Asteroid Exploration System Result Document

Date: Dec 12th, 2017
Author: Stephen Akaeze
Reviewers: Joseph Kintzel and David Hu

Comments from peer design review

1) Looking over your AsteroidSpecs class: Asteroid "IS A" AsteroidSpecs? I think I would just have those parts be instance data for your Asteroid objects, as they are what make each instance unique. I'm also a little confused by your Discovered class. These are messages(?), so maybe a more descriptive name might help with the readability. I would suggest that Asteroid be composed of Discovery, then have Life, Mineral, and H2O Extend Discovery.

2) Don't forget to implement authorization tokens into your external interface methods for each module. (Asteroid Inventory Service) Also don't forget to protect your update method with an authorization token for the mission control service that monitors spacecraft messages.

3) You have the Spacecraft being composed of Missions. I think a Mission might be an entity that is assigned a Spacecraft, or perhaps multiple spacecraft. I'm not sure composition makes sense between these two things. Perhaps aggregation, in the other direction. In fact, I might draw a line between the mission control and distribution/storage of mission and the spacecraft. I believe those should be two separate modules based upon the design requirements.

4) Missionspecs is kind of mysterious.

Design Patterns Used

The design pattern used by my Asteroid Exploration System implementation are as follows:

- Singleton Pattern (AIS, RSMS, MMS)
- Observer pattern (RSMS, MMS, CCUI)
- Factory pattern (RSMS, MMS)

Modular Approach Usefulness and support

The modular approach was helpful in implemented secured subsystem method requests. They are enable multiple levels of the subsystem method allowing the user to implement less secure methods underneath the service interface. Each module is a service

provider leading to a good design-and-conquer implementation. The service interface also enables a good level of abstractness

Modular redesigns

There were several module redesigns when the CCUI requirements were being addressed and designed. This includes ensuring that the lookup methods return objects that contain enough information to maintain the data validity in the CCUI.

Design Review usefulness

The design review was extremely helpful as it forced me to provide more clarification on my class diagrams and class dictionary for improved third person understanding

Authentication Service Reuse

The Authentication Service is instrumental in the smooth functionality of the AES implementation. It greatly augments the service interface to ensure that the system is not misused by any third party.

Potential Implementation

My AES implementation certainly provides adequate information for any development team to make it a reality.