

THE MANTID PROJECT: NOTES FROM AN INTERNATIONAL SOFTWARE COLLABORATION*

N. Draper, Tessella, Abingdon Science Park, UK

INTRODUCTION

The analysis challenges faced by large scale neutron scattering and muon spin resonance (μ SR) facilities continue to grow, with increasing user demand and ever increasing data volumes. One of the single most important barriers between experiment and publication is the complex and time consuming effort needed to reduce and analyse their data.

The Manipulation and Analysis of Instrument Data (MANTID) project[1] is a large international collaboration between ISIS, STFC (UK) and SNS, DOE to co-develop a high performance computing framework for the analysis of neutron scattering and muon spin resonance data. The MANTID software is cross platform and open source so it is free and available to all visiting scientists of the facilities.

PROJECT HISTORY

The MANTID project was started at the ISIS facility in 2007 with the goal of implementing a new framework for data analysis and visualisation for neutron scattering and μ SR experiments. Previous software projects at ISIS had generally concentrated on the needs of a few of the beamlines with the result that wider acceptance has been limited. Mantid is the first software project at ISIS to be fully funded and have a core of full time developers solely working on Mantid. The project began as a joint collaboration between the ISIS facility and the specialist scientific software consultancy Tessella providing the project management and a core of experienced software developers.

In 2010 the project was joined by Spallation Neutron Source (SNS) at Oak Ridge National Laboratory, significantly increasing the development team, while adding the additional needs of that facility.

Currently several of the remaining large neutron facilities and those planned to be built are evaluating the framework with a view to joining the collaboration.

AN OVERVIEW OF THE ARCHITECTURAL DESIGN

At the heart of the Mantid system is the Mantid framework where all of the data and analysis code is centred. The centralisation of all of our code in a framework that is independent of any visualization greatly

eases the task of introducing new user interfaces and maintaining multiple user interfaces as each is only concerned with the presentation of data to users, all data manipulations are done using the framework. So far we have built two primary user interfaces: a graphing and analysis tool called MantidPlot that has been built as an extension to the open source project QtPlot[2], a Python interface that provides us a command line and scripting interface. An early proof of concept Matlab interface was created, but has not to date been taken further.

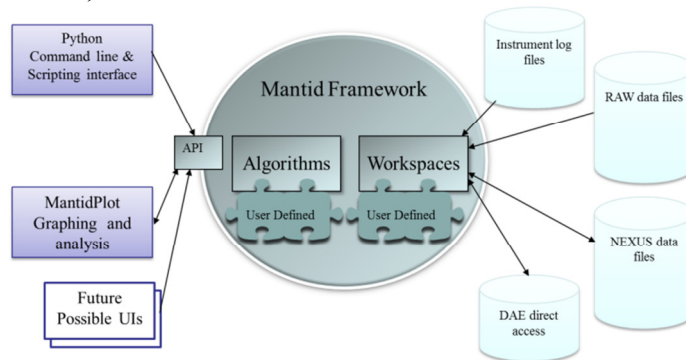


Figure 1: High level design of the MANTID project

The framework itself is built around two main concepts: algorithms that can manipulate data and workspaces that hold the data. The rest of the framework provides useful services to those two main concepts. While there are many algorithms and workspace types already built for the framework Mantid provides a plug in system to greatly ease the task of adding new algorithms or workspaces should a user require a manipulation that we have not provided.

Data input and output are provided using specialized algorithms. We support a wide range of facility specific and NEXUS formats[3], with the framework dynamically selecting the best file loading algorithm to match the file format. We also support live data loading and analysis by direct connection to the instrument data acquisition electronics.

PROJECT ORGANISATION

The organisation of any project, particularly collaborations working with a distributed team is vital to its success. In this case some of the key elements of this project were the need for clear direction and oversight within the project, together with the need to develop a suitable solution for the many thousands of scientists that visit these facilities every year. With this number of end users across all of the scientific techniques the facilities support it is not possible to interact with them directly. To ensure we have a broad coverage of all of the issues and

* Work at ORNL was sponsored by the Scientific User Facilities Division, Office of Basic Energy Services, US Department of Energy. Work at the ISIS facility was funded by the Science and Technology Facilities Council (STFC). Development for ILL instruments was funded by NMI3 (WP6).

needs of the beamlines a scientific steering committee has been established at each facility with at least one key scientist from each group of beamlines. The committees work together to identify areas for future development, prioritise the various tasks and maintain communication between the project team and the wider user group.

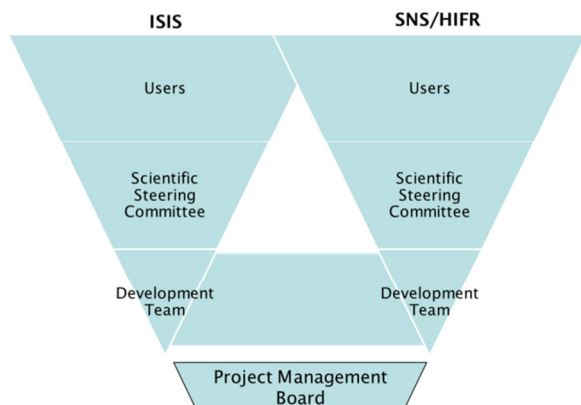


Figure 2: Organisation of the MANTID project

The development team, while distributed across two physical locations, work together as a single co-ordinated team, with a development team leaders working with a single project manager to provide a consistent service to both facilities. A joint project management board with members from all involved facilities provides oversight and strategic direction.

Team Communication

With any distributed team communication is vital in order to work effectively together; however the vital need for communication must be weighed against the time cost incurred. Within the Mantid project we use several paths of communication to ensure that the team know what everyone is doing while minimising the time spent away from development. The team make extensive use of Skype, for intra team calling and messaging, but also for a daily distributed “stand up” meeting that keeps everyone informed while taking very little time of everyone’s time. For more in depth code review and demonstrations screen sharing or video conferencing are used to minimize the need for travel.

The need for face to face meetings to build relationships and solve tricky issues cannot be underestimated, so in addition to frequent visits by the project manager to all of the facilities involved we also hold an annual developer workshop where all of the key developers meet at one of the facilities.

When the SNS joined the MANTID project there was also the opportunity to transfer a senior developer from the ISIS team to the SNS. This was an invaluable direct transfer of knowledge and skills to the newly joined facility and helped them rapidly understand the codebase and adopt MANTID within the organisation.



Figure 3: Several of the members of the MANTID Team at their annual developer workshop meeting in 2010.

Development Ecosystem

Building a large cross platform analysis framework is a significant challenge to any development team, and in order to optimise productivity the team they need a supportive development ecosystem to automate and ease tasks wherever possible. All work to be undertaken is stored in our Trac[4], online ticketing system. Issues are given a priority and assigned to individual developers.

Developers cannot submit changes to the codebase unless it contains a tag cross-reference to at least one ticket from Trac.

Initially the project was successfully run using SVN[5]. However, as the project grew, both in terms of the team size and the complexity, centralized version control became cumbersome. Git[6] is now used, with public, remote repositories hosted on GitHub[7]. The ability to generate branches easily and work offline, where the

major drivers. Committing to a local repository before pushing change-sets has reduced the load on the build servers and generally lead to smaller explicit individual commits.

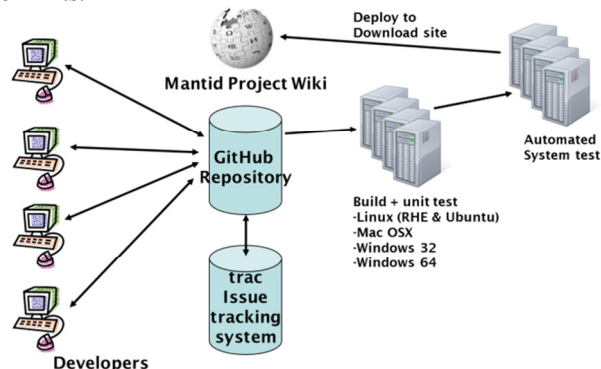


Figure 4: The MANTID development ecosystem

The Mantid project makes extensive use of Jenkins[8], a continuous integration server. Continuous integration is ultimately a quality control methodology that continuously builds and tests the software product. Fault and failure detection therefore is achieved at the earliest possible time and it is therefore significantly less time consuming or expensive to fix.

Our setup builds and tests the software on all supported environments with every commit to Git, providing rapid feedback to developers of any faults introduced. This duplicated at each facility, so that any server outages will cause minimum disruption to normal working.

In addition to the extensive automated unit testing performed by the build servers, we also have a second level of more intensive automated system tests that evaluate common workflows against full datasets and compare the results to stored reference datasets to ensure that inadvertent changes will not affect the results of data analyses. These system tests are run automatically at least every night, again providing rapid feedback to the development team.

A full MANTID release occurs every three months, but there are times when more frequent releases are required while working closely with users to develop new functionality. MANTID supports provides nightly releases for this purpose, every night if the builds and automated system tests have completed successfully the installers are automatically uploaded to the MANTID download site, and Linux repositories and are immediately available for users to download. All of this occurs without the need for any time from the development team.

Testing and the Release Process

Automated testing is incredibly useful, but cannot be the sole testing process for a quality software system. Within the MANTID project we add three further testing approaches to ensure that MANTID delivers a reliable robust system. The first is manual developer testing, each change submitted by a developer is code reviewed and

functionally tested by a different member of the development team. As well as ensuring better testing than by the author themselves this improves communication and shares knowledge within the team. This testing has to pass before the authors changes will be allowed onto the main “release” branch of the code.

In addition to the daily development release MANTID has full releases every 3 months that are accompanied by full documentation, release notes and user training. For these full releases additional testing is also included to ensure the software is robust and reliable. Scripted and unscripted manual testing of all of the user interfaces on each environment is performed to find any user interface or usability bugs. This is time boxed with frequent changes of area to maintain the motivation of the testers.

Finally once the release candidate has been well tested by the development team select instrument scientists are invited to a short beta test phase. This often provides high quality feedback both for the version under test and for future developments, as well as building confidence with the instrument scientists themselves.

CONCLUSION

The data analysis challenges for many facilities are increasing much faster than the resources of the scientific computing groups that service them. Collaboration between facilities on projects to solve common challenges is one approach to tackle this problem. The MANTID project is one of a few successful cross facility software projects within the neutron scattering and μ SR community and is continuing to grow.

ACKNOWLEDGMENT

The development team would like to thank all of the instrument scientists at ISIS and SNS for their feedback and help.

REFERENCES

- [1] The Mantid Project, <http://www.mantidproject.org>
- [2] QtPlot, <http://soft.proindependent.com/qtiplot.ntml>
- [3] NeXus Format, <http://www.nexusformat.org>
- [4] Trac, <http://trac.edgewall.org>
- [5] Subversion, <http://subversion.apache.org>
- [6] Git, <http://git-scm.com>
- [7] GitHub, <https://github.com>
- [8] Jenkins, <http://jenkins-ci.org>