

Building a Community of Open Source Contributors

Jay Trimble

Andrew Henry

The Vision

Open MCT represents a first step of a broader vision of shared ground systems. The vision is simple yet powerful. Move space operations software, that which we use to fly our missions, from closed proprietary systems to open systems. Use this open platform as the basis to build a community in which we build on each others work, where the space operations community is the beneficiary.

For decades, many of our ground systems have made little fundamental progress. Yes they look a bit more modern than they once did, and they use modern software technology, however, the basic tenets of how we plan and execute missions have not fundamentally changed since the Apollo era. This is in part, a great tribute to the remarkable work done by those that sent us to the moon. But it is also a symptom of an insular community built on proprietary software and methods from an earlier era. That era served us well, and built the foundations of space operations, yet, modern software technologies beckon with new opportunity.

Software technology is changing at an ever more rapid pace. Commercial companies are starting their own space ventures. Machine learning, AI and autonomy provide possibilities for new operational paradigms. Perhaps we are on the verge of new and more powerful operational paradigms. It is our goal to make freely available open source software a means to share and build on each other work.

Background

This paper is the sequel to the SpaceOps 2014 paper Open Source Software for Mission Operations - Technology, Licensing and Community [Jay Trimble]. That paper described the process of making our mission operations visualization software open source, the legalities, licensing, software repositories and basic capabilities. In 2014, we touched on the idea of building a community of users and contributors. This paper reports on progress in building a community, from a user and developer perspective, and follows up on a fundamental unanswered question from 2014 - is it possible, practical and beneficial to build an open source community in this domain. Is it practical to build an open source community around software for

visualization of mission data? Would contributors come? Would users and mission organizations adopt the software. Four years later there is a much richer story to tell.

The software

We embarked on building a new platform to build mission enhancing capabilities for space operations. Technically, we required the following:

- Composition without programming
- Data integration
- All of your data in one place
- View your data your way
- Operations unconstrained by domain walls

The Open MCT code base is on GitHub at <https://github.com/nasa/openmct>.

An informational site, with a live demo is at <https://nasa.github.io/openmct/>. Click on try it now to go to the live demo.

Composition

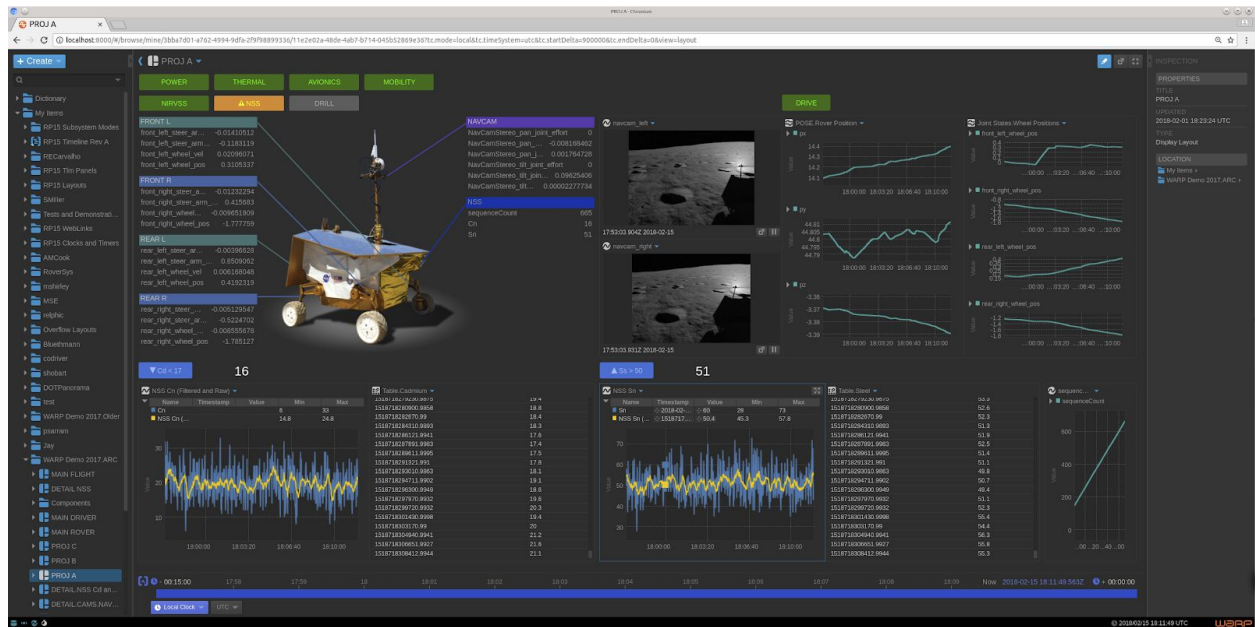


Figure 1 - A composition, built by a user

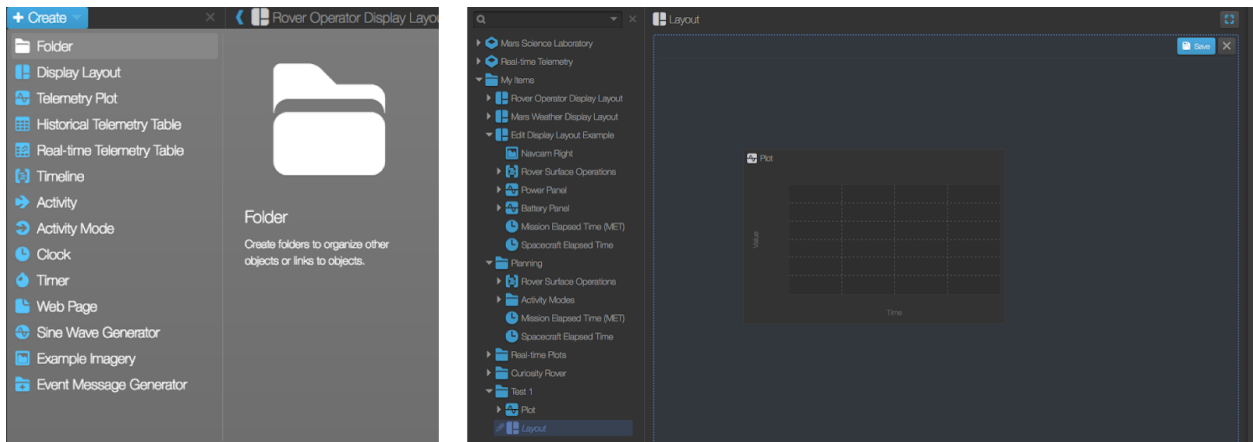


Figure 2, Create, then compose in a layout

Composition enables users to build their own displays without programming. Figure 1 shows an example composition from the Resource Prospector Lunar Rover Mission. Note the different types of data combined in one display. Figure 2 shows

Data Integration

Data integration refers to the capability to integrate data from multiple data sources and integrate and display that data in one environment.

All of your data in one place

The data explorer in Open MCT, shown on the left side of figure x, has all of a users data in one place. From there users may compose visualizations in layouts.

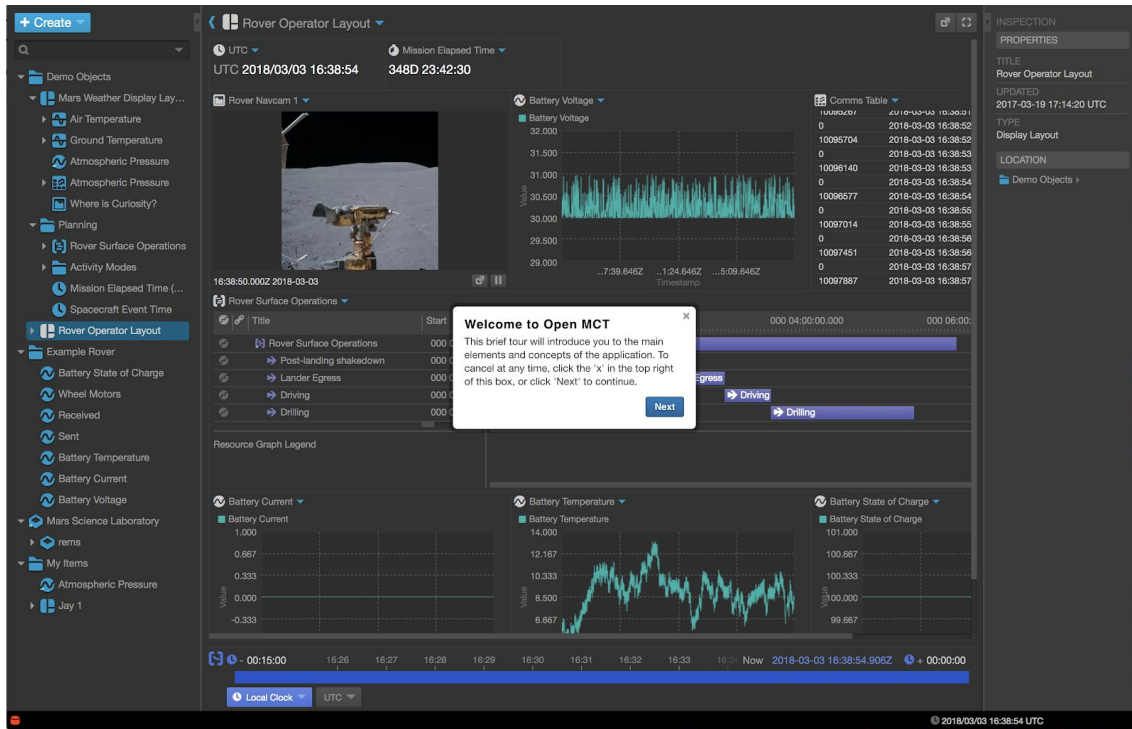
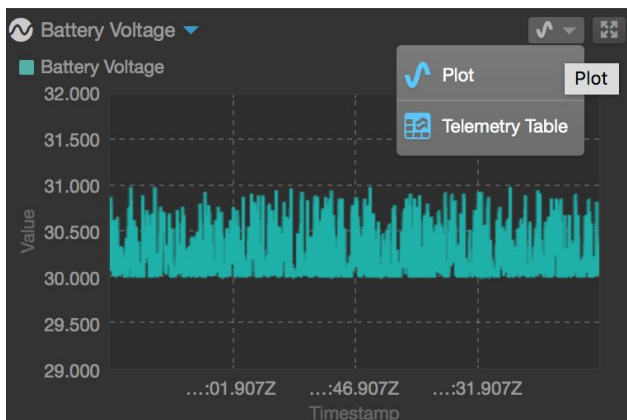
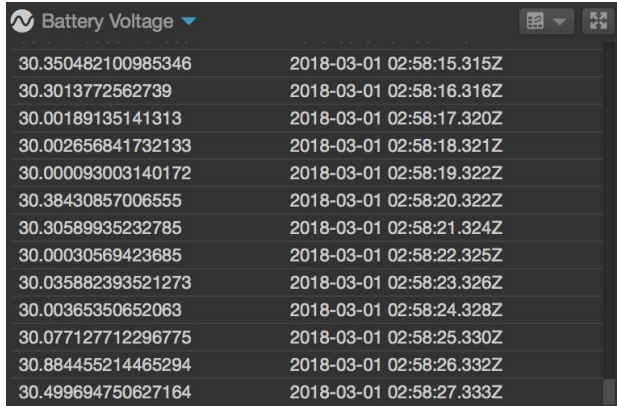


Figure x - The data explorer on the left has all of a users data in one place. This is from the open demo, available on line at <https://nasa.github.io/openmct/> (click on the try it now button).

View Your Data Your Way





| ID | Timestamp |
|--------------------|--------------------------|
| 30.350482100985346 | 2018-03-01 02:58:15.315Z |
| 30.3013772562739 | 2018-03-01 02:58:16.316Z |
| 30.00189135141313 | 2018-03-01 02:58:17.320Z |
| 30.002656841732133 | 2018-03-01 02:58:18.321Z |
| 30.000093003140172 | 2018-03-01 02:58:19.322Z |
| 30.38430857006555 | 2018-03-01 02:58:20.322Z |
| 30.30589935232785 | 2018-03-01 02:58:21.324Z |
| 30.00030569423685 | 2018-03-01 02:58:22.325Z |
| 30.035882393521273 | 2018-03-01 02:58:23.326Z |
| 30.00365350652063 | 2018-03-01 02:58:24.328Z |
| 30.077127712296775 | 2018-03-01 02:58:25.330Z |
| 30.884455214465294 | 2018-03-01 02:58:26.332Z |
| 30.499694750627164 | 2018-03-01 02:58:27.333Z |

Figure 3 - The same data, view different views

Figure 3 shows an example of viewing the data your way. Different mission control disciplines need to see different representations of the same data. Open MCT provides the capability to show different views of the same data object.

Operations Unconstrained by Domain Walls

Mission operations is traditionally organized by discipline. Systems controllers view system data, mission planners see planning data, flight dynamics teams see trajectory data... and the list goes on. Typically we design software by discipline. While other disciplines may have cross-domain data access, that access is usually limited. The combination of all your data in one place, composition and view it your way, gives operations teams the capability to design operations without the software-induced walls and technology constraints that limit operational design options.

Why Open Source

Collaboration

Before Open MCT, we had MCT. Our closed software had many barriers to collaboration. We could not distribute the code to potential collaborators without a signed agreement. This often took months to obtain, by which time there was little interest left in collaborating.

Open source eliminates the sense of ownership that impedes collaboration. The Apache 2 license provides flexibility in use and GitHub, with its fork and pull request model it provides both flexibility and a model for collaboration.

Collaborations Within NASA

Open MCT is developed at NASA's Ames Research Center in Silicon Valley, in collaboration with the Jet Propulsion Laboratory. Each deployment has a different set of plug-ins for customization and adaptation to specific environments and mission requirements.

JPL Missions

At JPL the Open MCT deployment is called VISTA (Visualization for Telemetry Analysis). It is operational on the Jason-3 Mission and deployed in the testbed for the Mars 2020 rover mission.

The Resource Prospector Mission

Resource Prospector (RP) is a Lunar Rover Mission planned to search for volatiles at a lunar pole in 2022. Open MCT will be used at four NASA centers to monitor the mission.

Opening the gates - new categories of users

Open Source has provided new categories of users, both inside and outside the space domain. The Applied Physics Lab (APL) at Johns Hopkins University is evaluating Open MCT for use in future missions. The APL example is instructive on the key factors that make open source an enabler for collaboration. The open source model would allow APL to take Open MCT, use it as they need without outside constraints imposed, and add contributions back into the community.

From Goddard Space Flight Center, the IceSat 2 mission is planning to use Open MCT for remote telemetry monitoring.

The Cloud Turbine Project at NASA Armstrong has customized Open MCT to provide data through standard file services such as Dropbox and Google Drive.

Users Outside the Space Community

Many of the challenges in mission operations that Open MCT addresses are applicable in a broad range of fields. As such, the project has seen interest from groups in a diverse range of areas. The Open MCT team are currently aware of 12 projects from outside of the space domain that have used Open MCT experimentally or have considered it for applications that include:

- Server and network infrastructure monitoring
- Monitoring of autonomous boats
- Monitoring of Internet of Things devices for agricultural applications in Africa
- A smart room dashboard project

- Software defined radio
- An IOT dashboard for a hackspace
- Following a mission in Kerbal Space Program

As of writing this paper the team are not aware regular ongoing usage of Open MCT in a production environment among these non-space domain users.

As well as those who use Open MCT for research or commercial purposes there are a class of users who develop projects based on Open MCT for fun. One contributor created a plugin to integrate Open MCT with Kerbal Space Program - a popular game that simulators space missions. As a consequence, users of Open MCT are able to follow a mission in Kerbal Space Program using real mission control software from NASA

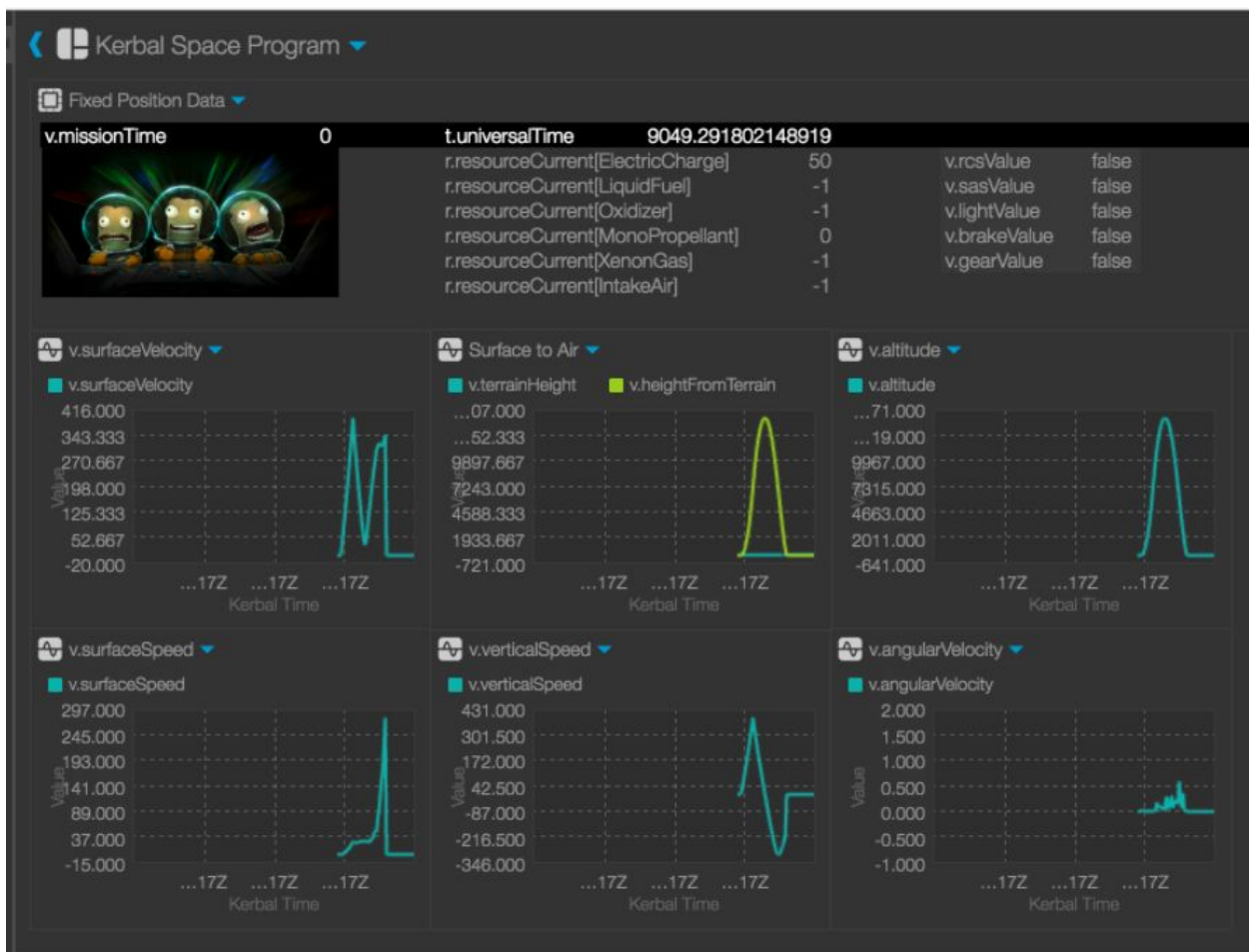


Figure X - Open MCT displaying telemetry data from Kerbal Space Program. Credit David Hudson

In another case, a hackspace in Berlin called c-base has built a monitoring dashboard on Open MCT that is used to display telemetry from a number of Internet of Things sensors measuring things like temperature, humidity, motion (<https://bergie.iki.fi/blog/nasa-openmct-iot-dashboard/>).

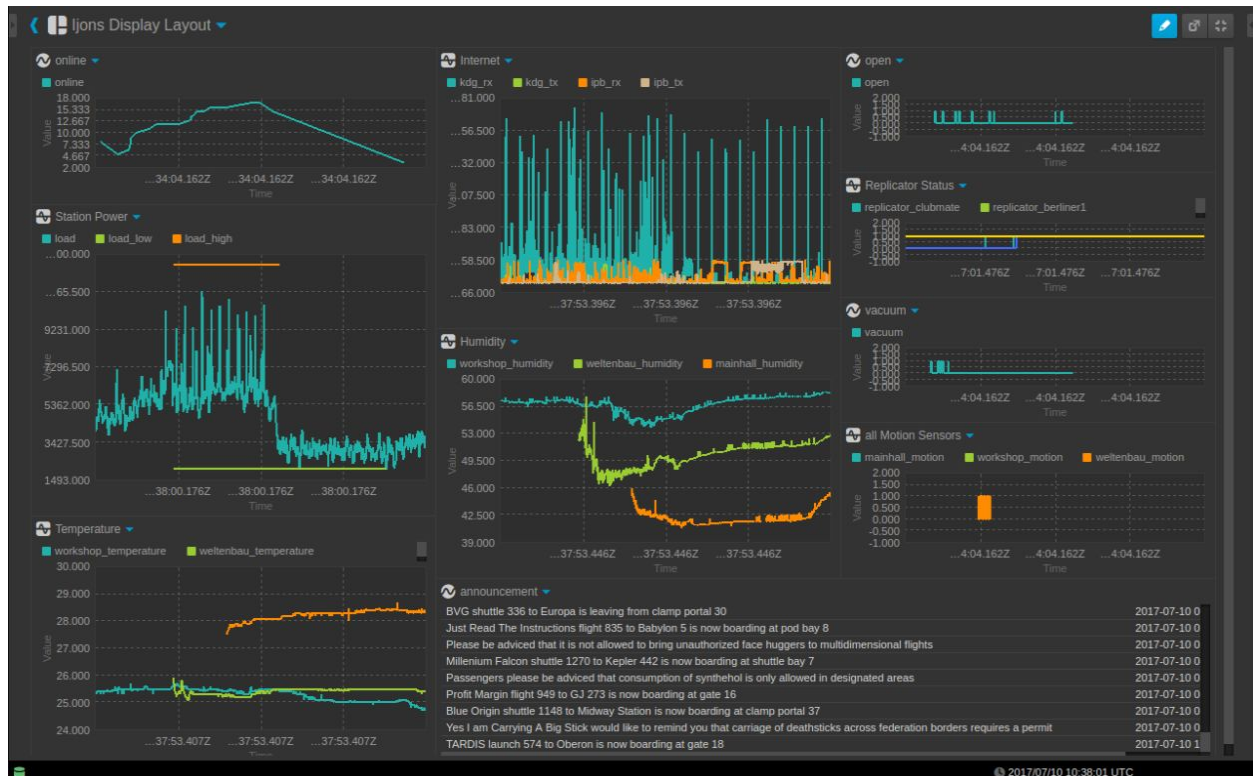


Figure X - Open MCT in use as a monitoring dashboard at the c-base hackspace in Berlin. Credit Henri Bergius (<https://bergie.iki.fi/blog/nasa-openmct-iot-dashboard/>).

Community

Statistics

The Open MCT team track usage of, and interest in the software. The project has grown from being completely unknown, to having been visited by over 140,000 people. The Open MCT project monitors site traffic to the project website and live demo at <https://nasa.github.io/openmct>, and GitHub.com provides some limited statistics concerning usage of the source code repository. In the case of GitHub.com, statistics on visitors to the site and downloads of the source code (via “clones”) are provided for the prior two weeks. These are reported below.

GitHub Repository Statistics

<https://github.com/nasa/openmct>

| June 2nd, 2015 - March 1, 2018 | |
|---|------|
| Stars Users who have favorited the repository | 4925 |
| Forks Users who have created their own repository based on Open MCT | 512 |
| Watches Users who monitor activity on the Open MCT repository and receive automatic notifications. | 224 |
| Signed Contributor License Agreements These agreements define the terms under which intellectual property is contributed to the Open MCT project, and are a requirement for contributing. | 51 |
| Code Contributions Number of pull requests that include changes or additions to code or build tooling. These include bug fixes and new features. Does not pull requests that are purely documentation or website changes. | 34 |
| Feb 16, 2018 - March 1, 2018 | |
| Clones Number of unique instances of the repository being "cloned". That is, the source code repository being downloaded. | 96 |
| Visitors Number of unique visits to the Open MCT source code repository during this period | 579 |

Open MCT Website and Live Demo Statistics

<https://nasa.github.io/openmct>

| May 6, 2016 - March 1, 2018 | |
|---|-----------|
| Page Views All requests for website pages, including repeat requests from the same users. | 1,214,368 |

| | |
|---|---------|
| Unique Visitors | 148,380 |
| Typical Daily Views The average page visits on a typical day. This excludes the period from May 20th 2016 to September 30th 2016 when the site experienced unusually high traffic due to being featured on the front page of Reddit. | 963 |
| Typical Daily Users The average number of people visiting the site on a typical day. This excludes the period from May 20th 2016 to September 30th 2016 when the site experienced unusually high traffic due to being featured on the front page of Reddit. | 104 |

The project has also received a number of non-code changes to documentation such as spelling and grammatical changes which are not accounted for here. Also not included in the above statistics are contributions made in the form of plugins, which are hosted in separate personal repositories and for which statistics are not available to us.

Contributors

Contributors to Open MCT broadly fall into two categories. First there are people who are using Open MCT in their own projects, and who develop new plugins, augment existing functionality, or fix bugs and feed them back into the codebase as pull requests. As an example of a bug being fixed by a user of Open MCT, there was a case where a change in one of the project's external dependencies caused our compilation process to fail. A developer using Open MCT at another NASA center caught and fixed this issue before the development team did. This was a major build issue that caused the build process to fail for new users, but did not immediately affect existing users. Because Open MCT has external users it was detected quickly, and because it's open source the same user was able to fix it themselves, and contribute the fix back to the community. We have also had improvements to our tutorials contributed by a user who was building a smart home dashboard using Open MCT, and a video produced by a Cubesat team in South Africa to help users get up and running with Open MCT.

Then there are the enthusiast contributors who volunteer their time to fix bugs or implement new features. Often these developers work on issues identified by the Open MCT team as "help-wanted" tasks.

The benefits to users of contributing to the Open MCT project depend on the motivations of the contributor. For users who are building their own project on Open MCT there may be a time savings in being able to fix their own issues. Likewise with new features, users who are able to do the coding themselves will likely get the results they want more quickly by implementing it

themselves than by waiting until the core team's development priorities align with their own, if this ever occurs. For enthusiasts of space exploration and NASA, Open MCT provides the opportunity to write code for a project that will be used to support NASA missions. As a publicly accessible site, contributors can reference their contributions from resumes and code portfolios, which can provide tangible real-world experience that is useful particularly to developers in the early stages of their career.

On occasion, enthusiast contributors have also suggested new features based on notional use cases that they have identified themselves. An example of this is a recent request by an enthusiast contributor identifying the potential need for a way of visualizing spacecraft tracking on a 3D globe. The contributor created a new issue in our GitHub repository outlining the perceived need, which was then seconded by a user with a real-world use case. The original contributor has since begun the task of implementing this visualization, with some guidance from us. Here we see an example of a community member anticipating a genuine user need, and implementing it themselves, with oversight by us. The priorities of the internal development team are dictated by the missions we are funded to support, so we would not have had the capacity to support this user's need immediately.

Supporting the Community

Aside from institutional requirements such as export control that are necessary to release open source software to the public in an organization like NASA, the time and cost involved in making source code publicly available is fairly small. In the simplest case code can be uploaded to a public repository such as GitHub in a matter of minutes. There is benefit just in getting this far. Making code publicly available means people can use the software, and the developers can potentially receive code contributions and feedback. The aim of the Open MCT project aims to go beyond that, and to foster a vibrant community of users and developers collaborating and feeding the fruits of their labors back to the wider community. Our belief is that users and developers are more likely to adopt software that is visibly and actively under development, is well documented, simple to setup and run, and has a responsive development team who are actively engaged with the community.

Fostering an Active Community

It has been the aim of the Open MCT project from the outset to try and maximize the benefits of the open source process by fostering an active and supportive community of users and developers. We aim to do this by:

- Maintaining an active repository, with multiple updates per day
- Being responsive to comments and questions from the community
- Providing clear and comprehensive documentation on developing for Open MCT
- Providing a simple and powerful Application Programming Interface

- Providing a supportive environment for developers with constructive feedback as part of our code review process
- Providing tutorials that walk developers through the process of extending Open MCT with a focus on common use-cases.
- Providing “help-wanted” issues that cater to a range of abilities, from simple bug fixes through to complex pieces of new functionality. A good “help-wanted” issue is one that provides a level of detail and contextual information appropriate to the complexity of the task. Tasks aimed at beginners might provide a little more information on the basics and provide links to examples, relevant sections of documentation, and pseudo-code; whereas a task intended for a developer who is more experienced with Open MCT can assume more pre-existing knowledge.

Investing Time in Open Source

This requires a moderate investment of time. On the Open MCT project we estimate that our team in total have invested approximately 8 hours per week on tasks such as:

- Discussions with potential users of Open MCT
- Providing technical support to users of Open MCT
- Providing technical guidance to contributors to the project
- Performing reviews of code contributions
- Answering questions from users and developers of Open MCT
- Writing a documented Application Programming Interface (API) aimed at developers who wish to extend the capabilities of Open MCT, or customize it for their own needs.
- Building and maintaining a website that hosts information about the project, technical documentation aimed at integrators and developers, a development roadmap.

The benefits of doing this have to be assessed and weighed against the investment of time required. Many engagements with potential users of the software do not go anywhere, and it is costly to spend team-members’ time on these sorts of exploratory discussions. On the other hand, even where engagements with potential users do not pan out, the team learns more about the users of Open MCT and their functional needs, challenges that they faced working with Open MCT, and we often benefit from an exchange of ideas. This is useful feedback that helps us to improve Open MCT.

Web Presence

The Open MCT project maintains a web presence across several different sites and media. Our informational site at <https://nasa.github.io/openmct> provides general information about the project, guides to getting started using Open MCT and to making code contributions, as well as technical documentation for developers. The site also includes a public roadmap giving an overview of our development focus over the coming months for new user features and API capabilities.

The GitHub repository at <https://www.github.com/nasa/openmct> is our code repository, and also our team's primary collaboration tool for development. New features and bugs are tracked as issues, and code reviews are carried out via Pull Requests. For more information on how the Open MCT project uses GitHub features as part of our overall software process, please see the section on Software Development Process. Our GitHub repository is also the primary means of interacting with the Open MCT team. Users and developers can raise questions, file bug reports, etc. as issues in GitHub.

Finally, we have made available an online demo and guided tour of Open MCT at <https://openmct-demo.herokuapp.com>. This demo guides the visitor through some of the concepts and features of Open MCT, as well as some of the user interactions involved in creating new displays. The online demo uses real weather data collected by the Curiosity rover to demonstrate the historical telemetry analysis capabilities of Open MCT, and also includes a display demonstrating the use of Open MCT for real-time mission operations which uses mocked telemetry data streaming to the client.

The Developer Experience

Technology

Open MCT is a browser-based application developed primarily in JavaScript. There were several motivations for using web technologies to build Open MCT, including a desire to support distributed operations and mobile access, however one of the factors considered was the popularity of web technologies amongst the wider developer community. JavaScript is one of the the most popular programming languages used today, and the most popular on open source site GitHub.com (<https://octoverse.github.com/>). This increases the pool of potential open source developers who might contribute to Open MCT.

We have also made efforts with our Application Programming Interface (API) to simplify the developer experience, and lower barriers to entry. In an earlier incarnation Open MCT required users to know or learn AngularJS - a JavaScript library - in order to extend Open MCT. For example, creating a new view required knowing Angular concepts such as controllers, scope, interacting with the event model (in particular interactions with the 'digest' cycle that synchronizes the data model with the view), and other arcanities of the Angular framework. It also used an extension model based on declarative configuration files that developer feedback revealed was a source of confusion. Significant effort has been taken to build a new API that exposes no additional dependencies on developers using Open MCT. It requires only a knowledge of JavaScript, and can be used with any framework or library that the developer prefers. We also moved toward an imperative, rather than declarative model so that rather than defining configuration files for new plugins, a developer makes calls to JavaScript functions. While also benefiting the development team, this is an example of development effort that has been motivated primarily to improve the experience of open source developers.

Tutorials

The Open MCT project maintains some tutorials which are aimed at introducing developers to the concepts and API involved in customizing Open MCT for their own projects. Based on feedback from users, the team identified a number of barriers to getting up and running with Open MCT. To this end we have implemented tutorials that focused on the most common initial use-cases for Open MCT encountered during the process of supporting projects based on Open MCT. The tutorials themselves are hosted in a public GitHub repository (<https://github.com/nasa/openmct-tutorial>) meaning that the public can contribute their own suggestions, fixes, or even new tutorials for Open MCT.

Processes

Submission and Code Review

In order to control the quality and security of code contributed to Open MCT, the project has a well defined process for accepting contributions from the open source community. This is detailed in our guide to Contributing to Open MCT hosted on our site (<https://nasa.github.io/openmct/contributing/>). Although it is open source software and as such anybody can download and modify the code, only members of the Open MCT team of NASA civil servants and contractors have the ability to merge changes into the canonical “master” branch of the repository. Even then, the Open MCT team employ an internal process to review all code changes going into this “master” branch, even by team members. Our internal development process is detailed separately in the section titled “Software Development Process”

Every contributor to the project outside of the NASA organization is required to sign a Contributor License Agreement (CLA), which defines the terms under which intellectual property is contributed to the Open MCT project.

Open MCT is made available through GitHub, which hosts code repositories using the “Git” version control system. GitHub allows users to easily fork or clone a code repository, allowing for unconstrained use of the codebase (within the terms of the license agreement). In order to make a contribution, a developer will first clone the Open MCT git repository, which makes a copy of the source code on their local machine. The user is then able to make changes to the code, test them locally, and then file a “Pull Request” to merge the code into the main Open MCT repository. A pull request is a mechanism by which code changes are highlighted, and comments can be made on the proposed changes. The Open MCT team use this mechanism to review and vet any proposed changes, and provide feedback to the contributor. Changes are reviewed for:

- Code quality - All contributions to Open MCT must meet minimum code quality standards, which are detailed in the coding standard guidelines hosted in the Open MCT

repository

(<https://github.com/nasa/openmct/blob/master/CONTRIBUTING.md#standards>).

- Applicability - Does the code appear to do address the issue?
- Tests - Does the code have sufficient test coverage through appropriate unit tests. We maintain a global minimum of 90% code coverage, enforced by a step in the build process that will fail if test coverage calls below this figure.
- Security - All code submitted through the open source process is reviewed by the team for vulnerabilities prior to merging. As well as code submissions, we consider the security risk introduced by any new dependencies, such as third party libraries.

To help submitters and reviewers, we require a checklist to be included with each code submission, and a separate checklist to be included by the reviewer once the submission has been reviewed.

To enforce standards the Open MCT project employs automation to scan submitted code for adherence with best practices, code standards, and test coverage. The team use CircleCI, a free build automation tool, to execute a full build of the codebase, run all unit tests, and generate a report on test coverage. As part of this process, CheckStyle and JSLint are also executed to scan the code for adherence to best practices. If any of these steps fail, the build will fail and GitHub will block merging of the code.

An Open Software Development Process

The Open MCT development team use GitHub.com as our issue tracker and code repository, and as a collaboration tool. New features and bugs in our open source code are tracked as issues in GitHub, and most discussion by the team is carried out in comments against the associated issue. The team will often use pull requests for work in progress in order to get consensus from the group on an approach being taken in code. This is particularly pertinent for public facing code such as our public API.

The team work in sprints of three weeks duration. A roadmap is maintained separately, which outlines strategic goals for the project over the coming year. A public version of this is made available via our website (<https://nasa.github.io/openmct/about-open-mct/>). The roadmap guides the planning for each sprint, and for every development task in the sprint an associated issue is created on GitHub.com. Once a code submission is ready for review, the same process is followed for an internal submission as for an external code submission, with the code being reviewed for best practices, and automated tools checking code coverage and adherence to standards.

There are several advantages to doing most of our development in the open like this. For one it means that we maintain a repository that is visibly active, with multiple updates per day. For someone visiting our repository for the first time this shows that the project is in active

development. Carrying out discussions and code reviews in a public forum also invites oversight from the public, and ensures that there are more eyes on the code. The public can participate in discussions, highlight issues in the code, suggest improvements to our code, build tools and processes.

Lessons Learned

- Need to have the time to dedicate to supporting Open Source. Code reviews, technical support and guidance all require an investment of time.
- Code contributions have generally been of a quality that requires only minimal changes to be mergeable. This may be somewhat self-selecting, as developers motivated enough to volunteer their time to contribute to an open source project are presumably. Where changes are required, these are usually undertaken by the contributing developer, with the guidance of the Open MCT team.
- Once a project is open-source and online you no longer control the profile of the project. The Open MCT team have done very little active promotion of the project, but it has been promoted by the open source community. The project was submitted to Reddit by a member of the public, and featured on the front page for a day or so. As a consequence, visits to our website peaked at over 20,000 per day for several days, and we observed a noticeable increase in interest from users and developers alike.
- The number of contributors who make substantial code contributions has been small, the ones who make substantial contributions tend to make multiple contributions.
- Help contributors to help you. The more documentation and tutorials provided, the less support is necessary to get users and developers up and running.
- There is still a non-trivial amount of coding necessary to display data in Open MCT, and the public API is currently limited in its functionality, forcing users to make use of our internal API. The result is that to build anything substantial in Open MCT it is necessary to learn two distinct APIs, with separate documentation, and very differing styles.

Where do we go from here

Mission operations software as open source is still in an experimental phase. To date, we consider our open source experience to be a net plus, with the involvement of the community making the software higher quality, and broadening the user and developer communities significantly. We will continue to support the open source community and the community of mission users, driving towards the goal of open mission operations systems software, from which we can all build on each others work.

- Further improvements to the developer experience (extending the API, additional tutorials, improving documentation)
- Ecosystem of plugins with the goal that users can build a mission control system based on Open MCT without having to write code.