Final Project Proposal- Team 9 (KubeSat)

Due: Monday, February 5, 2018, 8:00 AM (submit to BlackBoard, under Assignments)

File Type: Microsoft Word

Team Name: KUbeSat (Team 9)

Team Members and email addresses:

Jeff Anderson jeffery.anderson@ku.edu

Patrick Beuhler pbeuhler@ku.edu

Yue Dong y742d808@ku.edu

Blaine Harris blaine.harris@ku.edu

Jhoel Perez j766p038@ku.edu

Harrison Wynn harrisonwynn@ku.edu

Contact:

Project Sponsor: Dr. Jerry W. Manweiler

Project Description:

 Space weather can have widespread, serious impacts on everyday life on Earth. It can disrupt human safety, including electrical grids, GPS, communications, and the internet. Studying these changes can be beneficial towards improving and safeguarding those systems. Through a NASA sponsored project, our team will combine efforts with several others to launch a cube satellite. This satellite will collect information for a variety of projects, in particular information about energetic particles. The probe will collect both high energy and low energy particle data, both in high resolution. Total energy information and particle species information will also be collected. Data collected from a near Earth orbit accurately portrays the Sun's activity. Our team would like to collect, parse, and prepare this data for analysis. We have partnered with Dr. Jerry Manweiler and Fundamental Technologies. Through his experience in the field, we will make a C# application to perform all the desired operations, and help advance knowledge about energy fields surrounding the Earth.

Project Milestones:

* Detailed hardware to software requirements (October 2017)
* Use case diagram development (October 2017)
* UML diagrams (including class diagrams, state diagrams, communication diagrams, activity diagrams) (November-December 2017)
* Software Implementation (December 2017-March 2018)
* Testing (March 2018-April 2018)
* Delivery (May 2018)

Figure 1. Gantt Chart of the KUbeSat Timeline





Project Budget:

* C# in a Windows .NET environment (free on campus)
* Team Foundation Server for version control (free)
* SQL database hosted through FTECS (free)
* Training on C# (free, self-taught)

Work Plan:

* Jeff- C# trainer, documentation, implementation of REST API
* Patrick- Organizing tasks, communicating, documentation, implementation of Controller Class
* Yue- Documentation, implementation of L1 Conversion Class
* Blaine- Organizing tasks, communicating, documentation, implementation of L0 Conversion Class
* Jhoel- Documentation, implementation of L2 Conversion Class
* Harrison- Documentation, implementation of database and testing

Team Foundation Server llink:

<http://kubesat.ftecs.com:8080/tfs/kubesat_teamcollection/KUbeSatTest1/KUbeSatTest1%20Team>

Login: Patrick.Beuhler

Password: KuB3S@t17E

Final Project Design

The KUbeSat is a small satellite that will maintain an orbit close to Earth. Onboard sensors will detect the presence of a variety of energetic particles at various resolutions. Because sensors are built with specific goals in mind, data will be high resolution with respect to time or energy, or counts of particle occurrences. Telemetry data and magnetometer will also be collected. Our software acts to organize, process, and redistribute this raw satellite data. It has access to the data from the satellite, and will download and send the data for storage. Downloads shall be manually requested or performed on an automated basis. Automated downloads will be performed to avoid a backlog of information. Data will be requested from the source (satellite, magnetometer source). If new data is available, it will be downloaded and placed in the database. Because the probe is in space and subjected to many different stimuli, timing data from these raw measurements may vary. The timing data recorded by the probe will be checked against an external clock and if it matches, the next entry will be retrieved. If the times do not match, the correct time will be written to the satellite data, and the next entry will be retrieved.

After storing this raw data, to integrate it into a localized system, the data will be organized according to category and destination. Data can now be further processed for scientific analysis to varying levels. L0 is the instrument data/raw counts from sensors, L1 is the count rate, and L2 is the intensity/flux. Formats for data levels 1-2 will be consistent with NASA’s common data format (CDF). For L1 data to be generated, L0 count data will be retrieved from the database, along with timing data and calibration data. The count rate can be calculated from these two values. The calculated values will be uploaded to the database, L0 data marked as processed, uploaded to a local file system in preparation for calculating L2 data, and the next set of L0 data will begin to be processed. Processed L1 data will also be uploaded for public use.



Figure 2. Activity Diagram of L0 to L1 Dataflow (similar implementation for L2)

L2 data will be processed in a similar fashion. Data in the local file system from L1 will be used to calculate differential flux. The governing equation is

 $dJ = R/(g△E)$

Where R is rate (from L1 data), g is the geometrical factor of the instrument, and $△E$ is the energy bandwidth. Similarly, once L2 data is calculated, it will be stored in a public database, a local file system, and the L1 data marked as processed to avoid repeated calculation of old data.

Figure 3. Information Flow Diagram



Chart 1. Deliverable data levels, formats, and frequency of delivery

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Data Level  | Product Title | Contents | Volume(estimated) | Format | Latency | Frequency |
| L0 | Telemetry | Telemetry received from KUbeSat | 414 MB/day-TBR | Binary | From Receipt(T0) |  daily |
| L1 | Rates | Sorted, calibrated counts/second | 750 MB/day-TBR | CDF, CSV | T0 + <14 days | daily |
| L2 | Flux | Sorted, calibrated, in proper unit fluxes | 1200 MB/day-TBR | CDF, CSV | T0 + <1 month | daily |

The software will contain 4 main components. There will be the database, the public interface, classes that convert between L0-L2, and a controller class that orchestrates communication between all the others. Our main architecture will be based on a pipe and filter model, stored in a repository or client server model. Data will be streaming from the satellite, and therefore needs to be piped constantly, validated, and stored for further use. This is the highest priority because the data is irreplaceable. The L0 class receives packets of information, determines which are relevant, interprets the information, and outputs the result for database storage. Once the data is safely stored in the database, the software will be free to generate a consistent stream of publicly accessible/readable data (L1 and L2). This will also be performed on a regular basis. Requests shall be made through a website and fulfilled by our system. Each of the L1 and L2 classes will be fed the appropriate information from the stored database, perform the appropriate conversions, and return information for storage. The controller class makes calls to each of the L0, L1, and L2 classes, provides the necessary inputs from the database, and stores the returned information. In order to help with organization, two separate databases will be utilized. One will store raw and converted data products. This will be used for rapid access for the conversion classes and the public interface. Another database will be used to keep track of metadata. It will help organize by knowing how far each piece of information has been processed, when new data has entered the system, and when data needs to be re-converted.

The project will be implemented in C# in a Windows .NET environment. This takes advantage of existing architecture from previous projects from Fundamental Technologies, LLC (Dr. Manweiler’s company). This will help with validation testing, as the software may be tested against previously collected data sets for accuracy. Information from the probe, telemetry data, and additional sources (magnetic field data, priority event data) will be input to the software. The information requested from the satellite will be stored in a database, such as SQL, and made available for requests both from our software for converting L0 data farther down the pipeline, as well as making the calculated data (L1, L2) available for public use. Our software will make automated requests to this database, checking for validity as well as if it has already been utilized. Data will be publicly available for download requests. We will set up a website using a RESTful API to handle requests to our database for L1 and L2 data.

Ethical and Intellectual Property Issues

 The ethical issues with the KUbeSat are straightforward. Because it will be used for a scientific mission, there are few ethical issues. There is no conflict of interest to make money because the intent is to make scientific data publicly available, and the project is being funded by government organizations. There are no real users of the software because it is for scientific data collection, so there are no privacy concerns.

 There is a chance for misleading the public. If our data is incorrect, this could negatively impact the space science world. We will strive to ensure this does not occur by testing our software against previously collected data and comparing the results. This verification will help guarantee the validity of our results before public release.

 The general use and application of this software is restricted for the common user, therefore only people with the correct equipment to handle all the data collection and processing would make use of our program. While the public has will have access to L1 and L2 data, as is the goal of the project, L0 data has telemetry information, which is protected. Even though this is a sponsored project, we have been given the right to adapt the implementation of the software to our collective skillsets. However, the management organizations through which we were provided this opportunity, The University of Kansas and FTECS (Fundamental Technologies, LCC), would acquire the right to modify or reuse it for whatever they see fit. Our team will still own rights to the software.

 The principal property issues we face are if the redirection of data occurs anywhere within our software or if access to communicate with the CubeSat is passed on to other entities unrelated to either management or our team members. Given that the CubeSat is physically owned by NASA, any information extracted from it is rightfully theirs and therefore the distribution terms and conditions they have set will come into play. This threat would be avoided by blocking or restricting external IP addresses that can operate using the software.

Change Log

2/4/2018: Updated Gantt chart to more accurately reflect project timelines, the repository updated to include current location and login information, increased details on individual roles for implementation, updated budget items, updated final project design• Explain what parts of the Initial Project Description were changed and why
• 1-2 sentences per change,