ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-MOBPL02

TANGO KERNEL DEVELOPMENT STATUS

R. Bourtembourg†, J.M. Chaize, T. Coutinho, A. Götz, V. Michel[1],
J.L. Pons, E. Taurel, P. Verdier, ESRF, Grenoble, France
G. Abeille, N. Leclercq, SOLEIL, Gif-sur-Yvette, France
S. Gara, NEXEYA SYSTEMS, La Couronne, France
P. Goryl, 3CONTROLS, Krakow, Poland
I. Khokhriakov[2], IK, Moscow, Russia
G. Mant, STFC, UK
J. Moldes, ALBA-CELLS Synchrotron, Cerdanyola del Vallès, Spain
Birgit Ploetzeneder, ELI-BEAMS, Prague, Czech Republic
[1]previoulsy at MAX IV Laboratory, Lund, Sweden
[2]also at HZG, Geesthacht, Germany

Abstract

The TANGO Controls Framework continues to improve. This paper will describe how TANGO kernel development has evolved since the last ICALEPCS conference. TANGO kernel projects source code repositories have been transferred from subversion on Sourceforge.net to git on GitHub.com. Continuous integration with Travis CI and the GitHub pull request mechanism should foster external contributions. Thanks to the TANGO collaboration contract, parts of the kernel development and documentation have been sub-contracted to companies specialized in TANGO. The involvement of the TANGO community helped to define the roadmap which will be presented in this paper and also led to the introduction of Long Term Support versions. The paper will present how the kernel is evolving to support pluggable protocols - the main new feature of the next major version of TANGO.

ROADMAP

TANGO is a mature and reliable toolkit to build distributed control systems for installations which need to run for the next 10 to 20 years. For this reason TANGO needs a roadmap which will ensure its future evolution and stay a good choice for the coming years. At the TANGO meeting in May 2015 held at Solaris in Krakow (Poland) the community gave their input on the essential features of the current roadmap. Input was provided via email and discussed during an interactive session. The results presented in [1], [2] and [3] are summarised here:

(1) Improve Documentation, (2) Move to Git, (3) Remove CORBA completely, (4) Grow the community, (5) REST API, (6) Web browser application, (7) Secure encryption, (8) Database performance, (9) Device class Marketplace, (10) Long Term Support, (11) TANGO Virtual Machine, (12) Auto-generate Unit tests, (13) SysML support, (14) Replace Boost.Python.

This paper will go through each point of the roadmap and present the current status (as of October 2017).

IMPROVE DOCUMENTATION

The highest priority point as defined in the roadmap in 2015 was to re-factor and consolidate the documentation and to write a cookbook of recipes and concepts. Until recently, the main TANGO documentation was *The TANGO Book*, a large pdf file of more than 250 pages and many other documents in various formats. *The Book* contains a lot of precious information but was hard to read. It was decided to combine all the available documentation into a single source in Sphinx format on readthedocs. In the spirit of Write-the-docs [4] a documentation camp was held in the Grenoble area in May 2017. The camp lasted 3 days with 2 days in a house rented especially for the event and located in a quiet mountain village. The event was dedicated to working on reviewing the new documentation available on read-the-docs [5] and identifying and writing missing documentation for TANGO.

The meeting was sponsored by the TANGO Controls Collaboration. Travel and accommodation were paid for by the Collaboration. 11 enthusiastic documentation writers from 7 different institutes/companies participated (see Fig. 1).

Figure 1: The write-the-docs team at work.

Here is a brief summary of the achievements: the documentation has been re-organized to be a coherent whole, almost all existing docs have been converted to sphinx, new diagrams have been produced and integrated. Some old ones have been updated. New Getting Started

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-MOBPL02

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

sections have been added. The TANGO documentation now has a single location where it is maintained and can be easily extended. It is now much easier for beginners, intermediate and experienced users to find the information they need. The documentation is now built automatically and is easy to contribute to. All you need is a Github account. More details and photos can be found in [6].

MOVE TO GIT

TANGO kernel source code has been hosted for many years on Sourceforge.net [7] and was using subversion as revision control system. It was decided to move the source code to git on github.com. A new Github organization named tango-controls [8] has been created. At the time of ICALEPCS 2017 conference, all the main TANGO kernel projects except HDB have been migrated to git on Github. *tango-cs* Sourceforge tickets have been semi-automatically migrated to Github issues in their corresponding git repositories.

Githubs web interface and the Pull Request mechanism as described in [9] should foster external contributions and help improving code quality by simplifying the code review process and collaboration.

This move to Github is a huge step forward for TANGO kernel development because it enables access to many third party tools (Github apps/services) for free. For instance the TANGO kernel C++ library repository named cppTango [10] is now using Travis CI [11] and soon AppVeyor as well [12] to build the software and run automated tests on Linux and Windows platforms as soon as there is a new commit or pull request. Tests code coverage is computed by Coveralls [13] and code quality is analysed with Codacy [14]. The documentation can also be automatically generated and deployed on ReadTheDocs.

REMOVE CORBA COMPLETELY

The third most requested feature from the roadmap was the possibility to replace CORBA as synchronous communication protocol and remove all dependencies on CORBA with a pluggable protocol layer. CORBA is currently doing a great job but it is no longer actively developed and can be complex to install.

Proof of concept refactoring work has been done by IK [15] on cppTango to isolate CORBA code, implement *DevVarDoubleArray* type using an architecture allowing plugins and replace omni_thread with C++ 11 threads.

CORBA notifd events support related code has been removed in TANGO V10 development branch. Only TANGO events using ZeroMQ will be supported in TANGO V10. TANGO events based on ZeroMQ have been supported since TANGO V8.

TANGO Controls is being used in some big scientific instruments like synchrotrons or big telescopes to controls hundreds or thousands of devices. Runtime compatibility with previous versions has always been a major requirement for TANGO kernel development. This is essential in order to ease the upgrade to newer versions of TANGO by allowing a progressive upgrade of the control system. Runtime compatibility with TANGO V9 will be preserved in TANGO V10, except for the TANGO events using CORBA notifd. Source code compatibility with TANGO V9 will be preserved as much as possible. Continuous Integration will be improved in order to be able to test automatically the communications between different TANGO versions.

GROW THE COMMUNITY

Since the previous ICALEPCS conference, the TANGO collaboration is now governed by a steering committee [16], which is composed of one representative per institute who has signed the TANGO Controls Collaboration Contract. All steering committee members commit to financing the development of TANGO and participate in the decision making process. Two types of membership are recognised – core members and contributing members. All members contribute financially to maintaining TANGO. Core members also contribute code to the core. The new organisation does not change the free open source nature of TANGO i.e. TANGO remains freely available and open source for everyone. The new organisation ensures the sustainability of TANGO by financing resources to work on tasks of common interest like the roadmap.

The core members are ALBA, ELETTRA, ESRF and SOLEIL. The contributing members are DESY, INAF, MAX-IV, SKA Office, SKA South Africa and SOLARIS. Other institutes have expressed an interest in signing the TANGO Controls Collaboration Contract in the near future.

The 30th TANGO collaboration meeting organised by the French National Aerospace Lab ONERA took place in June 2016 at Toulouse, France. One year later, INAF organised the 31st TANGO collaboration meeting which took place at Florence, Italy (see Fig. 2).

Figure 2: Coffee break during 31st TANGO collaboration meeting in Florence.

For the first time, a TANGO Controls Users Meeting Russia [17] took place in Moscow, on May, 18th 2017. The

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-MOBPL02

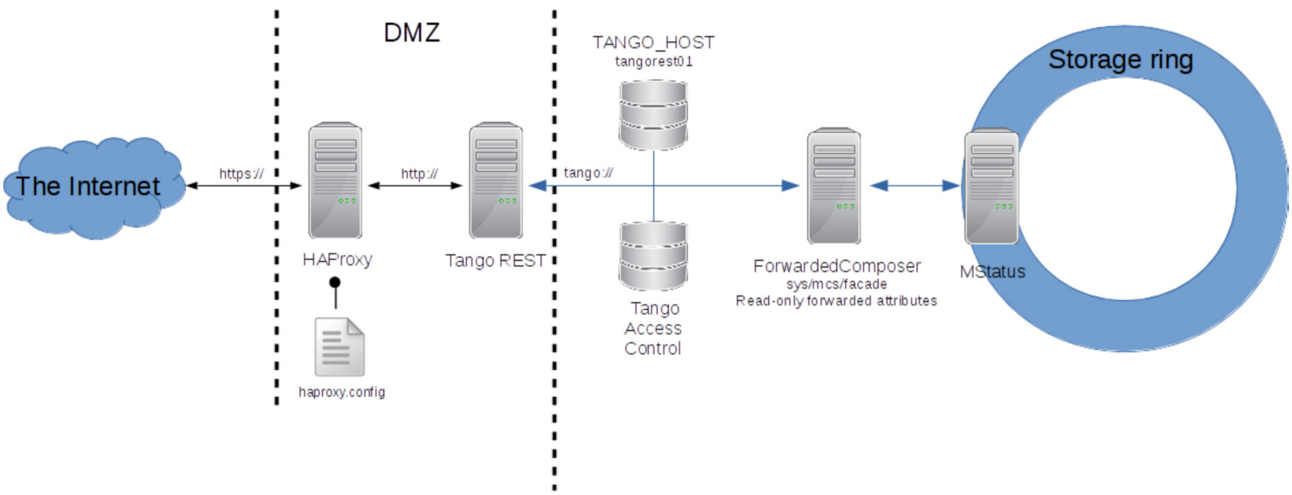

Figure 3: TANGO REST Server deployment at the ESRF.

meeting was held in Russian language and was sponsored by the TANGO Controls collaboration. The main goals of the meeting were: (1) get acquainted with TANGO Users in Russia, (2) share experience in using TANGO, (3) share feature requests, bugs, wishes for the next TANGO releases, (4) start building a TANGO community in Russia. The meeting gathered 21 participants from 8 different institutes, companies or universities.

REST API

Discussions on the TANGO Controls forum [18] helped to define a standard TANGO REST API. The API specification is documented on readthedocs [19, 20]. A Github repository [21] has been created by IK and provides a Java reference implementation which can be used for server/client development using Java. Several TANGO REST device servers have been implemented by the TANGO community: (1) mTangoREST.server developed by IK [22], implemented in Java and can run as servlet and (2) RestDS device server developed by JINR [23], written in C++ and supporting basic http authentication.

Figure 3 is showing a deployment scheme of a REST Server at the ESRF. In this installation, REST API exports read-only forwarded attributes and is accessible via secured http connection. Every request passes the HAProxy [24] configured to use the https protocol for secure connections. On its backend, HAproxy speaks with the TANGO REST server, which in turn can access only one TANGO control system (TANGO_HOST), where a device of class ForwardComposer [25] is defined. This device provides read only access to the MStatus TANGO device with status information about the storage ring at the ESRF. In addition TANGO REST API can be integrated with authentication and authorisation services like kerberos. Finally the TANGO REST API implementation uses the TANGO Access Control to validate every request made from the Internet.

The TANGO REST API specification is still a work in progress, and the next release candidates will take advantage of the HTTP/2 Server Push feature [26] to enable the possibility to push TANGO events to web clients.

WEB BROWSER APPLICATION

A generic web application for browsing and monitoring TANGO devices (See Fig. 4) is being developed by Helmholtz-Zentrum Geesthacht. It is using the TANGO REST API and is available on tango-webapp Github repository [27]. A demo has been deployed on Amazon Cloud. The link, login and password to play with this demo are described on tango-webapp README [27].

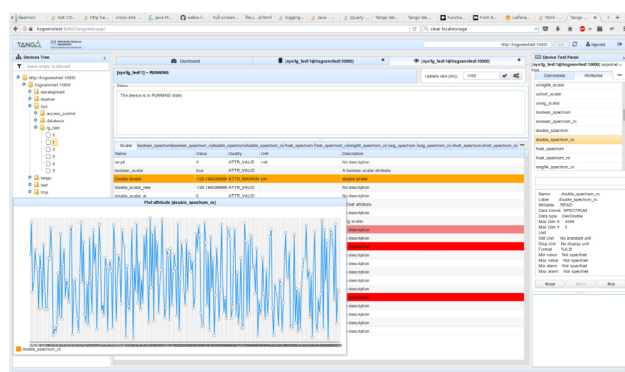

Figure 4: TANGO WebApp.

SECURE ENCRYPTION

This idea of the roadmap feature request #7 was to provide a secure way of accessing TANGO devices using a secure encrypted protocol. This can be achieved by using standard technologies (e.g.: using HAProxy [24] configured to use the https protocol) and the TANGO REST API. Setting up a VPN network can also enable access to a control system from public networks in a secure way.

DATABASE PERFORMANCE

Database performance issues related to memorised attributes history writes has been solved in 2015. This was

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-MOBPL02

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

achieved by disabling support for the history feature which is not compatible with memorised attributes which change rapidly (tens of Hz).

DEVICE CLASS MARKETPLACE

To increase the sharing of device server among the TANGO community, a Device Classes Catalogue [28] (See Fig. 5) developed by 3Controls [29] has been implemented and is available on the TANGO Controls official website. This work has been sub-contracted and has been financed by the TANGO Collaboration Contract.

Now TANGO device server programmers can advertise the device servers they consider might be useful for the TANGO community. In the past, the only known location where TANGO device server programmers could share device servers was on tango-ds Sourceforge [30], a subversion repository. With the Device Classes Catalogue, users can keep the device servers source code on their own repositories, which could be a git, subversion, CVS repositories for instance. Even if the repository is not public, the TANGO community will now be aware that this device class exists and will have a way to contact the original developer/maintainer.

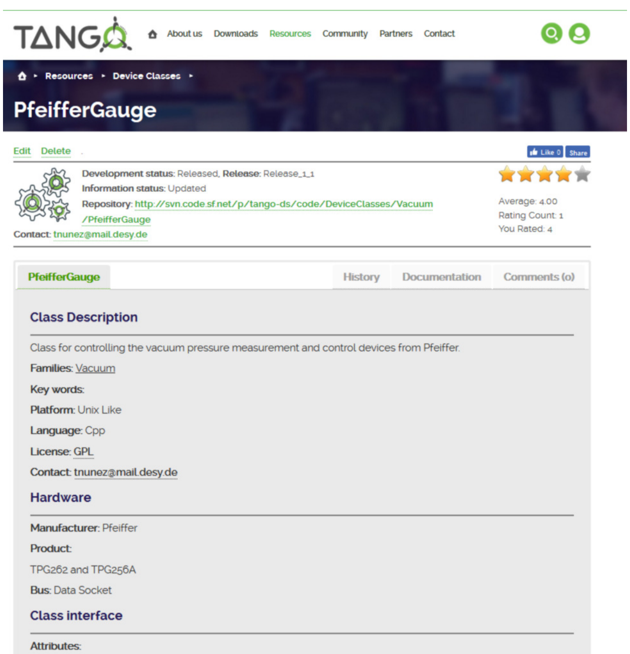

Figure 5: TANGO Device Classes Catalogue.

LONG TERM SUPPORT

TANGO V9 is the first TANGO release which will benefit from a long term support. It will benefit from bug fixes and simple feature patches backported from new TANGO major releases. It was decided to fix the length of the support to 5 years, starting from the moment when the next TANGO major stable release is made available. Previous TANGO versions did not benefit from Long Term Support because of a lack of resources. The maintenance of the LTS versions will be financed by the collaboration.

In cppTango, a branch named tango-9-lts has been created and is dedicated to this long term support. TANGO V10 development is done in the master branch. Similar branches will be created for Java and Python TANGO kernel repositories as soon as TANGO V10 development is started for these languages.

TANGO VIRTUAL MACHINE

The TANGO virtual machine, also known as the TANGO Box has been updated by 3Controls since the latest ICALEPCS with the latest stable version of TANGO and all its tools. It is available for download on the TANGO Controls official website [31].

AUTO-GENERATE UNIT TESTS

Roadmap feature request #12 is to generate Unit Tests automatically from POGO, the TANGO class code generator. Some work has been done for TANGO classes generated in Python HL (using the PyTango high level API) for which it is now possible to generate a Python package with a directory structure. Among other directories, a directory named *test* is generated with a python file dedicated to unit tests. One method per attribute defined is automatically generated in this file and the user just need to fill the gaps to implement good unit tests for his class. The Indo-SA SKA collaboration developed the simlib library [32] which is able to simulate in a basic and complex way a TANGO device. The basic simulation can be created easily from the TANGO interface definition xmi file generated by POGO.

JTango Maven archetype generates simple unit tests for Java TANGO projects [33].

SYSML SUPPORT

Roadmap feature request #13 was to add support for using SysML to specify device servers. The DSEE (Domain Specific Engineering Environment) tool [34] is able to automatically generate test cases and a basic TANGO device in Java from the M&C ML modelling language [35]. It is also able to generate SysML views of the state machine.

REPLACE BOOST.PYTHON

Boost.Python is currently used as the interoperable layer between PyTango and the C++ TANGO kernel. Since Boost.Python is not actively developed anymore and some bugs are due to this layer, a decision was made to investigate a replacement solution. PyBind11 [36] is currently being investigated by STFC [37] as an alternative solution to Boost.Python. The first results look promising.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-MOBPL02

TANGO C++ KERNEL STATUS

Part of the TANGO C++ kernel development is being subcontracted to IK for the TANGO collaboration. The migration to the cppTango Github repository [10] was a good opportunity to refactor TANGO C++ library kernel code. The Log4tango library has been merged into the TANGO C++ library. Autotools have been replaced by CMake to compile and install the library. This should simplify the compilation and installation process on Windows platform. Continuous integration with Travis CI has been set up and is now using docker containers [38, 39]. In the past, continuous integration was done with Jenkins on machines maintained by the ESRF, with restricted access. With Travis CI, the results of the builds and automated tests are publicly available. Using Travis CI and docker containers makes it easy to add new operating systems to the continuous integration. Nexeya [40] is currently working on setting up the continuous integration with AppVeyor for the Windows platform. Some work has been done to generate Debian C++ Library development packages, which can be deployed automatically on JFrog Bintray [41]. In the future, RPM or Yum packages could be deployed in the same way on Bintray. The first TANGO V10 development version is already available on Bintray. V10.0.1 implements the possibility to associate labels to Tango::DevEnum parameters passed into TANGO commands. V10.0.2 will add support for DevPipeBlob command parameters, allowing to pass complex parameters to TANGO commands. The TANGO event subscription mechanism is currently being refactored in order to improve the support for special network configurations.

In the coming years, besides TANGO 9 Long Term Support, the main activity will be to refactor the source code to make it easier to maintain, improve the code quality and implement the pluggable protocol feature.

TANGO JAVA KERNEL STATUS

TANGO Java kernel source code has been migrated to Github [42].

TANGO Java kernel software is now fully integrated with Maven and can be deployed automatically on Bintray Maven repository. In a near future it will also be available on Maven central. A JTango Maven archetype [33] has been created to help generate the skeleton for JTango projects. This archetype is already available on Maven central repository [43].

The JMX monitoring feature has been implemented in order to be able to monitor TANGO specific metrics through JMX.

The main activities in the coming years will consist in trying to remove the dependency to Jacorb and rely instead directly on the standard Java ORB implementation. Refactoring activities will take place in order to merge modules providing similar functionalities. Continuous integration with Travis CI and code quality checks with Github 3rd party tools have been set up. The results of these checks are accessible as Github badges on the project Readme file on Github. The implementation of pluggable protocols will then become a major item on the roadmap.

PYTANGO KERNEL STATUS

PyTango 9.2.2 has been released in September 2017. The main features of this new release are mainly the support of the TANGO V9 new features which were already implemented in the C++ and Java TANGO kernel: pipes, dynamic commands, forwarded attributes, mandatory properties and device interface change events. These new features have been implemented by STFC with some help from the PyTango developers. This work has been financed by the TANGO Collaboration Contract. PyTango code quality has been checked and improved with pylint and flake8.

PyTango 9.2.1 came with 2 additional modules for unit-testing, these provide a context to run and access a TANGO device without the database and to provide generic pytest fixtures [44] for TANGO-related unit-testing.

PyTango source code was already on Github. The repository has been transferred to the tango-controls Github organisation.

LABVIEW BINDING

The TANGO binding for LabVIEW [45] used to be a pure client platform. It now allows any LabVIEW application to turn into a TANGO device. The first release supports most of the TANGO 9 features. *DevEncoded* and *Pipe* data types are currently not supported but the provided features are sufficient to start writing advanced TANGO devices.

Special care has been taken regarding reliability and performances. The data exchanges between the two worlds are based on code inherited from the binding client API, which is known to be fast and stable.

The laser community (e.g.: ELI-Beamlines, Apollon Saclay) is using the binding server API.

TOOLS

The main TANGO tools written in Java (Jive, Astor, Pogo, Atk, AtkPanel) have been converted to Maven and are available on Bintray.

Sardana [46], Taurus [47], TANGO Alarm Systems [48], HDB++ [49], Vacca [50], Cumbia [51] and many other interesting TANGO related projects have evolved and are available on Github.

CONCLUSION

TANGO Controls is a growing collaboration for building distributed controls system. The collaboration contract has already proved it can ensure the sustainability of TANGO for the next years. It has had a positive impact on the TANGO kernel development by giving the possibility to subcontract some important tasks to companies specialized in TANGO development [52]. The

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-MOBPL02

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

outcome for the community is better documentation, better code quality, better tools and better support with the introduction of the first Long Term Support TANGO version.

As explained in this paper many of the features on the roadmap have been implemented, the next step is to consult the community again to identify missing features in the latest major release and to update the roadmap.

ACKNOWLEDGEMENT

The authors would like to thank the TANGO Controls community for their ideas, input and innovative use of TANGO, the TANGO write-the-docs participants for their enthusiasm and help in improving the TANGO documentation, the developers involved in the TANGO Controls kernel for their enthusiasm and hard work, Stuart James for his help in preparing this paper and Frédéric Picca for bringing TANGO into official Debian package repositories.

REFERENCES

[1] TANGO Roadmap presentation at TANGO meeting 2015, http://ftp.esrf.eu/pub/cs/tango/meetings/2015_may_solaris/Andy_Gotz_Roadmap_presentation.pdf

[2] TANGO Controls Roadmap, http://www.tango-controls.org/community/roadmap

[3] A. Götz *et al.*, "The Tango Controls collaboration in 2015", in *Proc. 15th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'15)*, Melbourne, Australia, Oct. 2015, paper WEA3O01, pp. 585-588.

[4] Write the Docs, http://www.writethedocs.org

[5] TANGO Controls Documentation, http://tango-controls.readthedocs.io/en/latest/contents.html

[6] First TANGO write-the-doc camp, http://www.tango-controls.org/community/events/write-docs-camp

[7] Old TANGO Controls kernel source code repository, https://sourceforge.net/projects/tango-cs

[8] TANGO Controls Github organization, https://github.com/tango-controls

[9] Git Workflow Explained – A Step-by-Step Guide, https://medium.com/@swinkler/git-workflow-explained-a-step-by-step-guide-83c1c9247f03

[10] TANGO Controls C++ Library Github Repository, https://github.com/tango-controls/cppTango

[11] TANGO Controls Travis CI build status, https://travis-ci.org/tango-controls

[12] AppVeyor Continuous Delivery service for Windows, https://www.appveyor.com

[13] Coveralls, https://coveralls.io

[14] cppTango codacy dashboard, https://www.codacy.com/app/tango-controls/cppTango/dashboard

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

[16] TANGO Controls Steering Comittee, http://www.tango-controls.org/about-us/executive

[17] TANGO Controls Users Meeting Russia, http://www.tango-controls.org/community/events/tango-users-meeting-russia

[18] TANGO Controls Forum: Defining a standard TANGO REST API, http://www.tango-controls.org/community/forum/c/general/development/tango-feature-request-4-defining-a-standard-tango-rest-api

[19] TANGO REST API specification, http://tango-rest-api.readthedocs.io/en/latest

[20] TANGO Controls REST API documentation, http://tango-controls.readthedocs.io/en/latest/development/advanced/rest-api.html

[21] TANGO REST API Github repository, https://github.com/tango-controls/rest-api

[22] mTangoREST.server repository, https://bitbucket.org/hzgwpn/mtangorest.server/wiki/Home

[23] RestDS device server repository, http://tangodevel.jinr.ru/git/tango/web/RestDS

[24] HAProxy, http://www.haproxy.org

[25] ForwardedComposer device server repository, https://github.com/tango-controls/ForwardedComposer

[26] HTTP/2 Server Push wikipedia article, https://en.wikipedia.org/wiki/HTTP/2_Server_Push

[27] tango-webapp repository, https://github.com/tango-controls/tango-webapp

[28] TANGO Classes Catalogue, http:// www.tango-controls.org/resources/ dsc

[29] 3Controls, http://3-controls.com

[30] tango-ds project on SourceForge.net, http://sourceforge.net/projects/tango-ds

[31] TANGO Box virtual machine, http://www.tango-controls.org/ downloads/ virtual-machine

[32] simlib library, https://github.com/ska-sa/tango-simlib

[33] jtango-maven-archetype, https://github.com/tango-controls/jtango-maven-archetype

[34] N. Marais *et al.*, "Control System Simulation Using DSEE High Level Instrument Interface and Behavioural Description", presented at the 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17), Barcelona, Spain, Oct. 2017, paper TUDPL03, this conference.

[35] Puneet Patwari *et al.,* "M&C ML: A modeling language for monitoring and control systems", *Fusion Engineering and Design*, Volume 112, 15th November 2016, pp. 761-765, https://doi.org/10.1016/j.fusengdes.2016.05.024

[36] pybind11 on readthedocs, https://pybind11.readthedocs.io/en/stable/index.html

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

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-MOBPL02

[38] Dockerfile for TANGO control system automated build, https://github.com/tango-controls/tango-cs-docker

[39] Docker image with MySQL and TANGO DB schema, https://github.com/tango-controls/docker-mysql

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

[41] TANGO Controls on JFrog Bintray, https://bintray.com/tango-controls

[42] TANGO Controls JTango Github Repository, https://github.com/tango-controls/JTango

[43] JTango Maven archetype on Maven Central, https://search.maven.org/#artifactdetails%7Corg.tango-controls%7Cjtango-maven-archetype%7C1.1%7Cmaven-archetype

[44] pytest fixtures, http://docs.pytest.org/en/latest/fixture.html

[45] LabVIEW binding for TANGO, https://github.com/tango-controls/labview-binding

[46] Z. Reszela, "Sardana – Scientific SCADA suite", presented at the 16th Int. Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17), Barcelona, Spain, Oct. 2017, paper FRBPL08, this conference.

[47] C. Pascual-Izarra *et al.*, "Taurus Big & Small: From Particle Accelerators to Desktop Labs", presented at the 16th Int. Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17), Barcelona, Spain, Oct. 2017, paper TUBPL02, this conference.

[48] S. Rubio-Manrique *et al.*, "Panic and the evolution of Tango Alarm Handlers", presented at the 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17), Barcelona, Spain, Oct. 2017, paper TUBPL03, this conference.

[49] L. Pivetta *et al.*, "New Developments for HDB++ TANGO Archiving System", presented at the 16th Int. Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17), Barcelona, Spain, Oct. 2017, paper TUPHA166, this conference.

[50] VACCA, a control system browser for TANGO Controls, https://github.com/tango-controls/VACCA

[51] G. Strangolino, "Cumbia: A New Library for Multi-Threaded Application Design and Implementation", presented at the 16th Int.Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17), Barcelona, Spain, Oct. 2017, paper TUPHA174, this conference.

[52] A. Götz *et al.*, "TANGO Heads for Industry", presented at the 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17), Barcelona, Spain, Oct. 2017, paper THCPL05, this conference.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.