==========

Draft Proposal & Project Plan

Guidance and Control Software Specification

February 18, 1999

CS 533 – Formal Methods of SW Eng.
Jim Carole

Group Project
Spring 1999
Prof. Rick Sheldon

Douglas Clapp
Steve Dankanich
Jerri Levison
Brian Scriber

1 Introduction

The class has been given the Software Requirements Specification for the Guidance and Control Software (GCS) used on the Viking Mars Lander. The specification defines the guidance and control system for the planetary landing vehicle during its terminal phase of descent and represents its on-board navigational software. The software developed from this specification serves two main purposes:

- Provide guidance and engine control of the vehicle during the terminal phase of descent onto a surface.
- Communicate sensory information about the vehicle and its descent to some other receiving device.

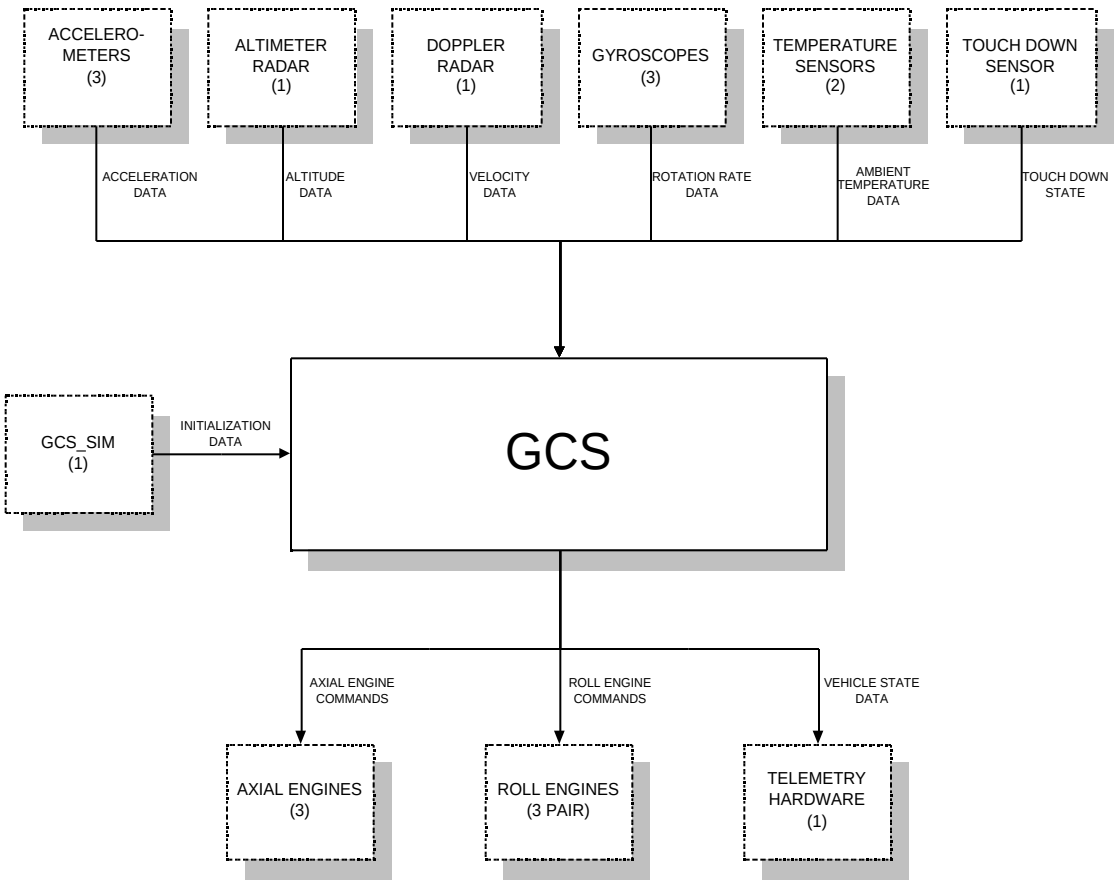
1.1 Problem Statement

The assignment for the class is to take the Software Requirements Specification for the Viking Mars Lander Guidance and Control Software and to rewrite part of it. The original specification was written using the old B5 style. The new specification is to be written using Zed. Zed is a formal specification language that uses mathematical notation to describe in a precise way the properties that a system must have, without constraining the way in which they are achieved. After a careful review of the specification, we are to select parts of it and specify the functionality of those parts using Zed. This document proposes the parts that we (team #2) wish to specify using the Zed notation.

2 Proposed Functionality to Cover Using Z Specification

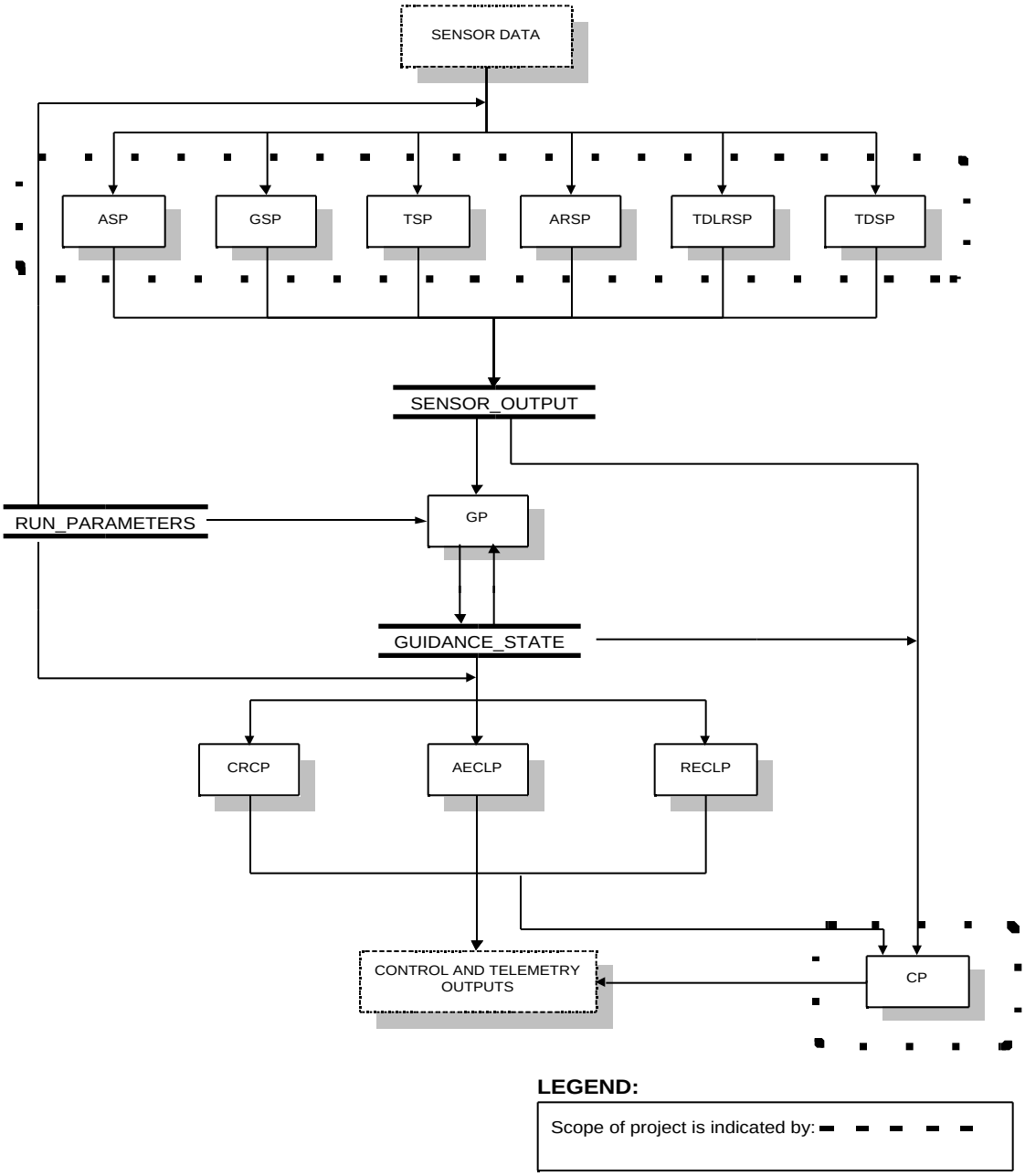
The following context diagram shows the high level functional scope for the GCS system. The top most part of the diagram contains the data sensors for the system. This is also shown in the expanded diagram following this high-level context diagram. (pg 4)

GCS High Level Context Diagram



The parts we propose to develop schemas for are shown in the diagram that follows. Basically, parts outlined with a heavy dotted line are those we propose to cover. This corresponds exactly to those functional units, which make up the Sensory Processing Subframe, also known as Subframe 1.

GCS Context Diagram



In addition to the Sensory Processing Subframe, we propose to develop two additional schemas. One will be for the initialization of the GCS system, referred to as init_gcs in the current specification while the other will consist of a schema to handle the scheduling of the three subframes for the system (Schedule schema). To summarize we propose the following schemas:

<u>Proposed Schemas</u>

Init_gcs (High level schema)

Scheduler (High level schema)

Used to schedule processes for the Sensor Processing Subframes, Guidance Processing Subframes and Control Law Processing Subframe. Subframe 2 and Subframe 3 (Guidance Processing Subframe and Control Law Processing Subframe, respectively) ***will not*** *be expanded beyond one high level schema. Subframe 1 will be expanded to include all of the following schemas:*

ARSP (One high level, three sub-level schemas)

Rotate variables

Determine altitude

Perform alternate processing even number frame

ASP (One high level, five sub-level schemas)

Rotate variables

Adjust gain for temperature

Remove characteristic bias

Correct for misalignment

Determine acceleration

CP (One high level, one sub-level schema)

Construct packet

GSP (One high level, three sub-level schemas)

Rotate variable

Adjust gain

Convert G_Counter

TDLRSP (One high level, seven sub-level schemas)

Rotate variables

Perform alternate processing even number frame

Determine radar beam states

Determine radar beam velocities

Process radar beam velocities

Convert to body velocities

Set values in K_Matrix

TDSP (One high level, one sub-level schema)

Determine status of touchdown sensor

TSP (One high level, three sub-level schemas)

Calculate solid state temperature

Determine solid state or thermocouple temperature

Calculate thermocouple temperature

In this set there are 32 schemas, which fully encapsulate the Sensor Processing Subframe together with the initialization of the system and scheduling of subframes. By adding any additional schemas we would violate the natural delineation of complexity. Adding part of a subframe would be impractical because partitioning the system as we have done appears to be well within the scope of what can be accomplished during the time left in this course.

2.1 Constraints

Equations:

Equations will be handled by using the GCS defined name of the equation for the function. The function will be followed by the parameters, by name, which will be used. The following example shows how we will use this:

(page 44 of GCS specification)

AR_ALTITUDE is a function of AR_COUNTER and AR_FREQUENCY

We will represent such an equation in our schema as:

Func_AR_ALTITUDE(AR_COUNTER, AR_FREQUENCY)

This approach will allow us to show dependencies between equations and schemas. In addition, this approach could easily be extended to include development of schemas for each equation in the future.

Development Environment:

The document format we will be using for all work will be Microsoft Word. We will be using font substitutions for all Z notations required.

3 Schedule

Here is our preliminary schedule. High level and low-level (sub-level) schemas are those identified in section 2.

Task	Date
Draft Proposal & Project Plan	2/18/99
High Level Schemas	3/08/99
SRR Review (with slides, 15 min. presentation)	3/09/99
Low Level Schemas	3/31/99
Draft SRS	4/01/99
Error Checking	4/15/99
Final SRS	5/06/99