SSC07-XII-3

Open Source Software for Small Satellites

Paul Wooster
DevelopSpace Initiative
paul@developspace.net

David Boswell
Mozdev Community Organization
davidwboswell@yahoo.com

Patrick Stakem
OpenFlightLinux.org
pstakem@openflightlinux.org

Jessy Cowan-Sharp
NASA Ames Research Center
jessy.cowansharp@gmail.com

ABSTRACT

The open source approach to software development has radically changed the way in which many software applications have been created and improved over time. This paper examines both the current state and potential future directions for free and open source software in the space realm, with a particular emphasis towards developers of small satellites. The significant benefits of open source software for small satellites are described, including how open source software can increase reliability, decrease cost, and increase flexibility as compared to closed source software or software developed exclusively in-house. Software currently under development by the open source space software community is examined. The potential for increasing the openness of small satellite flight software is described and discussion is included regarding how members of the small satellite community can contribute to such activities.

INTRODUCTION

Free and open source software is software licensed in such a manner as to allow its use, modification, and redistribution to others under like terms. Software developed and released in an open source manner is seeing increased popularity, ranging from much of the software infrastructure underlying the Internet to operating systems and desktop applications. Today, general purpose as well as "space-oriented" free and open source software is already assisting many space development and support activities.[1,2]

BENEFITS OF OPEN SOURCE SOFTWARE

Using and developing free and open source software offers a number of distinct advantages as compared to proprietary/closed source code. Integrating existing source code into software offers decreased cost and faster availability by eliminating the need for substantial development and/or procurement of proprietary software. Developing software in an open source manner leads to significantly greater visibility, feedback, and useful contributions towards the functionality of the software than would otherwise exist – this tends to increase the quality of the software (in terms of both features and reliability) while decreasing the investment that must be made by the organization developing the software. Many of these benefits come about as a result of a wide variety of people using and in so doing, testing the software. Releasing software in an open source manner can also increase interest in the project and provide an educational opportunity for those examining the software.

A number of characteristics that frequently accompany small satellite projects make them well suited to the use of open source software. A tight budget leads to a desire to limit licensing fees. The need for new, innovative solutions creates a desire to be able to add functionality above that of existing software. A rapid schedule leads to a desire to have a code base to build-upon. Having massively parallel, distributed peer-review of software can also lead to increased probability of mission success.

CURRENT OPEN SOURCE SOFTWARE FOR SMALL SATELLITE TEAMS

A variety of open source software exists that is applicable towards teams developing and operating small satellites. This runs the gamut from software of relevance to the conceptual and preliminary design of a satellite to the software used in ground and flight systems. A listing of open source engineering software, with a particular emphasis on software of relevance to space systems development, is maintained online as part of the DevelopSpace Open Source Engineering Tools project.[3] We highlight two such tools here, followed by a more detailed description of an effort to develop a Linux distribution tailored to the needs of the flight software world. References 1, 2 and 4 also provide overviews of open source software in the space arena.

Java Astrodynamics Toolkit

The Java Astrodynamics Toolkit (JAT) is a cross-platform library of software components centered around astrodynamics, enabling mission design and other simulations related to spacecraft development.[5] Users can create programs in either Java or Matlab based upon the JAT components. Reference 6 provides a comparison of the performance of JAT relative to the commercial, closed source alternatives STK and FreeFlyer. While not offering a complete stand-alone environment such as these alternatives, JAT provides a great deal of flexibility to a mission designer by allowing the tailoring of the simulation to the needs of a mission. As free, open source software released under the General Public License (GPL), users are free to examine and modify the code to meet their needs, as well as distribute the software (in either original or modified form) to others without incurring licensing fees.

Open-SESSAME

The Open-Source, Extensible Spacecraft Simulation And Modeling Environment (Open-SESSAME) is a C++ software package aimed primarily at the simulation of spacecraft dynamics and control.[7-9] The Virginia Tech Space Systems Simulation Laboratory originally developed Open-SESSAME, and also provides the lower-level code running on their Distributed Spacecraft Attitude Control System Simulator as free, open source software.[10] Both of these packages are also released under the GPL.

FlightLinux

Between 1999 and 2002, NASA's Goddard Space Flight Center's Advanced Information System Technology Office funded the FlightLinux Project, an early effort to assess the technological readiness of an open source operating system for onboard spacecraft use. Many lessons were learned, and the advantages of this approach were delineated in the Project's final report.[11] However, issues emerged that hindered the application of this approach, whereby a worldwide community of developers could be brought together to address and contribute to very interesting and unique space-related projects. The FlightLinux team believes they have a means to overcome these barriers, where the first step is the creation of a community and an open source distribution of an operating system, OpenFlightLinux.[12]

Building on NASA’s investment in FlightLinux, OpenFlightLinux provides customized operating system and applications builds. OpenFlightLinux is built from source code. It can be customized as a development platform, a server, an embedded real-time system, an appliance, avionics, a cluster, even a desktop. Based on aerospace industry best practices in maintainability, reliability, and fault-tolerance, OpenFlightLinux provides an ideal platform for mission critical applications. The system build includes those parts necessary for the application, and does not include anything else. The OpenFlightLinux team is developing and deploying open source tools to facilitate custom system builds and address dependency problems. Real-time, low latency embedded systems are also addressed.

OpenFlightLinux is free and open source software. One can download pre-configured loads, or the source code. Applications can be configured as required. A simple x86 distribution, and a PowerPC version, targeted to the RAD-750 space-based processor, have been completed.

A version of OpenFlightLinux for desktop use allows for the migration of applications from the ground to the space environment. The file system onboard the spacecraft may be network-attached to the ground system computer via a network supporting TCP/IP, which has been demonstrated on-orbit. OpenFlightLinux is also designed to allow scalability by means of clustering.

The advantage of this operating system and development approach to the small satellite developer and operator is clear. In resource constrained, yet mission critical applications, having project-wide visibility and control of all of the software at the source code level can be essential. Having a process for distributed, collaborative development, test, and operations is enabling. Having the same software platform across the spectrum of applications is cost-effective.

Interestingly, OpenFlightLinux is not restricted to esoteric space projects. The team is actively working on Medical-FlightLinux, a version that implements NASA-developed planetary imaging techniques for medical applications, as well as Flight-Linux-FPGA, which allows configurable hardware as well as software. Using Flight-Linux-FPGA is opening the door to the concept of rehosting traditional operating system functions into hardware, while retaining flexibility. Participation in OpenFlightLinux is welcome at all levels.

BUILDING AN OPEN SOURCE SPACE SOFTWARE COMMUNITY

While much potential exists for developing open source software of relevance to small satellite developers, to maximally leverage these efforts a community needs to be encouraged to grow around these projects. A community needs a website that contains both development and communication features and that is run by an active community manager. When the appropriate tools and infrastructure are in place, every visitor becomes a potential volunteer, tester, contributor and user.

An active community does not just spontaneously happen. 'An open license does not guarantee that hordes of active developers will suddenly volunteer their time to your project.'[13] In the introduction to *Producing Open Source Software*, Karl Fogel goes on to say that 'Management in an open source project isn't always very visible, but in the successful projects, it's usually happening behind the scenes in some form or another.'[13] A successful community provides the interactions that are a significant part of open source software's value.

The white paper *Return on Community: Proving the Value of Online Communities in Business* also states that successful communities require active management.[14] Much can be learned from successful open source projects such as Apache, Mozilla and Linux, but communities that have failed such as ostdev.net, possenet.org, e-speak.net and possl.org have lessons to teach as well. These now abandoned projects were created by organizations that built open source sites, but failed to build open source communities. 'In order to build a development community, you need to attract people, interest them in what you're doing, and keep them happy about the amount of work they're doing.' In *The Cathedral and The Bazaar*, Eric Raymond explains that communities develop around a clear mission and effective guidance.[15] A community development plan and a community manager can encourage the growth of a community and monitor its progress to make sure the goals of the program are being met.

Setting up the legal and institutional infrastructure for an open source strategy can often demand all of the attention of an organization so that there is no clear plan for how to use that infrastructure to derive value. Investing additional resources into fostering a community can also seem like a 'nice to have' option and not a requirement. Without an effective plan for building a community however it is possible to have the components of an open source strategy without gaining any of the benefits.

CosmosCode and DevelopSpace

CosmosCode is a NASA project based out of the Ames Research Center which aims to foster the development of open source software related to space and to build-up a community around such software.[16] As part of this effort CosmosCode is developing an open source space software project hosting infrastructure. The project is also working to identify means by which NASA organizations can more fully participate in and make contributions towards open source space software development, and to encourage such interactions. This will go beyond the simple release of NASA software under an open source license[17], as is currently done through websites such as those in references 18 and 19. The DevelopSpace Initiative is a private, non-profit organization aiming to apply open source concepts towards all aspects of space systems development, including not only in the software domain but also hardware and overall system design, as well as associated reference materials.[20] The overall goal of DevelopSpace is to build-up the technical foundations for human space activities, including open sharing of relevant technical resources and fostering related technical activities and innovations. DevelopSpace aims to enable a wide variety of individuals and groups to participate in the exploration, development, and utilization of space. The DevelopSpace.net website offers project hosting and other services for those interested in developing space systems in an open manner.

Small satellite teams are encouraged to engage with both CosmosCode and DevelopSpace in order to receive additional design feedback and review, foster a community around their projects, and gain contributions to their on-going activities. Drawing on the resources of the open source world can help small satellite teams succeed within an environment of constrained cost and schedule.

SUMMARY

Open source software offers the potential for significant benefits in the conception, development, and operation of small satellites. The benefits of open source software in this context include decreasing cost, decreasing development time, and increasing reliability. A variety of open source software exists of relevance to the small satellite community, ranging from general purpose engineering, networking, and analysis software to software developed specifically for space-related purposes. A number of websites to foster further open source space software development are being developed, including NASA's CosmosCode project, focused on space-related software, and the private DevelopSpace Initiative, for open and collaborative space engineering more generally.

REFERENCES

1. Boswell, D.W., "The Software of Space Exploration", http://www.onlamp.com/pub/a/onlamp/2006/03/30/software-of-space-exploration.html

2. Byrne, D.J., "Open-Source Software and Its Role in Space Exploration", http://www.cio.com/article/111950

3. DevelopSpace Open Source Engineering Tools project website, http://www.developspace.net/wiki/Open_ Source_Engineering_Tools

4. Berndt, J.S., "A Survey of Selected Open Source and Freely Available Software for Engineering and Science, and for Education, Home, and Work," AIAA Houston Horizons, Volume 31, Number 3, January-February, 2006

5. Java Astrodynamics Toolkit (JAT) website, http://jat.sourceforge.net/

6. Gaylor, D.E., R.C. Page III, and K.V. Bradley, "Testing of the Java Astrodynamics Toolkit Propagator," AIAA-2006-6754, AIAA/AAS Astrodynamics Specialist Conference and Exhibit, 21 - 24 August 2006, Keystone, Colorado

7. Open-Source, Extensible Spacecraft Simulation And Modeling Environment (Open-SESSAME) project website, http://spacecraft.sourceforge.net/

8. Turner, A.J., "An Open-Source, Extensible Spacecraft Simulation And Modeling Environment Framework", Master's Thesis, Virginia Polytechnic Institute and State University, September, 2003

9. Turner, A.J., and C.D. Hall, "An Open-Source, Extensible Spacecraft Simulation And Modeling Environment Framework," AAS 03-501, AAS/AIAA Astrodynamics Specialist Conference, Big Sky, Montana, August 3-7, 2003

10. Distributed Spacecraft Attitude Control System Simulator (DSACSS) Framework project website, http://dsacss.sourceforge.net/

11. NASA Goddard Space Flight Center FlightLinux website, http://flightlinux.gsfc.nasa.gov/

12. Open FlightLinux project website, http://www.openflightlinux.org

13. Fogel, K., Producing Open Source Software: How to Run a Successful Free Software Project, http://producingoss.com/, 2005

14. Warms, A., J. Cothrel, and T. Underberg, "Return on Community: Proving the Value of Online Communities in Business," http://www.interaction-ivrea.it/courses/2002-03/conncomms/docs/wp-returnoncommunity.pdf

15. Raymond, E.S., "The Cathedral and the Bazaar," 2002, http://www.catb.org/~esr/writings/cathedral-bazaar/

16. NASA CosmosCode project website, http://www.cosmoscode.org/

17. Moran, P.J., "Developing An Open Source Option for NASA Software," http://www.nas.nasa.gov/News/Techreports/2003/PDF/nas-03-009.pdf

18. NASA Goddard Space Flight Center Open Source Software, http://opensource.gsfc.nasa.gov/

19. NASA Ames Research Center Open Source Software, http://opensource.arc.nasa.gov/

20. DevelopSpace Initiative project hosting and reference website, http://www.developspace.net