ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THCPL05

TANGO HEADS FOR INDUSTRY

A.Götz, J.M.Chaize, R.Bourtembourg, P.Verdier, V.Michel,
T.Coutinho, J.L.Pons, ESRF, Grenoble, France
I.Khokhrakiov, IK, Moscow, Russia
P.Goryl, 3Controls, Krakow, Poland
A.Stanik, Prevac, Rogów, Poland
S.Gara, Nexeya, France
G.Mant, STFC, UK
S.Viénot, JYSE, France

Abstract

The TANGO Controls Framework continues to mature and be adopted by new sites and applications. This paper will describe how TANGO has moved closer to industry with the creation of startups and addressing industrial use cases. It will describe what progress has been made since the last ICALEPCS in 2015 to ensure the sustainability of TANGO for scientific and industrial users. It will present TANGO web based technologies and the deployment of TANGO in the cloud. Furthermore it will describe how the community has re-organised itself to fund and improve code sharing, documentation, code quality assurance and maintenance.

INTRODUCTION

The Tango community continues to grow with more and more small and large sites adopting Tango. There are currently over 40 sites world-wide covering synchrotron light sources, lasers, wind tunnels, telescopes and physics experiments. At least 15 industrial companies provide TANGO support or use TANGO in their projects. The most recent large sites to join over the last 2-3 years are the 3 ELI laser sites, the SKA radio telescope project, a number of facilities in Russia, India and China. In addition to large sites a number of smaller sites have adopted Tango including some startups. Long standing members like the ESRF are basing new projects like the EBS [1] on Tango. The increased adoption of Tango for long term projects puts a strong requirement on the community to make Tango sustainable in the medium and long term i.e. 10 to 30 years. This is a tall order for any software project and needs specific actions to ensure it happens. This paper explores how the long term sustainability is being addressed with the help of industry for the community and industry.

SUSTAINABILITY

A good definition of Software Sustainability is

Sustainability means that the software you use today will be available - and continue to be improved and supported - in the future.

This definition comes from the Software Sustainability Institute [2] and covers the essential aspects of "**availability**", "**continued improvement**" and "**support**". These three topics illustrate well that software sustainability is not restricted to what is commonly called software maintenance (Fig. 1). There is no widely accepted unique definition of these topics [3] and so each project interprets sustainability according to its needs. In the case of the Tango Controls core the three areas are defined as:

1. **availability** – guaranteed open source licence, public source code repositories, open development workflow
2. **continued improvement** – continue to improve code to support new features, interfaces, languages, tools and paradigms to satisfy users needs
3. **support** – bug fixes, porting to new platforms, maintain backwards compatibility

Another very important issue is Sustainable Software Design [4]. The software has to be designed and architected in a way to make it sustainable. This means the core ideas must be kept intact or be able to adapt to new needs or new platforms without a huge effort. The core architectural features of Tango Controls are:

- **distributed devices** – Tango is based on network agents each implementing control algorithms, a state machine, methods and data fields accessed via the network. This central concept called the Device in Tango is implemented in the Device Server Model which provides a framework and guidelines for developing Device Classes and Servers.
- **orchestration** - Devices implement microservices [5] [6] which can be orchestrated to control simple, complex systems and systems of systems. Tango provides all the tools to configure, deploy and manage any number of Devices and Control Systems.
- **network transparency** – a fundamental design goal of Tango is to hide the network details from the developer e.g. the binary protocol is not exposed, while implementing all the necessary

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THCPL05

communication paradigms required by a control system.

- **naming service** – transparently maps a logical namespace to network addresses within a control system and across control systems.
- **configuration database** – provide a service for permanently storing configuration and descriptive data .

On top of these basic design features Tango Controls provides advanced tools for archiving and managing alarms out-of-the-box. Developers implement their own agents / microservices using the Device Server Model or re-use existing ones.

The core design features have proven sustainable over the last 17 years of Tango Controls existence. The concept of a distributed network agent and microservices has seen interest growing in the current era of IoT and cloud computing [7] [8]. This proves that the Tango architecture is well adapted to being sustainable over the coming software generations.

CORE DEVELOPMENT

During the first 10 years of Tango Controls the Software Sustainability of the Tango Controls core libraries and tools was guaranteed by the motivation of the original developers and the need to provide a full solution. Once Tango Controls matured and was considered complete by the original developers and users, the commitment and availability of the original developers reduced. New projects requiring the skills of some of the core developers motivated them to try their hand at doing something different. This made it necessary to find new core developers to ensure the Software Sustainability in the long term. The advantage of the integration of new core developers was essential to ensure the software knowledge was transferable to them as they also introduced new ideas to the core. Both of these form part of software sustainability.

One interesting observation made over the years is that as more sites adopted Tango this did not necessarily mean there were more core developers. Even if more sites means more users, third party contributions, ideas and help with bug fixes, more sites also means more maintenance as the number of use cases increases and Tango is tested in increasingly diverse environments and applications. This has the positive effect that more bugs are found and fixed but also the negative effect that it needs more core developers to maintain. To address this issue the Tango Controls Collaboration was created and the decision was taken to include industrial partners in the core development.

TANGO COLLABORATION

The Tango Controls Collaboration is made up of members of the Tango Controls community who depend on Tango for their critical operations and who need to ensure its sustainability in the long term. Members sign a collaboration contract with the home institute (currently the ESRF) in which they commit to contribute financially to the maintenance and development of Tango. Some of the partners (so-called core partners) commit additional development resources to the Tango Controls project. Decisions on how to spend the budget and which features to implement next are decided by the Steering Committee. The decisions of the Steering Committee meetings are available online [9]. The Tango Controls Collaboration started in 2016 with 8 members – ALBA, DESY, ELETTRA, ESRF, INAF, MAXIV, SOLARIS, and SOLEIL. In 2017 the SKA-O and SKA-ZA organisations joined to bring the number up to 10. The budget has enabled a large number of projects which were stuck, due to lack of available resources, to advance and even be completed e.g. complete the move of the kernel to git, maintenance of Tango LTS9, start of Tango V10, conversion of documentation to Sphinx, Device Server Catalogue, improved website, completion of PyTango 9, continuous integration for Linux and Windows, and sponsoring of a number of events like the Write-the-doc camp, streaming of last Tango Meeting in Florence etc. [10].

The collaboration contract is signed for 5 years nominally which allows the collaboration to sign contracts of up to 5 years with industrial partners. This means the industrial partners can commit to working on Tango for a longer period and helps to stabilise the partners that the collaboration works with .

In its first two years of existence the Tango Controls Collaboration has been very useful and successful in helping Tango better serve the members and community at large. The way it has been setup with a home institute (the ESRF) simplifies the financial book keeping and management. This proves that Open Source projects can profit a lot from having (some) money invested in them. The second result of the collaboration has been the successful introduction of industrial partners into the core development team. The next step is to attract industrials as members of the collaboration and new members to ensure the long term sustainability and quality of Tango Controls.

INDUSTRIAL PARTNERS

The Tango Collaboration is working with some industrial companies and small startups to implement new features in the core. The one essential characteristic of these industrial partners is that they are all members of the community and were already power users of Tango. This is a requirement to ensure that they understand the core and that the collaboration can entrust them with tasks in the core.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THCPL05

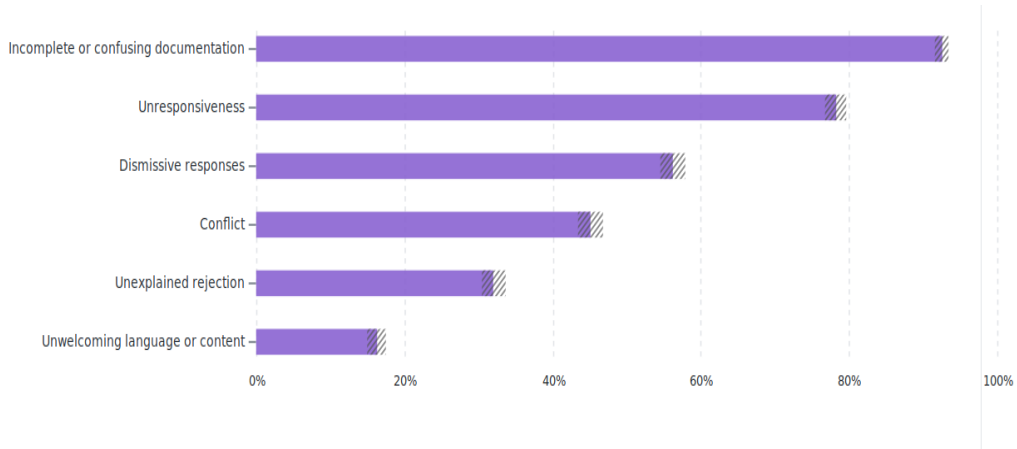

Figure 1: Problems encountered in open source (Source: opensourcesurvery.org)

So far 4 industrial partners have been financed by the Tango Controls Collaboration to contribute to the core of Tango and/or be in charge of core tasks. These partners are:

- **IK** [11] – core development in C++, Java, REST api, cmake, continuous integration, web design ...
- **3Controls** [12] – documentation, device server catalogue, TangoBox VM, ...
- **Nexeya** [13] – continuous integration of Tango and PyTango for Windows, cmake, Python Pogo templates, ...
- **STFC** [14] – implementation of missing features for PyTango 9, ...

The collaboration will continue to work with these and other industrial partners to maintain and develop Tango. It is important to note that the responsibility for maintaining Tango remains with the core members of the collaboration. The industrial partners contributions are checked and accepted by core members. The whole process depends on the Open Source Development workflow which has been setup on GitHub.

COMMUNITY RULES

Integrating new members into the kernel development team (including industrial partners) makes it necessary to ensure that the development workflow and rules are well defined and followed. The responsibility for the stability of the kernel remains with the core members of the Tango Collaboration. This guarantees the continuity and trust of the community in the development process and in the kernel. This is necessary for all sites which depend on Tango. The current workflow for kernel developments is as follows:

- discuss improvement or bug fix with at least one other kernel developer
- if agrees then create pull request (PR)
- start working on PR in local git
- assign other kernel developer to PR when finished and request a review from other kernel developer(s)
- fix or improve PR until reviewer(s) agree
- assigned kernel developer merges PR into master, bug fixes in LTS branch

Transparency is an essential part of building trust in any community. The Tango Controls community communicate via a forum, mailing list, annual general assemblies, steering committee meetings, monthly kernel developers meetings and GitHub. Discussions take place in the open and are documented so they can be followed by everyone.

GITHUB DEVELOPMENT

Git and GitHub play an important role in order to implement the above workflow. The power of git and the ease of use and high quality tools made available by GitHub have been a game changer. Issues can be created, worked on, compared, reviewed by a distributed team easily. The projects dashboard allows progress to be charted and mapped during development sprints. All Tango kernel projects (except for HDB) have been moved successfully to GitHub and can be found at https://github.com/tango-controls.

LONG TERM SUPPORT

Long Term support has been requested by most Tango sites but is also essential for industrial users who base their solutions on Tango. The main reason it did not exist up until now was the lack of resources to back port bug fixes to previous versions. The Collaboration has decided to finance the maintenance of an LTS version of Tango. For this reason Tango V9 has been declared Long Term Support and will be maintained for 5 years at least after the next stable release is made available

As part of the LTS support the missing features in PyTango 9 have been implemented by STFC.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THCPL05

CONTINUOUS INTEGRATION

The advantages of continuous integration [15] are so obvious today they are not questioned any more and are a required in almost all software projects. Tango uses Travis for continuous integration (CI) builds for Linux platforms thanks to IK and Nexeya is working on CI for Windows on AppVeyor. Binary builds are made available via https://bintray.com/tango-controls .

Windows is the platform of choice for many industrials. Before setting up CI on AppVeyor only major release of Tango were made available for the most common Windows compilers and platforms. If a user had a version of the compiler which was not supported, or wanted to debug or modify the code then it was the responsibility of the users to build Tango on Windows. This limited the number of people who could try, test or contribute to the latest versions of Tango on Windows. This was especially true for PyTango which depends on Boost.

With the setting up of CI for Tango and PyTango on AppVeyor users will soon have access to the stable and development versions of Tango and PyTango on the following platforms:

1. VC9 – 32 bits, VC9 – 64 bits
2. VC10 – 32 bits, VC10 – 64 bits
3. VC12 – 32 bits, VC12 – 64 bits
4. VC14 – 32 bits, VC14 – 64 bits

As an additional bonus for this work Nexeya has converted the build process on Windows to Cmake and builds Conda packages for PyTango. The Cmake Windows script is integrated in the standard Cmake scripts used to build Tango on Linux too. Tango now has a unified way of building on all platforms.

DOCUMENTATION

A recent survey by GitHub [16] [17] on how users perceive Open Source software found that the one topic which most users miss in Open Source is documentation. Tango documentation traditionally consisted of The Book (a large pdf file distributed with the source code release) and many smaller documents in diverse formats (web pages, Word documents, pdf etc). The Book contains a lot of information but suffers from being hard to read, maintained by a single person and not easy to contribute to. A decision was taken in 2015 to combine all the documents into a single source and make this easy to modify. The goal of this change was to write documentation like code [18]. This means following the same workflow as source code i.e. git version control, changes by pull request and continuous integration. The Sphinx format has been chosen for the documentation. It is versioned on GitHub and CI is done on Travis. The final result is available on http://tango-controls.readthedocs.io/en/latest/. As soon as any changes are merged with the main git repository the documentation is rebuilt and available online. Multiple versions are maintained for the different versions of Tango.

The reorganisation of the documentation would not have been possible without the help of 3Controls and a dedicated group of document coders who met in the French Vercors mountains for 3 days [19] (sponsored by the Collaboration). The Write-the-docs camp was a success and proved writing documentation can be as much fun as writing code! The Write-the-Doc event will be repeated on an annual basis. One goal of the next phase will be to translate the main documentation into other languages to help new communities - Russian or Chinese volunteers welcomed!

DEVICE SERVER CATALOGUE

One of the main advantages of a growing Tango community is the large number of Device Classes which exist for many different hardware and software platforms. However, it has been difficult to share these classes until now. The reasons for this are multiple and challenging. Some of them are inherent due to the fact that it is often easier and quicker to generate a new version of a class than to adapt to an existing version. Others are due to the fact that many classes are not published on the common repository, often for good reasons. Other reasons are the lack of an easy-to-use and attractive interface. In order to address some of these issues the collaboration has requested an industrial partner (3Controls) to design and build a new catalogue of Device Classes which is well integrated in the web site.

The new Device Class catalogue (http://www.tango-controls.org/resources/dsc/) has been developed as a plugin in the Django based tango-controls.org website. The catalogue is able to mine public repositories, upload metadata from private repositories and accept manual input. The catalogue offers an attractive easy to use search machine which displays the results an an easy to read manner. The catalogue offers a user rating and feedback system to help gather input from users. The next phase is to upload more classes and to manually edit the classifications as some of them have been entered incorrectly. The usefulness of the catalogue will be further improved by providing pre-built binaries of a selected number of commonly used Device Classes.

COMMERCIAL LAB SYSTEM

One of the goals of the Tango Controls collaboration was to provide industrial users with a modern toolkit for building control systems as an alternative to proprietary solutions.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THCPL05

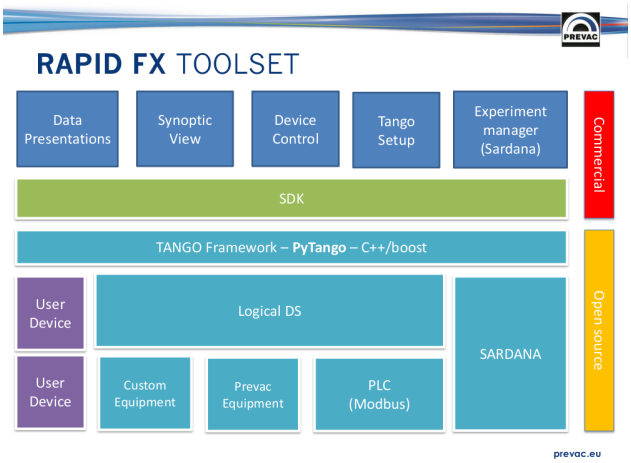

Figure 2: Prevac's Rapid FX toolset based on a Tango, Sardana and commercial tools.

Prevac [20] is an example of a company providing a Tango based lab control system. Prevac builds high and ultra-high vacuum equipment for research. They have installations in over 30 sites world wide. Prevac decided to base their latest range of software products, Rapid FX, on Tango (see Fig. 2). In order to integrate Tango into the Windows world and profit from the large .NET solutions, Prevac has built a C# binding to Tango. This allows them to use .NET for the user interface, web etc. while using Tango for the communication protocol. They have adopted PyTango and Sardana for control sequences. Their customers can write their own Device Classes in Python, C++ or Java to integrate their own equipment. This win-win situation is fruitful for both Prevac, and their customers. Prevac plans to release their Windows Tango installer to the community.

WEB APPS

The web is today's user interface platform per default. Since the development of the Tango REST api by IK Tango is now compatible with the web. Two generic solutions are the WebApp from IK and JYSE for Tango from JYSE [21] (Fig. 3).

The Tango WebApp solution is a generic replacement Jive for small Tango systems which can be deployed locally or on the cloud [22]. WebApp uses the latest web techniques for fast loading and scaling. It provides a dashboard for managing multiple windows.

JYSE for Tango is an interactive tool, entirely browser based, which allows non-programmers (as well as programmers) to very quickly develop attractive and ergonomic web displays for Tango. Users visually compose their display screens, connect them to any Tango device data exposed via the REST API, and share their URL for use on all types of devices including phones and tablets. JYSE for Tango can be either hosted in the cloud or deployed on site

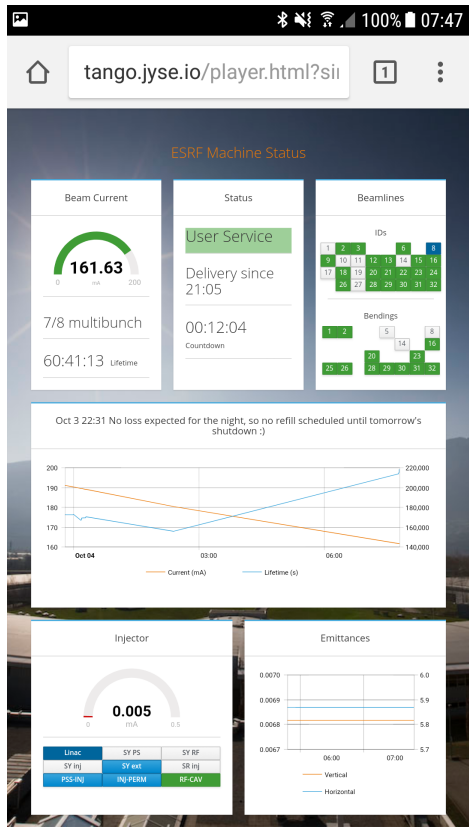

Figure 3: ESRF machine status displayed on phone with JYSE for Tango.

INDUSTRY AND OPEN SOURCE

One issue facing the adoption of Tango by industry is how to break into a market dominated by proprietary solutions which implement expensive industrial standards. The advantages of Open Source for the industrial controls world are many. Some of them are common to all Open Source solutions e.g. avoiding lock-in, access to source code, possibility to extend or modify code, and others specific to industrial controls e.g. higher security because users have full access to source code. Despite these advantages the uptake of Open Source control systems in industry is slow. Some of the reasons for this are:

- Tango Controls is a framework in the first place. Therefore adopting it to build a product specific for a use case, requires quite some development effort. This effort is sometimes a show stopper for small companies, because it is expensive and time consuming.
- For big companies usually they already have their own solution. This solution can vary from fully in-house developed to fully provided by a 3rd party integrator. Sometimes they want to migrate from their solution. In this case Tango can be a suitable choice but it lacks being well known in the industry world.
- Lack of certification is another issue. For example if Tango Controls would be certified by

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THCPL05

ISO [23] or GOST [24] in Russia, it would be much easier to "sell" it.

- Open Source licensing is not always well understood by industrial partners. Legal advisors discourage management to use open source.
- Sources availability in public repositories is considered as a potential point of malicious code injection even there are easy procedures to avoid having unnecessary updates. Responsibility is not well defined.
- Open Source products are still generally regarded as featuring low-quality, bad documentation and weak support.
- Tango Controls is coming from scientific world and is regarded as suited for academic, non-critical applications despite its maturity and being run in laboratories operating 24/7.
- Some industrials are adopting Open Source (including Tango) but not sharing their developments or publishing their results. The main reason for this is that paying for something is still considered better than getting it for free; in the industrial world, competition is aggressive and « sharing » is not the way of beating the competition; licensing is a way of making money.

We hope this will evolve in the future with the new digital economy just like Linux has ended up taking over the server market. We still think the future of Tango in the industrial market has strong potential. There are not many solutions which offer all the features of Tango Controls for building control systems in Python, C++ and Java. The market for a modern Open Source control systems toolkit is still open.

CONCLUSION

Tango has adopted industry and setup successful collaborations with a number of startup firms and larger groups to help develop and maintain Tango. Some industrial users have successfully adopted Tango. The goal of the Tango Collaboration is to continue working with industrial firms to develop and maintain Tango and to grow the industrial offerings of Tango. This is essential for the sustainability of Tango. The setting up of the Tango Controls Collaboration has been critical for creating partnerships with industrial companies.

ACKNOWLEDGEMENTS

A big thank you to the 10 sites who have joined the collaboration and helped support Tango financially and strategically – without them this paper would not exist! Thanks to Olga Merkulova for her mind map of the new documentation and designing the new web site (to be released soon) and to the participants to the write-the-docs. As always a special word of thanks to the Tango Community (past and present) who have all helped make Tango a success and a great collaboration.

REFERENCES

[1] *The ESRF's Extremely Brilliant Source - a 4th Generation Light Source*, J.M.Chaize et. al., paper FRAPL07, ICALEPCS 2017 (this conference)

[2] Software Sustainability Institute: https://www.software.ac.uk/about

[3] C.C.Venters et. al. (2014), Software sustainability: The modern tower of Babel. RE4SuSy: Third International Workshop on Requirements Engineering for Sustainable Systems; 2014.

[4] M.P.Robillard (2016), Sustainable Software Design , *Proceedings of the 2016*, 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering.

[5] S.Newman, Building Microservices: Designing Fine-Grained Systems, O Reilly 2014.

[6] Martin Fowler on Microservices, http://martinfowler.com/articles/microservices.html

[7] Krivic P., Skocir P., Kusek M., Jezic G. (2018) Micro-services as Agents in IoT Systems. Agent and Multi-Agent Systems: Technology and Applications. KESAMSTA 2017. Smart Innovation, Systems and Technologies, vol 74. Springer, Cham.

[8] A. Assik, *Potential of Agent Architectures*, https://dzone.com/articles/the-potential-of-agent-architectures

[9] Tango Controls Steering Committee: http://www.tango-controls.org/about-us/executive/

[10] *Tango Kernel Development Status*, R.Bourtembourg et. al., paper MOBPL02, ICALEPCS 2017 (this conference)

[11] IK, http://www.ingvord.ru/

[12] 3Controls, http://3-controls.com/en/home-2

[13] Nexeya, https://www.nexeyaonline.com/tango-uk-middleware-control

[14] STFC, http://www.stfc.ac.uk/

[15] Martin Fowler on *Continuous Integration*, https://martinfowler.com/articles/continuousIntegration.html

[16] Github Open Source Survey, http://opensourcesurvey.org/

[17] R. Stuart Geiger, *Summary Analysis of GitHub Open Source Survey*, arXiv:1706.02777v1

[18] A.Gentle, *Docs like Code*, https://www.docslikecode.com/

[19] Tango Write-the-Docs camp, http://www.tango-controls.org/community/events/write-docs-camp/

[20] Prevac, https://www.prevac.eu/

[21] Jyse, https://www.jyse.io/en/

[22] Tango WebApp, https://github.com/tango-controls/tango-webapp

[23] ISO standards, https://www.iso.org/home.html

[24] GOST standard in Russia, https://en.wikipedia.org/wiki/GOST