sophisticated software in the spacecraft/rover itself. Without some change in the software development process, such more complex software systems will take longer to develop and verify, potentially increasing overall mission cost, introducing delays, and potentially introducing additional risk. Model based design is being investigated as a means to improve the development process for spacecraft software so as to help overcome some of these problems.

The work reported here describes ongoing use of model-based design in a spacecraft setting, and the particular areas of benefit it conveys. Because of the risk-averse nature required of the development of NASA’s spacecraft (in most cases well beyond reach of rescue or repair), ensuring the quality of the software remains of paramount importance. Also reported here is how software assurance techniques are being adapted to work with model-based design. One of the important synergies between model based design and software assurance is the opportunity to detect an increased fraction of software related defects earlier in the development process than would otherwise be the case, thus leading to net savings of cost and schedule, while ensuring quality.