CubeSat Software Requirements Engineering

Nicholas Alexander[1], Dr. Hassan Reza[2]
[1]Weisberg Division of Computer Science, Marshall University
[2]Department of Computer Science, University of North Dakota

Background

Imagine for a moment that you are a construction worker hired by a company to construct a new building, but the company has failed to supply any type of floor plan for the building. Instead, the company has asked you to "do your best." Once you completed the building, the company decides that they do not like a set of rooms—but at this point it is too late to change. In reality, an architect would supply a floor plan that the construction worker would follow precisely to ensure the company is satisfied. Software engineering follows a similar process.

The first step in creating any complex software system is creating a formal requirements specification which outline the functionalities and qualities of the system [1]. Without a proper understanding of the problem, a blindly implemented solution is crippled from the beginning. The requirements—which are agreed on by both client and vendor—serve as a nonpartial basis for determining if a software system is satisfactory.

Quality Attributes

The concept of a quality attribute is simple: describe how well the system must perform a specific function or task. Quality attributes must maintain two core properties: measurability and verifiability. For every attribute, it must be possible to assign a non-arbitrary value denoting the quality of performance. The performance values must then be used to verify that the system has satisfied some set of standards.

There are several categories of quality attributes, but this work focused on three specific categories: availability, performance, and security. Availability relates to all efforts to ensure the system remains online and able to serve requests. Performance details the responsiveness of the system. Lastly, security encompasses all methods of keeping the system and data protected.

The goal of this research was to generate a small, yet detailed, set of quality attribute scenarios describing a CubeSat systems.

Sample Quality Attribute Scenarios

Availability

Source	Communication Window
Stimulus	Satellite Orbit
Environment	Within Communication Range
Artifact	System Communications
Response	Communication established
Response Measure	Communication should be establish at least once every 24 hours

Figure 1: Communication Window

Source	Communication System
Stimulus	Signal Interference
Environment	Degraded Radio Waves
Artifact	System Communications
Response	Notify ground station operator about signal interference
Response Measure	System should recognize degraded communications with 65% accuracy

Figure 2: Radio Interference

Performance

Source	Job Processing
Stimulus	System Begins Processing Job
Environment	Normal Operations
Artifact	System Controls
Response	System must complete a job and begin a new job to prevent process starvation
Response Measure	System should complete any single job within 48 hours

Figure 3: Job Processing

Source	Data Transmitter
Stimulus	Image Downlink Requested
Environment	Within Communication Range
Artifact	Image Data Transmission
Response	Image is sent to the ground station
Response Measure	Transmitter should be capable of sending a single JPEG image during a communication window [2]

Figure 4: Transmission Rate

Security

Source	Authorized Individual
Stimulus	Requests System Access
Environment	Normal Operations
Artifact	System Controls
Response	Authenticated user is granted access
Response Measure	System should always grant access to an authenticated user

Figure 5: Successful Authentication

Source	Environment
Stimulus	Fatal Ground Station Event
Environment	Critical Conditions
Artifact	System Controls
Response	System controls are transferred to a remote ground station
Response Measure	Remote ground station must establish communications with satellite within 48 hours of the event [3]

Figure 6: Disaster Recovery

Conclusion

This research has produced, in total, roughly a dozen quality attribute scenarios. These scenarios have not covered every possible scenario that the system may encounter but instead a rough overview of major components and subsystems. Another important note is that all values used in creating these quality attribute scenarios have been estimated or referenced from other similar systems. Each of these values may change after actual production of a CubeSat.

Unfortunately, a major goal of the research was not met. A set of use case scenarios were initially intended to be related to a set of quality attribute scenarios in manner more advanced than simple citations. Due to unforeseen circumstances, this goal was not achieved. This issue will likely be addressed in future additions to this work.

Future Work

In the future, the combination of use case and quality attribute scenarios will be used to intelligently design a software architecture for a CubeSat.

Acknowledgements

This research was funded by United States National Science Foundation (NSF Award #1359224) with support from the United States Department of Defense.

This research was sponsored by the University of North Dakota under the guidance of Jeremy Straub and Dr. Ronald Marsh.

References

[1] IEEE, *"Guide to the software engineering body of knowledge,"* 3rd ed., IEEE Computer Society, 2014, Online.

[2] G. Garriott, "Low earth orbiting satellites and internet-based messaging services," in INET '96, Internet Society, 1996, Online.

[3] D. Ichikawa, "Cubesat-to-ground communication and mobile ground station development," Online.