Research Report CS-2008-5

# Making SDC-tools better usable by NSIs: Sustainability of the Argus Software

Matthias Templ

December 2, 2008

The aim of this paper is to point the best way to move forward and to ensure the sustainability of $\mu$-Argus [2] and $\tau$-Argus [1] for the future.

## 1 Contributors

At the moment the development of the Argus twins depends strongly on one or two developers at Statistics Netherlands.

To guarantee the minimum requirement - the continuity of the software - the development of the software should be in the hands of a community. Moreover, developing and maintaining a software with the help of a strong community may be allow a step forward to trustworthy computing because the code development requires the highest possible transparency and a community may detect errors in the code sooner.

## 2 Dissemination

In order to make possible that several contributors simultaneously work on the code it is highly recommended to use a online subversion system for code development. Different access rights can then be given to each of the contributors. Each uploaded version is saved and can easily be compared to older versions of the code. Different tasks can be assigned to different developers. Tools for communication between the contributors such as discussion forums, wikis, chats, bug tracking, to do lists, etc. can be used in combination with a modern online subversion system. Already available routines should be provided to the whole community in order to avoid innovations to be introduced twice.

Several different subversion systems are freely available on the web. The most popular can be found on http://subversion.tigris.org/, but there are several other systems to choose from.

# 3 Organisation of the Development

One organisation which has the overview of the whole project should function as a coordinator to guarantee a minimum amount of organisation among the community.

In order to motivate researchers to contribute to the development the code should be splitted in different classes (modules). By this principle, the contributors do not have to knowledge of all the code. Then it is only necessary to know the interfaces for the successful development of parts of the code in the field of interest.

For each module a person in charge should be assigned to guarantee an effective, transparent bug-fixing process. Such a responsible person shall not be elected based on his biography or reputation. It is recommended that such a person in charge should posses detailed knowledge of the code and of code development in general.

# 4 Open-Source Code

One step to trustworthy computing is to provide the entire code via open-source under a *General Public Licence* (have a look at http://www.gnu.org/copyleft/gpl.html), because it is likely that there is a growing number of people working with the source code. Everybody has then access to the code, has the possibility to learn from it, evaluate and modify it and contribute to the development of an open source software project. Since most of the code from $\mu$- and $\tau$-Argus is written in C or C++ the code can be integrated and used in other software.

Without providing the entire code available to the community as an open-source code, researchers would hardly join the development. By a freely available code we mean the entire code of the procedures integrated in the Argus twins as well as the available code of procedures written by the contributors.

# 5 Statistical Software

$\mu$- and $\tau$-Argus can be seen as a stand alone implementation (code: C or C++, graphical user interface: Visual C++). The usage of additional complex statistical routines is only possible by sending statements to a statistical software via batch mode and vice versa. So, Argus and other software has to be installed to be able to use statistical routines which already exists. The integration of the routines of Argus in a statistical software makes an additional software installation unnecessary. However, the integration of Argus into another software is highly labour-intensive.

# 6 R

## 6.1 Integration of Argus into R

If it will be decided to integrate Argus into another software one has to think into which software the code from Argus should be implemented. Since all NSI's should make use

of the SDC tools, it is highly recommended to choose the software system R ( [3]). R is highly expendable, can be used for free and is the standard statistical software in the world. In addition to that, many researchers on SDC and NSI's already do their development of SDC methods in R. It is then possible to invite these researchers to contribute to a common research infrastructure on SDC. For an integration of Argus into R major problems occur whenever the code from Argus is developed within Microsoft C or C++ Compilers. It is then not possible to integrate the code directly and some parts of the code must be rewritten, which can be cumbersome.

### 6.2 Calling R from Argus

On the other hand it is also possible to run R from Argus via the R batch mode functionality. Routines from R can then be used from Argus. Since, a lot of sophisticated methods are available in the R-package *sdcMicro* ( [4, 5]), one can simple apply these methods via batch mode.

### 6.3 Calling Argus from R

The last option is to run $\tau$-Argus within R which can be easily done. Since $\mu$-Argus cannot be run from batch mode, R can hardly call all the routines of $\mu$-Argus.

The main disadvantage to use R is that the users must learn a programming language. It is hard to get a R-developer and very time-consuming (some months to some years depending on the previous knowledge of other object oriented programming languages). Unfortunately, one have to know R very well to make proper use its advantages. Within R it would be possible to reproduce any step of the anonymisation process very easily and to apply the methods in an explorative way. Such "play-back" options and the explorative application of SDC methods are not possible within the current implementation of Argus and only hardly possible within a possible future version of Argus, but can easily applied in the R environment.

The main advantage of integrating Argus in R is that all the functionality of R can be easily accessed, especially the development tools of R, the graphical excellence of R, the data import/export facilities and the implemented statistical methods from R, but also functionalities to create graphical user interfaces are available. However, the development of a new graphical user interface is very labour-intensive and the documentation of the graphical user interface development facilities is very limited.

## 7 Cross Platform Independence

In these days Microsoft Windows is present on nearly every computer of the NSI's. However, Linux becomes more and more important and it is likely that in future more and more NSI's switch to a Linux operating system. The development of the Argus twins correlates highly with the Microsoft Windows operating system, using the C or C++ compilers from Microsoft and Visual Basic for the graphical user interface, for example.

Nevertheless, the code should be successively integrated in a cross platform independent code, i.e. to use freely-available, open-source, cross-platform independent compilers and free toolkits for generating the graphical user interface (GTK, QT or tcl/tk). The advantage is not only to be able to provide the code to a broader community but also to be independent if Microsoft changes their operating system or essential parts of their environment.

## 8 Maintenance, Intellectual Rights and Financial Aspects

To provide freely-available, open-source code the intellectual properties of the authors must be fully respected. Each procedure has its own author(s) which may be themselfs a procedure of a part of the whole development of Argus. One or two maintainers must maintain an online repository, each contributor should have access to change parts of the code in an online repository.

Free open-source software development of SDC software is heavily dependent of a broad and motivated development community from NSI's and Universities. A permanent financial body would be fine (but less important as a motivated community) and would help a lot to set activities to support the community (organisation of meetings, etc.).

## 9 Conclusion

The development of Argus was funded by the European Commission within the FP7 project CASC, but also by Eurostat within several projects (a description of all these projects can be found on http://neon.vb.cbs.nl/CASC/). But not only for this reason the entire code of Argus should be freely-available. The distribution of Argus as open-source already guarantees that Argus or (the most essential or best written) parts of the code of Argus will be used in future. To avoid that people sell this code within commercial software, the code should be provided in form of a GPL, version 2 or newer. This helps also that researchers which only will use an open-source version of the code for their own purposes have to respect the copyright of the code and the intellectual properties of the authors.

An open-source development platform for Argus allows to keep control over the development. In order to motivate researchers to contribute to a common open-source project, it is recommended to provide modern development tools which allow an effective organisation of the work. To be as attractive as possible it is recommended to build the possible new Argus on top of a powerful statistical system such as R.

The integration of Argus into another software sounds (theoretically) promising, but unfortunately, the amount of work may be higher as a re-implementation of the methods.

Several activities for such re-implementations already have set up, because researchers often need the source code to understand, extend and change the code for their purposes. For $\mu$-Argus the methods are already re-implemented in the R-package *sdcMicro*. The authors of this package will also try to provide the $\tau$-Argus counterpart in the near

future as an R-package, whereas the time for providing this package will be reduced if the source code of ($\tau$-)Argus will be provided.

Acknowledgment:
The authors work on this report was partly funded by Eurostat within the project *ESSnet on Statistical Disclosure Control (2008-2009)*.

## References

[1] A. Hundepool, R. Ramaswamy, de Wolf P-P., L. Franconi, S. Giessing, Matteo Fischetti, , J.J. Salazar, C. Castro, and P. Lowthian, 2007. http://neon.vb.cbs.nl/casc.

[2] A. Hundepool, A. Van deWetering, Ramaswamy R., L. Franconi, S. Polettini, A. Capobianchi, P-P. de Wolf, , J. Domingo-Ferrer, V. Torra, R. Brand, and S. Giessing. $\mu$-argus version 4.1 software and users manual, 2007. http://neon.vb.cbs.nl/casc.

[3] R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[4] M. Templ. sdcMicro: A package for statistical disclosure control in R. In *Bulletin of the International Statistical Institute, 56th Session*, 2007.

[5] M. Templ. *sdcMicro. Manual and Package. Version 2.5.1.* Statistics Austria and Vienna University of Technology, Vienna, Austria, 2008. *http://cran.r-project.org/src/contrib/Descriptions/sdcMicro.html*.