

# BUSMASTER – An Open Source Tool

Lorenz Tobias, ETAS GmbH

Gopinath Mujumdar, Amitesh Bharti, Robert Bosch Engineering and Business Solutions Ltd.

In August, ETAS and Robert Bosch Engineering and Business Solutions (RBEI) jointly published BUSMASTER, a free open source PC software for the design, monitoring, analysis, and simulation of CAN bus systems. The software can be downloaded from <http://rbei-etas.github.com/busmaster/>.

The current BUSMASTER version is based on the preceding software tool CANvas, conceptualized, designed and developed by RBEI. It offers import filters for network description files and simulation programs compliant with standard industry formats. For CAN connections, hardware from different vendors is supported.

The BUSMASTER software project, sponsored by RBEI and ETAS, is open to contributions from research and industry. The software can be developed and managed with free software tools. Thanks to the modular architecture, third-party software developers can easily add new functions to the software. The license also permits the provision of proprietary add-ons, which can be dynamically linked to the open source core.

The openness of the project managed by the sponsors provides for flexible modification and extensions regarding bus systems, protocols, and hardware interfaces. In addition, it will facilitate short cycles in the solution's onward development.

ETAS and RBEI (Robert Bosch Engineering and Business Solutions Limited, India) recently published a PC-based rest bus analysis and simulation tool (see Figure 1) on the open source platform GitHub. The major motivation for this is to launch and prototype a new business model in a market with high entry barriers. The project was carried out in two phases: the preparation phase and the operation phase. The preparation phase included the investigation of many Open Source Software (OSS) aspects, e.g., source quality, business models, open source infrastructures, legal implications, customer communication, community buildup, marketing, publishing processes, and organizational aspects. The operation phase starts with the going live on GitHub and includes for example community management, engineering services, and support.

This document describes the challenges of legal implications (copyright, license, and contribution), community buildup, and business model, which were analyzed and solved during the project preparation phase. The project was jointly carried out by ETAS and RBEI. The OSS was

conceived, developed, and prepared by RBEI.

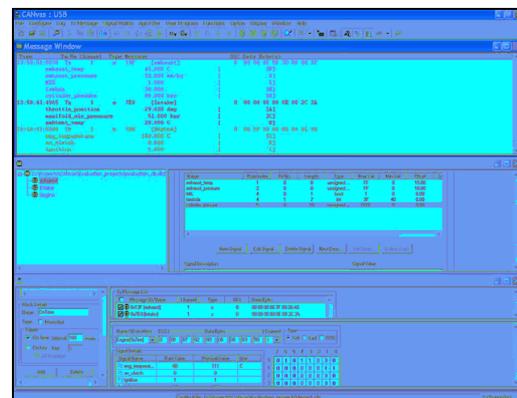


Figure 1: Tool screenshot

## Copyright

The legal introduction to Open Source Software starts with the copyright. Creators of original work have exclusive rights on it, known as copyrights. Copyright holders make use of two common legal constructs to extend their exclusive rights to others: licenses and contributor agreements. A license grants permissions for the usage or redistribution of the work. Although this is not possible in every jurisdiction, a contributor agreement allows the contributor to transfer or assign

his exclusive rights. The license and the contributor agreement are explained in the following two chapters.

### **License**

The Open Source Initiative (OSI) [1] is a community-recognized body for reviewing and approving of licenses conformant to their Open Source Definition (OSD) [2]. The OSD defines Open Source as not just access to the source code, but that the distribution terms of the Open Source product (software, hardware, and digital content) must comply with 10 criteria: Free redistribution, source code, derived works, integrity of author's source code, no discrimination against persons or groups, no discrimination against fields of endeavor, distribution of a license, the license must not be specific to a product, the license must not restrict other software, and the license must be technology-neutral.

An Open Source license with a copyleft effect preserves the same rights of the original work in derived versions of the work. Such licenses roughly fall into one of three categories: Strong copyleft licenses, weak copyleft licenses, and permissive licenses without copyleft effect. The license itself defines what derived works are. Usually they are created by copying parts of the original source code, by statically linking the object code, and sometimes already by dynamically linking the object code. This shows that even a loose coupling may "infect" other programs with the Open Source license. For that reason copyleft licenses are sometimes called "viral" licenses.

If a license infringement happens, the entitlement can be the removal, the injunctive relief, the compensation of damage, or the disclosure of the work. In the first two cases the claim is successful even when the violator unknowingly infringed the license. In the case of damage compensation a nomination of co-owners in the complaint is necessary.

There are technical and organizational ways to mitigate the risk of license infringements. Nowadays there are tools available, such as Blackduck [3] or Palamida [4]. They can scan the source code for OSS components and report possible license infringements.

### **Contributor agreement**

Beside an Open Source license, a contributor agreement (CA) defines terms under which intellectual property and copyrights are contributed to the project owner or company. There are two basic transfer options. A copyright assignment agreement (CAA) is used to assign all copyrights to the project and gives a very broad license back to the contributor. The second option is a copyright license agreement (CLA), which is a very broad license from the contributor to the project. In jurisdictions where an assignment is not possible (e.g., Germany) the effect is similar to the license option. In any case a contributor agreement grants patent licenses to the project in the necessary scope to use contribution. For the project, this strongly reduces the risk of copyright and patent infringements.

Nowadays CAs are available in the form of different templates from the project Harmony Agreements [6]. This project was launched in May 2010 with the aim of harmonizing and reducing the number of different contributor agreements. To make it even more comfortable for Open Source projects, there is an agreement selector available that, upon selecting certain options, generates complete agreement contracts suitable for a project.

However, not all Open Source hosting providers allow CAs. For example, SourceForge is one of the most prominent hosting providers, but they do not allow CAs in their terms of use [7]. GitHub on the other hand places no restrictions on their OSS projects [8].

### **Choosing the right license**

After explaining licenses and contributor agreements, the question of a proper Open Source license remains. Today, many projects include "open" or "free" in their names solely for marketing reasons and with no relation to Open Source. Shared source programs often refer to "open source" because their source code is available upon registration, but their license usually prohibits its change or reuse in other projects. A real OSS project (free / libre) has to choose a license approved by the OSI. The freedom provided by these licenses is an important requirement to motivate other parties to

collaborate on the project development and to ensure future code improvements. Some projects realized that the license itself is not that important for the attractiveness of the project. That is the reason why, following several license changes, Mozilla Firefox is available today under multiple licenses: Mozilla Public License (MPL), GNU Public License (GPL) and GNU Lesser General Public License (LGPL).

Of course, the rights and obligations of a license should be studied and the implications be known before using it for a project publication. However, in general, it is beneficial to choose one of the top 20 licenses. According to blackduck [5] the GPL is the most dominant strong copyleft license used by more than 50% of projects, followed by the permissive licenses BSD/MIT/Apache, which are used by more than 20% of projects. Almost 10% of projects use the LGPL as the dominant weak copyleft license.

For the ETAS and RBEI bus analyzer tool a LGPL license was chosen. The reason lies in the business model, which shall permit an ecosystem of free or commercial add-ons. These add-ons are dynamically linked to the tool's modular plug-in architecture. A GPL would require these add-ons to use the same license and would therefore not be consistent with an ecosystem. On the other hand a permissive license would not require anyone to contribute modifications back to the originator and would also not be beneficial for market penetration of the tool.

In contrast to PC-based software, embedded software usually requires static linking between software components under different (proprietary) licenses. For such projects only a permissive license without copyleft effect can be used, such as the BSD license. The only requirement on this license is the proper attribution in the final product. This means that the product documentation must state which software components used under OSS licenses are included in the product.

### **Contributing to OSS projects**

Company employees face the problem that national employment laws or individual employee contracts mean that

any results created during the term of employment belong to the employer. There are at least three solutions to this problem.

The first solution is changing the employee contract, by adding a clause to it or providing an extension. This requires a deeper trust to the employee in choosing a third party in the expected interest of the employer to which the employee contributes his work results. This requires a high degree of trust between the employee, the employer and any third-party to whom the results will be contributed that the business interests of the employer will be respected.

The second solution is a contract between the employee and the project which clarifies the ownership rights of work results contributed to the project. This is often done in public funded projects or in standardization committees. It also allows the employee to collaborate with such projects or its partners.

The third solution is a well-defined and established process for the publication of contributions. All information leaving the employer should find its way through this process. This is very similar to the established publication process for results, e.g. conference papers, but compared to this, it must be much leaner. Small changes, like bug fixes, should only require approval by the project manager, while larger changes, such as new functionality with possible business potential, should be approved by senior management. External contributions should always be accepted if the technical quality is sufficient and if they are generally useful. This should be achieved in collaboration with the contributor in order to maintain motivation for further contributions.

### **Building an OSS community**

The attractiveness of a project depends heavily on the first impression of the software. Therefore it is necessary to lower the barriers for using and developing the software. If a tool provides all functions expected from it in an intuitive and obvious way and these functions are well

documented, e.g., by a good user manual, tutorials and examples, then the first obstacle for new users is removed. Open Source developers appreciate high quality, well documented source code, and support for cost free build environments and tool chains. Together these are the major success factors for project attractiveness from a technical perspective [9].

From a marketing point of view it is beneficial to use an OSS hosting facility that is already popular with a lot of projects and developers. This is true for SourceForge, which is more project-oriented, and also for GitHub, which is more oriented towards individual users. In the latter case, the user clones complete project repositories from other users, makes modifications and sends pull requests back to the original user. This results in several almost identical repositories among all GitHub users and is one of the reasons for the huge growth of this platform in the last couple of years.

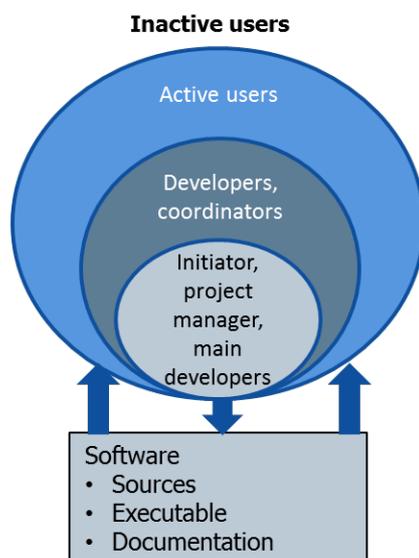


Figure 2: OSS Community

When building an OSS community, the motivation for individuals to it must be evaluated and the project adjusted accordingly. Collaborators are often found in the academic environment. They start using the OSS solution because it is cost attractive. Often they become developers as soon as they make useful adaptations for their own purposes. For them, the motivation lies in communicating with other users and developers and getting

the feeling that their contributions are welcome and quickly incorporated in the project's source code. For partner companies the major motivation is business potential.

In case of the ETAS and RBEI rest bus simulation tool, hardware manufacturers are interested in contributing just the required parts (drivers) to make the software work with their own products. The more the software supports their interfaces, the more attractive their complete solutions are on the market. This is the specific motivation for them to contribute to the project.

If the project and the number of developers contributing to it grow, a project structure becomes necessary. This usually consists of a steering committee for the long term strategic goals of the project. If necessary, smaller groups can be setup to focus on specific technical aspects, e.g. driver support, interoperability, usability, and accessibility. Usually the more a developer is recognized by his project contributions, the more influence he gets on the long term development. This process of self-organization is referred to as having "meritocratic government and rights" among all participants and nowadays is often the focus of social research.

### Business models with an OSS platform

While the majority of Open Source users are happy with what they can download free of charge from the Internet, business customers prefer buying complete solutions based on an open source product from a reliable supplier. The reason for this is to obtain long term support in answering questions or fixing issues with the product. Customer specific adaptations or integration into specific business environments, as well as training and coaching, are also frequently requested. For this kind of business model it is helpful to provide a link that redirects visitors or users of the OSS to a business solutions homepage.

Another widely-used business model is to provide premium versions or commercial add-ons in relation to the OSS. Usually these solutions contain additional functionality which is beneficial or necessary for business environments. For

example, enabling interoperability with other business solutions or support of the customer's specific hardware components are typical premium features. Such premium versions are also more extensively tested based on company standards or they could be certified when upon customer request.

In commercially driven OSS projects, an especially close cooperation with the community is necessary to know and consider their interests. Conflicting interests often lead to clones of the project where a significant number of developers left the original project and continued with the development with their abandoned part of the community. One way to prevent this is to show that the most relevant activities always remain in the original project. This has not always to do with the development itself but is also closely linked with the marketing and communication of the project. Another way to reduce the risk of clones is having trademarks on the project names. Often the project is recognized on its name. So using a trademark forces derived projects to have a different name. It then requires a large marketing effort for the derived projects to pull the community of developers and users on their side.



Figure 3: Business and Open Source web site linking

There was a two-step marketing approach to mitigate risks regarding the technical quality of the project. After publication ETAS and RBEI contacted first customers in September and October 2011 to retrieve and incorporate feedback. Beginning in November 2011 intensive marketing activities were done. This includes distributing the Open Source versions as supplement to magazines and other ETAS and RBEI hardware and software products.

### Summary

ETAS and RBEI are among the first business units within the Bosch group to publish and lead a new Open Source

project. As well as the main interest in market penetration, Open Source is elaborated as a new business model. The risks of this strategy have been evaluated, however, it remains to be seen whether the publication of a formerly commercial product under an Open Source model will lead to increased attractiveness and business opportunities. In any case, as the first such program in this application field it will be in the focus of the Open Source community and will appear in different media. From a business perspective opening an Open Source product equally touches on technical, legal, and marketing topics. The decisions and experiences made during this project will be guidance for further upcoming Open Source involvements of the Bosch group.

### References

1. Open Source Initiative, URL: <http://www.opensource.org/>, last viewed: 2011-08-08
2. The Open Source Definition (Annotated), URL: <http://www.opensource.org/osd.html>, last viewed: 2011-08-08
3. Blackduck, URL: <http://www.blackducksoftware.com/>, last viewed: 2011-08-08
4. Palamida, URL: <http://www.palamida.com/>, last viewed: 2011-08-08
5. Blackduck Open Source License Data, URL: <http://www.blackducksoftware.com/os/licenses>, last viewed: 2011-08-08
6. Harmony Agreements, URL: <http://harmonyagreements.org/>, last viewed: 2011-08-08
7. SourceForge Terms of Use, URL: [http://sourceforge.net/apps/trac/sitelegal/wiki/Terms\\_of\\_Use](http://sourceforge.net/apps/trac/sitelegal/wiki/Terms_of_Use), last viewed: 2011-08-09
8. GitHub Terms of Service, URL: <http://help.github.com/terms-of-service/>, last viewed: 2011-08-09
9. Open Source community building: a guide to getting it right, URL: <http://www.visionmobile.com/blog/2011/01/open-source-community-building-a-guide-to-getting-it-right/>, last viewed: 2011-08-10

10. Open Source Community Building,  
URL:  
[http://www.opensourcejahrbuch.de/download/jb2006/chapter\\_05/osjb2006-05-03-stuermermyrach.pdf](http://www.opensourcejahrbuch.de/download/jb2006/chapter_05/osjb2006-05-03-stuermermyrach.pdf), last  
viewed: 2011-08-10
- 

Dr. Tobias Lorenz  
ETAS GmbH  
Borsigstraße 14, 70469 Stuttgart,  
Germany  
Phone: +49 (711) 89661-385  
E-Mail: [tobias.lorenz@etas.com](mailto:tobias.lorenz@etas.com)  
Website: [www.etas.com](http://www.etas.com)

---

Gopinath Mujumdar  
Robert Bosch Engineering and Business  
Solutions Ltd.  
No.123, Industrial Layout, Hosur Road  
Koramangala, India  
Phone: +91(80)6657-4069  
E-mail:  
[Gopinath.Mujumdar@in.bosch.com](mailto:Gopinath.Mujumdar@in.bosch.com)  
Website: [www.boschindia.com/](http://www.boschindia.com/)

---

Amitesh Bharti  
Robert Bosch Engineering and Business  
Solutions Ltd.  
No.123, Industrial Layout, Hosur Road  
Koramangala, India  
Phone: +91(80)6657-5371  
E-Mail: [Amitesh.Bharti@in.bosch.com](mailto:Amitesh.Bharti@in.bosch.com)  
Website: [www.boschindia.com/](http://www.boschindia.com/)

---