Open Source Development: A suitable Method to introduce a standardized communication protocol?

Position Paper for the 1st Workshop on Open Source Software, ICSE 2001

Dipl.-Inform. Achim Spangler
Department of Bio Resources and Land Use Technology, Crop Production Engineering
Technical University Munich, Germany
spangler@tec.agrar.tu-muenchen.de

Abstract

Open Source developments like telnet and Apache are very important for an extensive and compatible use of the TCP/IP and the HTTP protocol. Both contain features which rely on a compatible implementation of complex interactions between computers of different platform types. This can be successfully achieved with the help of Open Source. Reconciliation of different interpretations of the standard, clarifications and extensions can be realised by discussion systems and some generally accepted conciliators of Open Source projects. As long as companies have a commercial interest in selling solutions based on the protocol, it makes economic sense for them to jointly support its common implementation. This paper examines the possibility to use an Open Source reference implementation of the agricultural communication protocols DIN 9684 [1] [2] and ISO 11783 to establish a capable and compatible implementation in agricultural mobile process control and data collection.

Keywords
Open Source, communication protocol, DIN 9684, ISO 11783

1 Introduction

The Open Source philosophy is embodied by a widespread community of interested people who discuss their opinions with request for comments (RFC), email lists or bulletin boards, and who use code sharing to create a capable reference implementation. This mechanisms are very important for communication networks, where the software of different computers must interact with each other smoothly.

2 Role of Open Source for communication protocols

One of the first examples of this dynamic process is the development of the TCP/IP protocol and its central applications. This extends from the "Internet Engineering Task Force, or IETF, the body that creates and governs the Internet standards," which "operates by a process that has a great deal in common with open source" [6] to applications like telnet, which are initially implemented and shared by students. The communication within the TCP/IP standardising community is performed with the help of request for comments (RFC) where a draft standard needs enough interest in the community, a detailed description and a working implementation [9]. Every reference implementation is adopted to a variety of hardware platforms, so that a compatible interaction of widespread computers is assured.
Accordingly the Apache webserver, which is used on over "60% of the world's web sites, including many of the most heavily trafficked, such as Yahoo!" [6] has the power to convince commercial software developers (like AOL) to change their product, if the Apache community detects a misinterpretation of the standard

in their commercial product [3].
Typical for open communication protocols is the fact, that the release of a new standard represents only a starting point of a constant evolution, which needs intensive discussion to reconcile different interpretations of the standard and to create clarifications and extensions. This process should be supported by reference implementations, which reflects the actual state of the standardisation, and which can guarantee compatibility on different hardware platforms.

3 Open communication protocol for mobile agricultural process control

Comparable to the protocols TCP/IP and HTTP the german standard DIN 9684 was released 1999 with a lot of details, which needed a co-ordinated evolution to get a usable working network. The focus of the protocol is to enable based on CAN a manufacturer and construction independent documentation and process automation function. Despite closed networks like within a tractor, the configuration of the network (which devices) and each connected device (attributes) isn't known during development of single devices. Corresponding to the strategies of automatic production planning systems [7] the interactions should use an abstract view on all devices and their services. Therefore an effective framework for the communication of measurement values or setpoints of process information like speed, wheelslip and working state is important.
The resulting network uses co-operation of devices with no master-slave hierarchy, where each one is responsible for its own and interacts with the others by services for documentation and control. This mechanism is comparable to CORBA with the exception that the possible services are organised in a data dictionary of process data information with a unique set of interface methods.
Complex interdependencies can arise if some devices are dependent on the services of other devices to realise their own services. This leads to the possibility of direct resource conflicts if two different devices wants to control the same resource (e.g. driving speed of the tractor). Indirect resource conflicts are also possible if different devices use different services of one device, which can't be served independently (e.g. driving speed and power-take-off of the tractor).
The avoidance and solving of such resource conflicts needs unique strategies, which must evolve corresponding to emerging problem situations. So the different agricultural devices should be programmed based on a unique well maintained commercial or Open Source program library, which offers suitable interface functions as frontend to complex background interactions.
There is no commercial library available at the moment, because the number of different hardware platforms (operating system, microprocessors and compilers) is too high and the quantities per version are too low, so that the costs are too high for a commercial interest. As a result every company create its own solution. Insufficient knowledge and unsuitable software engineering methods for the area of complex communication networks and inadequate communication between the companies for the evolution of the protocol lead to different incompatible dialects of DIN 9684. Additionally each variant implements only a small part of the whole protocol, so that the optimisation of the usability of the machines is in no acceptable relation to the additional costs.

4 Use of Open Source for DIN 9684

This hindering situation for a broad introduction of DIN 9684 could be solved with the help of Open Source, even if this philosophy is very new to this kind of industry and to this kind of communication protocol. Its strategy must take into account some important project management topics.

Initial development of Open Source software
A reference implementation can be started best comparable to projects like telnet within the research of a university, because of the neutrality of this institution. This way the developer of the software can get the generally accepted conciliator, who decide on the further progress of the project, comparable to Linus

Torvalds for LINUX. Researchers are additionally free to choose to implement the complete standard in a conformant and capable way, whereas company developers must cope with the restrictions of time and ordinary hardware platform.

Compatibility

The reference implementation must use a hardware abstraction layer to ensure the use of unique communication algorithms for all platforms. According to Apache the different versions of the hardware abstraction layer can be maintained as part of the project, to allow a jointly adoption to new hardware types.

Different interpretations of the protocol and wishes for changes in the software should be discussed in email lists. If the finding of a generally accepted decision fails, the approved project leader can settle on this topic to ensure further compatibility. Looking at LINUX, the decisions of Linus Torvalds are accepted also by companies like IBM, so that the project evolution stays unique.

Comparable to TCP/IP extensions to the original protocol can be implemented and tested in this project before they are integrated to the standard. This way a de jure standardisation, which is often controlled by economic strategies of the involved parties, can become a de facto standardisation, which is promoted by generally accepted realisations.

Quality

Important for the long term acceptance of the project is its quality, which needs extensive regression tests with a suitable simulation system. If the companies focus their commercial interest on the applications which use the project, they can be motivated to invest development time and money in a suitable test environment. They should develop jointly a test system for PC where the microcontrollers and their CAN communication can be simulated by parallel processes with process communication. Regression tests can ensure based on this the correct function of the unique communication parts, so that the testing costs for every change of the software can be reduced.

A central bug tracking system, which is also used for Mozilla, can help to detect, analyse and solve errors, which couldn't be avoided by the regression test. In this case the tests should be extended to regard the arised problem.

To further the confidence of the companies a concurrent versions system (CVS) should show all changes so that the local security approval can focus the relevant parts of a new revision. Such a security audit needs a suitable documentation which helps people to understand the software even if they are new to the used programming language. Tools like doxygen [5] can perform this with its javadoc comment style, generated dependency and inheritance graphs.

The greatest quality source is a modularised concept with well defined interfaces, so that distributed maintenance of the project is possible. This structure must be controlled and updated during the evolution of the project to map new functionality. Every infrastructure change should be introduced by discussions where standard modelling methods (e.g. UML) are used.

Long term commercial management

The Open Source community has enough methods and tools to ensure a very high grade of quality. But this is always dependent on a good project management with generally accepted leaders and an active community which invests development and test time to advance the capability and the quality of the project. As soon as companies plan to base their commercial products on an Open Source project, a good long term project management must be ensured. This can be done if the projects core team starts its own company, which is sponsored by beneficial companies and which offers commercial support for special customer projects, support with granted response time or training of the developers of a company. Alternatively single developers can be hired by companies which earns money with products and services based on the project.

Sponsoring of Open Source projects can be stimulated if the used license forces companies to inform their customers about the use of the

project. This way the project website can publish a ranking of the donations from companies, so that a possible customer can use this for the comparison of alternative products from different manufacturers.

5 Conclusions

A partial project of the research group IKB-Dürnast [10] has the task to gather all possible process data from field work based on the protocol DIN 9684. Lacking suitable products or development tools it was decided to develop an Open Source solution called LBS-Lib [8] under the "Lesser General Public License" (LGPL) [4]. During the analysis all unclear parts of the standard are settled with the help of the standardising group. Its capability outranges the existing commercial products.

Two national and international workshops were hosted to inform the developers of the companies about the project and to stimulate its further evolution. But their reaction revealed the limits of such a project, as most developers are only familiar with closed networks within one machine. As the needs of a dynamic open network are very different from closed ones, they doesn't understand a complex object oriented concept which is designed to realise the complete standard even for complex scenarios. Thus they doesn't accept an object oriented design and its requirements on flash and working memory and microprocessor. Interesting is the fact, that members of two international agricultural terminal manufacturers, who are familiar with such complex systems comprehended the project design, had some concrete suggestions and tries to integrate it in their products.

This shows that the typical disruptive effects of Open Source can only happen, if enough people and companies realise the need for such a project, accept the need for new concepts and have enough experience in the addressed application area to accept the projects design. Otherwise the community of the project stays to small.

6 Acknowledgements

I give my thank to the *Deutsche Forschungsgemeinschaft* (DFG) for financing the research project and to the personnel of the experimental station Duernast.

7 References

[1] Auernhammer, H., Landwirtschaftliches, BUS-System LBS, *KTBL-Arbeitspapier 196*, Germany, 1993

[2] Deutsches Institut für Normung, DIN9684: Landwirtschaftliches BUS System, 1997

[3] Behlendorf, B., Open Sources: Voices from the Open Source Revolution, http://www.oreilly.com/catalog/opensources/book/brian.html, 1999

[4] Free Software Foundation, GNU Lesser General Public License, http://www.gnu.org/copyleft/lesser.html, 1999

[5] van Heesch, D., doxygen, 1997 - 2001

[6] O' Reilly, T., Ten Myths about Open Source Software, http://opensource.oreilly.com/news/myths_1199.html, 2001

[7] Siegert, H. J., Robotik. Programmierung intelligenter Roboter, *Springer Verlag*, Germany, 1996

[8] Spangler, A., LBS-Lib, http://ikb.weihenstephan.de/LBS-Lib/, 2001

[9] Tannenbaum, A. S., Computer Networks, Prentice Hall Inc., Upper Saddle River, New Jersey, USA, 1996

[10] Information System Site Specific Crop Management Dürnast (IKB-Dürnast), http://ikb.weihenstephan.de/en/, 1998 - 2001