Project no.: **619209**

Project full title: **Analysis of Massive Data STreams**

Project Acronym: **AMIDST**

Deliverable no.: **D9.3**

Title of the deliverable: **Open source software strategy report**

Contractual Date of Delivery to the CEC:	**30.04.2015**
Actual Date of Delivery to the CEC:	**30.04.2015**
Organisation name of lead contractor for this deliverable:	**AAU**
Author(s):	**Helge Langseth, Anders L. Madsen, Thomas D. Nielsen, Antonio Salmerón**
Participants(s):	**P01, P02, P03, P04**
Work package contributing to the deliverable:	**WP9**
Nature:	**R**
Version:	**1.0**
Total number of pages:	**14**
Start date of project:	1st January 2014 **Duration:** 36 month

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:

The open source software strategy document defines the key decisions adopted to establish an open source project attached to AMIDST aimed at creating a community around it. This will help as a means to supporting future research and application scope extending the results and developments made during the lifespan of the research project. The strategy includes decisions that affect design aspects of the AMIDST toolbox as well as software licensing.

Keyword list: open source software, strategy.

Table of Contents

Document History

Version	Date	Author (Unit)	Description
0.3	22/7/2014	Antonio Salmerón	First draft finished
0.6	23/4/2015	Helge Langseth, Anders L. Madsen, Thomas D. Nielsen, Antonio Salmerón	Initial draft reviewed by the PSRG
1.0	29/4/2015	Hanen Borchani, Helge Langseth, Anders L. Madsen, Ana M. Martínez, Andrés Masegosa, Thomas D. Nielsen, Antonio Salmerón	Final version of document

1 Executive Summary

The open source software strategy document defines the key decisions adopted to establish an open source project attached to AMIDST aimed at creating a community around it. This will help as a means to supporting future research and application scope extending the results and developments made during the lifespan of the research project. The strategy includes decisions that affect design aspects of the open source AMIDST toolbox as well as software licensing.

The starting point of AMIDST as an open source software project (OSP) will coincide with the end of the AMIDST research project. The initial release will comprise the developments carried out in WPs 2, 3 and 4.

The strategy definition has taken into account the experience reported from renowned successful open source projects (Stürmer, 2005) and the particular characteristics of the AMIDST project.

2 Introduction

According to the Open Source Initiative[1] (OSI), beyond access to the code, open source software is defined in terms of the following ten requirements:

1. **Free redistribution**. The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.
2. **Source code**. The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicised means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.
3. **Derived works**. The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.
4. **Integrity of the author's source code**. The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.
5. **No discrimination against persons or groups**. The license must not discriminate against any person or group of persons.
6. **No discrimination against fields of endeavour**. The license must not restrict anyone from making use of the program in a specific field of endeavour. For example, it may not restrict the program from being used in a business, or from being used for genetic research.
7. **Distribution of license**. The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
8. **License must not be specific to a product**. The rights attached to the program must not depend on the program being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.
9. **License must not restrict other software**. The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

[1] http://opensource.org/osd

10. **License must be technology-neutral**. No provision of the license may be predicated on any individual technology or style of interface. In this sense, conformant licenses must allow for the possibility that (a) redistribution of the software will take place over non-Web channels that do not support click-wrapping of the download, and that (b) the covered code (or re-used portions of covered code) may run in a non-GUI environment that cannot support popup dialogues.

The requirements described above are related to the chosen license, and licensing is indeed an important aspect to take into account when setting up an open source software project. The AMIDST open source software project fulfils the above-mentioned requirements by adopting a license compatible with the OSI (Apache License, Version 2.0), as we will discuss in Section 4.2.

3 Elements of an open source software project

Figure 1, taken from Stürmer (2005), shows an overview of the elements of an Open Source Software Project (OSP).

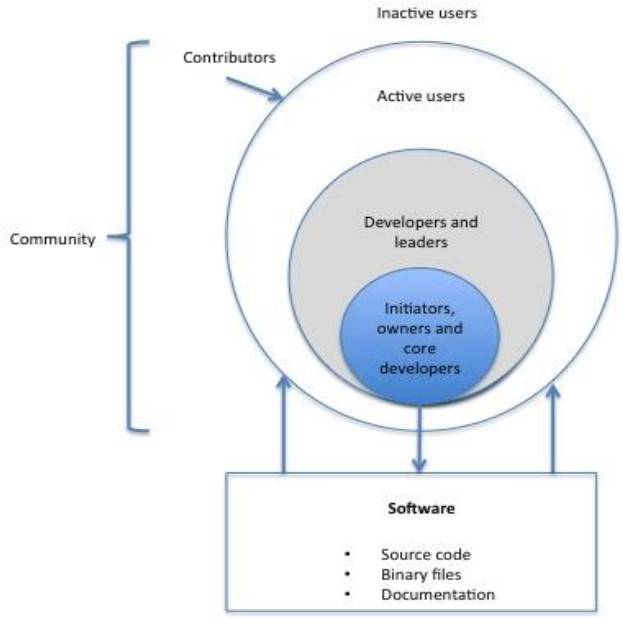
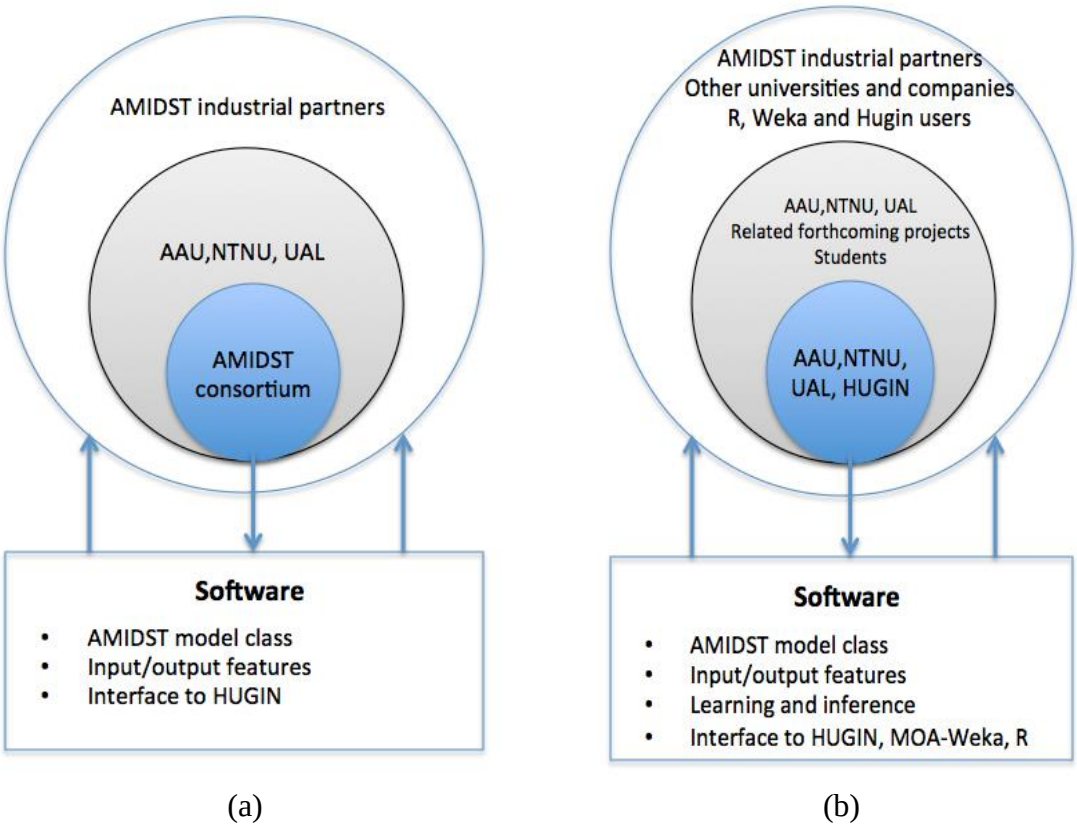


Figure 1. Elements of an open source software project

The terms in the figure are defined as follows.

- **Software**. The open source software (OSS) consists of the complete source code, the executable binary files and all documentation artefacts.
- **Contributors**. All the active members of the OSP, for which their individual aspects and roles must be defined.

- **Active users**. Users of the software that participate in mailing lists, report on bugs, contribute with donations, etc.
- **Inactive users**. Users of the software that do not provide any feedback to the OSP.
- **Developers**. Those who actively work on the source code. A developer does not necessarily have access to the code repository. Programming skills are a must.
- **Leader**. Responsible of the human aspects of the OSP, as well as the promotion (together with the developers) of the community building, taking care of management tasks.
- **Core developers**. A small group of programmers responsible for the largest part of the software development. Unlike other developers, they have a long experience in the OSP and have a deep knowledge of its history and architecture.
- **Project owners**. A group of people forming an executive board responsible of coordinating and controlling the evolution of the OSP. They have root access to the development server or the collaboration platform and are thus able to grant code commit access to new developers.
- **Initiators**. First persons in the project contributing from the beginning.
- **Community**. The set of all contributors, but from an external point of view, covering aspects as shared values and common vision.



(a) (b)

Figure 2. Instantiation of the diagram in Figure 1 to the AMIDST OSP. The current status is shown in (a) and the desired status at Month 36 of the project is displayed in (b).

Figure 2(a) shows the current status of the AMIDST OSP. The *software* is currently composed by the AMIDST model class defined in Deliverable 2.1 (Borchani et al. 2014), facilities for reading and writing files, and an interface module able to interact with the HUGIN COTS software. Current

active users are the industrial partners of the AMIDST consortium responsible of the use cases, namely Daimler and BCC-Cajamar. The *developers* are the three academic partners (AAU, NTNU and UAL), basically through their post-docs. The *project owners* are the members of the AMIDST consortium, where the project management is effectively carried out by the Project Science Review Group, composed by one representative from HUGIN, AAU, NTNU and UAL. Finally, the *initiators* are all the members of the AMIDST consortium.

The plan is to move towards the scenario depicted in Figure 2(b) by the end of the project lifespan. The software will be fully functional, containing implementations of the inference and learning algorithms developed in the project. The three academic partners are expected to be the core developers, and the group of developers will be expanded by including staff from future projects related to AMIDST. Students are expected to contribute with developments associated with their Master's/PhD theses. The set of active users should be significantly bigger by the end of the project, thanks to the links to other software platforms HUGIN, MOA-Weka and R (see Section 5.7)

4 Prerequisites of an OSP

This section covers the issues that must be addressed before an OSP takes off. It includes decisions about the programming language, licensing and analysis of the timeliness of the project.

4.1 The programming language

The choice of the programming language is crucial as it affects the design of the software itself from the beginning. It also has an impact on the potential future contributors to the OSP. There is a wide variety of alternatives, ranging from scripting languages as Python to traditional languages as Java and C. While the choice of a scripting language might expand the set of potential contributors, it could limit the features of the software as, for instance, in terms of efficiency. On the other hand, the use of an efficient language like C could dramatically prevent possible contributors to participate in the OSP.

Taking this into account, together with the past experience of the AMIDST partners in software development, the **selected language** for implementing the open source **AMIDST toolbox is Java.** The minimum required version is **Oracle Java 8,** which, with respect to previous versions, provides functional programming and scalability facilities. Java is sufficiently spread among potential future contributors to the OSP and allows the development of highly efficient software. Furthermore, it provides a multi-platform environment, what is an added value for heterogeneous communities.

4.2 License for the AMIDST open source toolbox

The open source AMIDST toolbox comprises the software to be developed in WP2, WP3 and WP4. The choice of a license has a clear impact on the development of the project, as it can limit the ways in which the project is allowed to grow up (Scacchi and Alspaugh, 2012). The license initially considered in the Description of Work of the AMIDST project for the AMIDST open source toolbox was the LGPL-3.0[2]. The choice of LGPL would allow the use of the AMIDST toolbox in commercial software, as it does not require that the source code of the entire product is made available, but only the part that is under the LGPL license, as long as the software is not a derived product of AMIDST, i.e., AMIDST is not embedded or modified in the new product, but just used

[2] http://opensource.org/licenses/LGPL-3.0

through an interface, for instance. A strong point of LGPL is that it would potentially promote the distribution of AMIDST, as it would be distributed in any derived product thereof. The controversial aspect of the LGPL license is that any derived product (i.e., a product that modifies or embeds the AMIDST toolbox) should also be licensed under LGPL. This aspect can be a limitation for commercial development.

Taking this restriction into account, the consortium decided to use a less restrictive license, namely the **Apache License, Version 2.0**[3]. This change was discussed during the first Review on January 22nd, 2015, and accepted by the Project Officer.

The Apache license is inspired by the idea of free software, and does not impose any limitation by itself on the use of the licensed software, as long as the license is properly referenced. It means that commercial products would be able to embed and modify the AMIDST toolbox and still not be forced to distribute their derived source code. Both licenses, LGPL and Apache, are compatible with the OSI requirements.

4.3 Other prerequisites

Other relevant prerequisites for an OSP are the following.

- **Availability of a high quality initial source code**. This facilitates the incorporation of contributors (Raymond, 2001). This item is guaranteed as the initial source code will be the result of WPs 2, 3 and 4, and will therefore consist of fully operative software.
- **Public demand**. The demand of the AMIDST open source toolbox is potentially high, as indicated by the variety of application domains covered by the industrial partners in the AMIDST project.
- **Degree of novelty**. The main novelty of the AMIDST open source toolbox will be the use of technology based on probabilistic graphical models for processing massive data streams. This will come along with new methodological developments as well as adaptation of existing techniques in order to scale up to handling massive data streams.
- **Applicability**. This is also ensured by the composition of the AMIDST consortium, with industrial partners covering distinct industry sectors, namely automobile industry and finance.
- **Early releases**. Early code releases can be used as ways to support the engagement of other developers, to raise awareness of the AMIDST toolbox and to support initiatives on building a community around the software. The **first release** is scheduled on **June 2015**, and includes the AMIDST modeling framework, input/output features and an interface to the HUGIN COTS software. The **second release** will include the inference module and the ability to interact with the R statistical software. It is scheduled on month 24 (**December 2015**), in order to be available for the next Review meeting. The third release is expected on December 2016, and will incorporate the learning modules and the connectivity to MOA-Weka (see Section 5.7).

[3] http://opensource.org/licenses/Apache-2.0

5 Elements of a long-term OSP strategy

After a successful initialisation, several aspects influence the long-term trajectory of an OSP (Stürmer, 2005). Each item has an impact on three dimensions of the OSP: **recruiting** of new contributors to the OSP, **collaboration** among the contributors and **production** of software releases.

5.1 Modularity

- *Recruiting*. The design of the software architecture can facilitate the arrival of new contributors. The structure of the AMIDST OSP should be extensible and all the necessary resources like documentation of plug-in development and extension manager should be made available. The AMIDST software is built around the Maven automation tool for managing software projects and is organised according to the directory structure recommended by the Maven project.[4] Maven has been chosen because it is widely spread and renowned for its abilities to handle dependencies.
- *Collaboration*. Modularity allows the specialisation of the developers in different aspects of the software, making their contributions somehow independent of the core developers. This means that extensions should undergo a quality assurance process by qualified members of the community, and also that extensions should be thoroughly documented.
- *Production*. If much of the functionality is put into external components, the kernel can remain small and robust, what facilitates the development around it. An added value for production is that modularity allows the interaction with **external software**.

5.2 Documentation

- *Recruiting*. Documentation is the basic vehicle for knowledge transfer, but it also provides an easy way for newcomers to contribute, as they can start off writing documentation material in order to get familiarised with the software. Documentation should contain few and clearly focused artefacts. It should be kept up-to-date.
- *Collaboration*. Elaborating documentation is a laborious task and should therefore be promoted. Authors of the AMIDST OSP documentation will be publicly acknowledged and publications around the documentation will be fostered.
- *Production*. The source code must be completely documented in order to allow new contributions and also to support other programmers to develop new applications based on the AMIDST open source toolbox. This means that the *application programming interface* (**API**) must be especially well documented.

5.3 Collaboration platform

- *Recruitment*. Some collaboration platforms allow users to give their opinion on the repositories they contain. The opinion can be collected, for instance in terms of giving stars according to the users' preferences, so that the number of stars is used to generate a ranking of software repositories within the platform. Getting a high rank on a collaboration platform

[4] http://maven.apache.org/

makes the OSP visible to potential contributors. Leaders of the AMIDST OSP will enforce the active use of the collaboration platform services, in order to increase the activity ranking of the OSP. AMIDST will also have its specific web site with the aim of highlighting its individuality.

- *Collaboration*. The responsibility of a collaboration platform is to provide modern, fast and reliable services for the work on the OSP. Also the development team requires a certain freedom to configure the platform for their needs to efficiently work on the software (Stürmer, 2005). The platform chosen for the AMIDST OSP is **GitHub**[5]. It incorporates powerful collaboration features as integrated issue tracking, team management, bug reporting and features requests.
- *Production*. GitHub facilitates the collaborative software development with tools as the collaborative software review and the revision control system, compatible with **git**[6] and **svn**[7].

The relation between the tools supporting the development of the AMIDST OSP is explained in Figure 3. Appropriate links to both platforms will be included in the AMIDST website (amidst.eu).

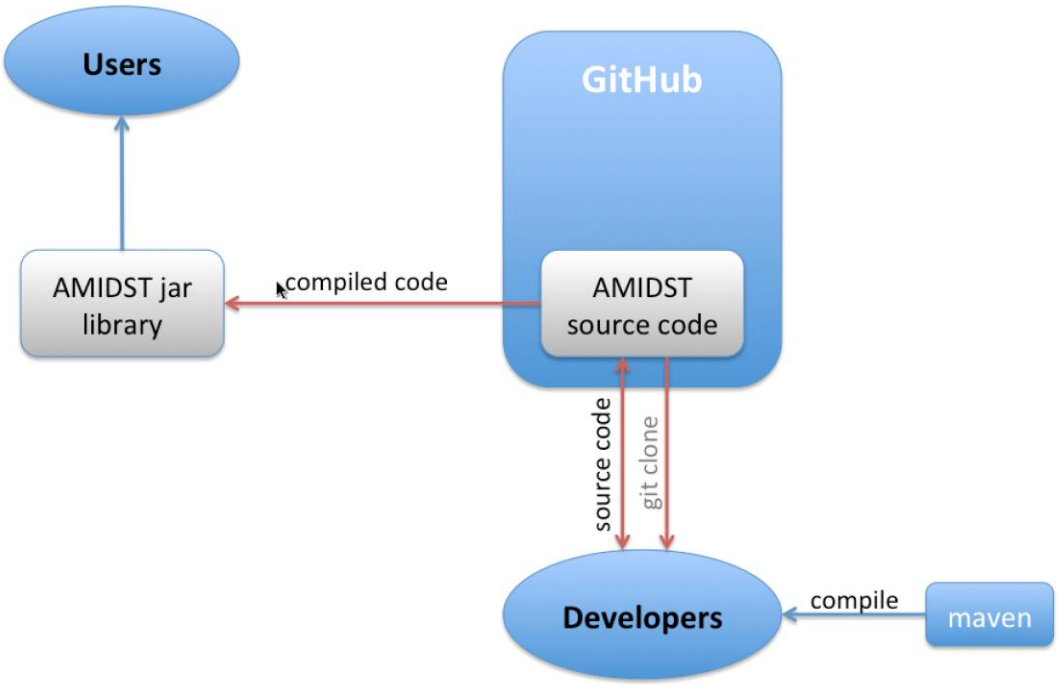


Figure 3. Relation between the platforms supporting the AMIDST OSP.

[5] github.com
[6] http://git-scm.com
[7] http://subversion.apache.org

5.4 Release management

- *Recruiting*. Frequent releases of new versions help to make the OSP visible and increase the attractiveness to participate in it. The AMIDST OPS will release frequent updated versions incorporating the patches developed by the contributors as well as new developments. One or more members of the community will be responsible for the release management.
- *Collaboration*. The interest of the community must be taken into account when deciding the release process. Parts of the community may be interested in different branches of the OSP. The adoption of a plug-in based architecture facilitates to keep a stable core software and release new versions as plug-ins related to the different branches.
- *Production*. It is important to keep backward compatibility in new releases. To enforce this, the **API** must be clearly **specified from the beginning** and kept **stable** during the life span of the AMIDST OSP. Any change must be thoroughly documented and, if possible, migration scripts should be designed to facilitate compatibility.

Beyond the initial releases specified in Section 4.3, we aim to release new versions at least every second year. Our plan is to use the AMIDST software in future projects, which may increase the frequency of new releases.

5.5 Physical meetings

- *Recruitment*. Presentation of the OSP gives the possibility of making personal contacts and using the software, thus stimulating new people to contribute to the project. Appropriate events would be open source exhibitions as well as specific forums like the European Workshop on Probabilistic Graphical Models (PGM).
- *Collaboration*. Physical meetings are a means for intensifying personal relationships of contributors, for knowledge transfer and for decision making.
- *Production*. By including intensive programming activities in the meetings, like programming in pairs, developers will have the chance to speed up their contributions to the OSP.

5.6 Foundation

- *Recruitment*. The existence of a reputed and acknowledged foundation behind an OSP can be beneficial in general, and particularly it can increase the confidence of users and help on recruiting new ones. The AMIDST OSP will inherit the reputation of the AMIDST consortium and the results of the AMIDST research project will be a guarantee of quality.
- *Collaboration*. The existence of a foundation or a responsible organisation behind the OSP will bring stability and compensate for the fluctuation of contributors. It will also take care of the licensing aspects of the OSP. The partners of the AMIDST research project will ensure the existence of a stable structure giving support to the AMIDST OSP after the research project is completed.
- *Production*. The organisation responsible of the AMIDST OSP will seek for funding sources to support the most active contributors.

The organisation behind the AMIDST OSP is expected to be composed by at least one of the academic partners in the consortium.

5.7 Internationalisation

- *Recruiting*. The AMIDST OSP is an international endeavour from its origin, and hence its scope is wider than a local project's. Furthermore, the possibility of meeting people from various places can serve as an incentive for getting new contributors. Another way of promoting internationalisation of the AMIDST OSP will be by interacting with other open source and COTS software platforms. This will be achieved by implementing the appropriate interfaces within the AMIDST open source toolbox. Interfaces enabling interaction with COTS software HUGIN[8], the R statistical software (R Core Team, 2014) and the MOA-Weka tool for massive online analysis[9], will be developed.
- *Collaboration*. Getting contributors from all over the world helps the project to be active at any time in the day, what is a motivating issue for contributors.
- *Production*. The international character of the AMIDST OSP requires that all source code comments and documentation be entirely issued in English, setting it as the common language for the OSP.

6 Summary

This document describes the open source software strategy for the AMIDST OSP. As opposed to other OSPs, the AMIDST OSP will be built upon the outcome of the AMIDST research project, and will have the support of a strong consortium behind it. Nevertheless, a long-term strategy is necessary to guarantee the success and stability of the OSP. The key points of the strategy are the following.

- The programming language is Java. The minimum required version is Oracle Java 8.
- The software will be released under Apache License, Version 2.0.
- The software design will adopt a modular architecture.
- There will be a stable API thoroughly documented. The API will provide the functionality of the open source AMIDST toolbox.
- The collaboration platforms are GitHub and Maven.
- Physical meetings will be fostered, both in host conferences and workshops and in dedicated events.
- The members of the AMIDST consortium will articulate the necessary collaboration to set up a solid organisation behind the AMIDST OSP.
- The open source AMIDST toolbox will interact with other platforms through interfaces. Interfaces to Hugin, R and MOA-Weka will be made available.

[8] http://www.hugin.com
[9] http://moa.cms.waikato.ac.nz

7 References

1. H. Borchani, A. Fernández, O.E. Gundersen, S. Hovda, H. Langseth, A.L. Madsen, A.M. Martínez, R.S. Martínez, A. Masegosa, T.D. Nielsen, A. Salmerón, F. Sørmo, G. Weidl: *The AMIDST modelling framework – Initial draft report* (2014) Deliverable 2.1 of the AMIDST project, amidst.eu.
2. E.S. Raymond (2001) *The Cathedral and the Bazaar: Musing on Linux and Open Source by an Accidental Revolutionary*, Revised Edition. O'Reilly Media. Sebastopol, CA.
3. R Core Team (2014). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.
4. W. Scacchi, T.A. Alspaugh (2012) Understanding the role of licenses and evolution in open architecture software ecosystems. *The Journal of Systems and Software* 85, 1479-1494.
5. M. Stürmer (2005) *Open source community building*. Master Thesis. University of Bern.