North Dakota State University--Fargo

From the SelectedWorks of Jeremy Straub

December, 2013

Software for OpenOrbiter

Christoffer Korvald
Jeremy Straub
Scott Kerlin
Ronald Marsh

Available at: https://works.bepress.com/jeremy_straub/153/
The software development effort for the OpenOrbiter project consists of four teams: operating software development, payload software development, ground station software development and testing. These teams are designing and developing the software required to create a turn-key spacecraft design which can be produced at a price point of under USD $5,000 by faculty, students and researchers worldwide. Through this process, students are gaining valuable real-world experience in areas of indicated interest. Each team is headed by a team lead who is responsible for conducting weekly meetings and organizing the activities of the team. During the Fall, 2013 semester, team leads were aided by students in the CSCI 297—Software Project Management Through Experiential Learning class who acted as project management consultants to the teams. The team members also collaborate through the GitHub source code management system, in addition to via in-person interactions.

The operating software is responsible for the autonomous operation of the spacecraft while out of range of the ground station (and largely while in range as well, though, theoretically a controller could override the autonomous decision making) and communications while in range. It will be responsible for controlling and commanding the subsystems and turning them on/off at the appropriate time in addition to monitoring the general health and status of the satellite and taking appropriate corrective actions as required.

The payload software is responsible for processing the images acquired by the camera. It will perform super-resolution and mosaicking to maximize quality (via enhancement of multiple images) and minimize data (via the removal of unneeded and overlapping areas) that has to be transmitted to the ground.

The ground station software will track the spacecraft, transmit commands to it and receive data from it. It will control the ground station antenna to orient it in an appropriate direction to maximize signal strength. It will also provide the system’s user interface.

The testing team is developing validation standards against which all of the software components will be measured and developing applicable tests at numerous levels (i.e., unit, integration, system and usability) to validate the performance of the system. Testing team members actively participate in the other three teams in addition to their testing team duties.

The software development effort for the OpenOrbiter project consists of four teams: operating software development, payload software development, ground station software development and testing. These teams are designing and developing the software required to create a turn-key spacecraft design which can be produced at a price point of under USD $5,000 by faculty, students and researchers worldwide. Through this process, students are gaining valuable real-world experience in areas of indicated interest. Each team is headed by a team lead who is responsible for conducting weekly meetings and organizing the activities of the team. During the Fall, 2013 semester, team leads were aided by students in the CSCI 297—Software Project Management Through Experiential Learning class who acted as project management consultants to the teams. The team members also collaborate through the GitHub source code management system, in addition to via in-person interactions.

The operating software is responsible for the autonomous operation of the spacecraft while out of range of the ground station (and largely while in range as well, though, theoretically a controller could override the autonomous decision making) and communications while in range. It will be responsible for controlling and commanding the subsystems and turning them on/off at the appropriate time in addition to monitoring the general health and status of the satellite and taking appropriate corrective actions as required.

The payload software is responsible for processing the images acquired by the camera. It will perform super-resolution and mosaicking to maximize quality (via enhancement of multiple images) and minimize data (via the removal of unneeded and overlapping areas) that has to be transmitted to the ground.

The ground station software will track the spacecraft, transmit commands to it and receive data from it. It will control the ground station antenna to orient it in an appropriate direction to maximize signal strength. It will also provide the system’s user interface.

The testing team is developing validation standards against which all of the software components will be measured and developing applicable tests at numerous levels (i.e., unit, integration, system and usability) to validate the performance of the system. Testing team members actively participate in the other three teams in addition to their testing team duties.