

Utilizing the CoRE Behavioral Model for CubeSat Software Requirements

14th Annual CubeSat Workshop

Noah Weitz

San Luis Obispo, CA

Friday, April 28, 2017



What Makes CubeSats Different?

Two factors separating CubeSats from other aerospace projects:

1. Project timeline
2. Organizational Scale

CoRE Method

- Stands for Consortium Requirements Engineering method
 - Developed by the Software Productivity Consortium
- Based on principles of object-oriented programming
- Two model structure
 - Behavioral
 - Class
- Requires precise description of acceptable software behaviors
- Requires familiar language for easy understanding

Behavioral Model

- Requires software to be viewed within the environment context
 - What affects the software
 - What interfaces with the software
- Two environmental quantities
 - NAT – based on nature
 - REQ – based on what the system must enforce



Modifying CoRE for CubeSats

1. Utilize behavioral model to collect data on environmental quantities
2. Separate the monitored and controlled variables
3. Formalize the requirements into statements

Collect Environment Quantities

- Determine quantities
 - NAT and REQ
- Collect information on each quantity.
 - **Name**- name of quantity
 - **Type**- what is measured
 - **Value**- potential ranges
 - **Description**- importance for either monitoring or controlling

Example: temperature

- **Name**- Temperature
- **Type**- volts
- **Value**- 0.4-2.4 volts
- **Description**- Measures temperature of the spacecraft for health purposes

Separate Environmental Quantities

- Determine whether the quantity is monitored or controlled
 - How is the quantity being used on the spacecraft?
- Example: Temperature is being measured for health purposes
 - Monitored quantity

Formalize Requirements

- Compile information into formal statements
 - “shall” statements
- Organize formal statements
 - Keep controlled and monitored separate
- Example: The system shall measure the temperature within a range of 0.4 to 2.4 volts to track the health of the spacecraft.

Future Work

- Integrate and modify the class model
 - Allows for requirements or portions of requirements to be reused.
 - Saves time for future requirement development
- Integrate the use of Computer-Aided Software Engineering (CASE) tools
 - Further streamline requirements development process



Questions?

