ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-TUBL03

Tango CONTROLS RFCs

V. Hardion, MAXIV Sweden, Lund, Sweden
A. Götz, R. Bourtembourg, ESRF, Grenoble, France
S.Blanch-Torné, ALBA-CELLS Synchrotron, Cerdanyola del Vallès, Spain
L. Pivetta, Elettra-Sincrotrone Trieste S.C.p.A., Basovizza, Italy
P.P. Goryl, M. Liszcz, S2Innovation, Kraków, Poland

Abstract

In 2019, the Tango Controls Collaboration decided to write down a formal specification of the existing Tango Controls protocol as Requests For Comments (RFC). The work resulted in a Markdown-formatted specification rendered in HTML and PDF on Readthedocs.io. The specification is already used as a reference during Tango Controls source code maintenance and for prototyping a new implementation. All collaborating institutes and several companies were involved in the work. In addition to providing the reference, the effort brought the Community more value: review and clarification of concepts and their implementation in the core libraries in C++, Java and Python. This paper summarizes the results, provides technical and organizational details about writing the RFCs for the existing protocol and presents the impact and benefits on future maintenance and development of Tango Controls.

INTRODUCTION

The TANGO control system is a device-oriented controls toolkit for controlling any kind of hardware or software and building SCADA systems.

The first version of the Tango Controls was designed and developed more than 20 years ago [1]. Since then, it has evolved to follow technology progress and needs of new features and improve its quality. There are still some technical challenges that relate to the legacy of the source code and dependency.

One of the challenges is concerning the heart of TANGO, which use CORBA for all the client/server communication. This open architecture allows distributed objects to communicate with each other, which is a perfect match for a scientific and heterogeneous control system like Tango, considering each hardware as an object.

From its date of creation, CORBA has seen different support from big names in the industry, even being part of the standard library of programming languages like Java. This architecture was the seed of many other types of architecture like web services (JBOSS).

Motivation

Nowadays, CORBA is still used by very specific domains but not maintained to face the evolution of computer science. In 2013 the Tango Community mentioned for the first time a study for the replacement of CORBA.

The implementation reference of Tango written in C++ shows a real entanglement inside the source code, in which OmniORB, the C++ CORBA library, is leaking in all the public API. Removing CORBA would mean refactoring the entire Tango C++ library by abstracting at many different levels.

Backward compatibility in Tango is essential, and the idea to replace the library would generate a lot of uncertainty regarding the fundamentals of Tango. Having a precise specification, an idea that emerged during the 2018 Tango kernel meeting in Krakow and lacking before the RFC project, would have been the best way to remove this risk.

Implementation Agnostic The Community works towards improving the maintainability of Tango Controls. The Collaboration proactively makes the framework immune to obsolesce of libraries and technologies (e.g. CORBA) it is based on. It is expected it may require rewriting all source code-base for a particular language.

Knowledge sharing Another factor to consider is the retirement of the initial Tango Developers. There is a risk of losing a deep knowledge of the Tango protocol implementation. Before the RFCs, the tango library source code was, in fact, the only specification. This could lead to losing the compatibility between versions of the Tango Controls or between libraries for different programming languages after bug fixes or features' implementations.

Compatibility The Community has found that formal documentation of the protocol and Tango Controls concepts is needed [2]. This assures that maintenance and development will not break compatibility.

WRITING RFCS

Writing proper specifications, understandable by software developers and not influenced by the implementation, was not an exercise that the Tango community used to do.

The model of the specification was inspired by the process established by ZMQ [3] which is one of the protocols used by Tango with an open specification and very well documented process called Request For Comments (RFC).

After a presentation at the 2019 Tango Meeting (DESY, Hamburg 2019), it was decided to start writing the specification with the involvement of all major institutes and companies that use Tango Controls.

The Process

The Tango RFC process is a clone of the ZMQ RFC one with a minor adaption to the Tango organisation. The scope

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-TUBL03

of this RFC was to collect the existing specifications of the Tango V9 in regards to the concepts and model, protocol behaviour and conventions.

The objective is to describe the expected behaviour when an operator interacts with the model of Tango and also to describe the communication and processes which allow two different implementations of Tango to communicate between each other. Contrary to ZMQ, the API is not part of the scope of specification for two reasons. First, the current C++ implementation uses CORBA object, which would make the specification very tight to the implementation. Secondly, the different implementations use specifics of the language implemented with, i.e. High-Level Device Servers, although they are trying hard to respect a similar API.

In the future, it would be interesting to fix the API into specification. But for practicalities, it was better to focus on the abstraction formed by the Tango Model and to test this new way of working.

The process of writing the specification is based on C4 and COSS, which are very similar to how the software developers work in a git environment.

C4 which stand for Collective Code Construction Contract [4] is inspired by the Github-flow [5] but applied to documentation. This way of working ensures of the transparency of the edition by keeping track of:

- changes expected by the authors to the existing specification (logging issue),
- the different phases of building the new specification,
- the comments and the acceptance of the editors

This process allows anyone of the Tango Community to participate to the edition whatever the degree of membership. Only the editors roles was restricted to the representative of the Tango Consortium [6] to ensure a review from an experienced person.

COSS stands for Consensus-Oriented Specification System [7] is a lightweight process which describe the lifecycle of the specification. It aims at achieving small specification which are not necessary fully complete although accepted as the most useful description of the moment. Consensus-oriented means that all the participant have to find a common ground before to be validated. Further detailed specification can be achieved by iteration.

In the case of the Tango community, the knowledge has been spread over tens of different institutes which often use Tango their own way with their own semantic. COSS is important for the Tango Specification as it concentrates the point of view of many stakeholders by consensus. The lifecycle also describes what part of the specification is obsolete, also by consensus. The Tango v10 project will probably introduce deprecated feature which should be clearly stated. The complete set of state described in the COSS specification allows to cover the all lifecycle (Fig. 1).

Tools

Creating a sustainable technical specification requires good tooling support. The Tango Control RFCs are the result of a collective effort of numerous contributors. The authoring process requires collaboration, involves many discussions, and ends with a peer review of the resulting document. These factors led to the adoption of the *Docs as Code* [8] philosophy and raised a need for a full-fledged DevOps platform that facilitates planning and tracking the work, provides a version control system, allows for code reviews and enables integration with external services to publish the documents. GitHub was selected as a platform offering all the required features. After migration in 2021 [9], all RFC development takes place on GitLab [10]. The RFC documents are published on the Read the Docs platform [11].

The RFC specifications are drafted using the Markdown markup language [12]. Markdown format was selected because of its close integration with the GitHub (and now GitLab) platform. The documents are stored in human-readable plain text files which makes them suitable for putting in a version control system. GitLab automatically renders Markdown as HTML which allows one to preview the modification during the development and provides instant feedback on how will the final document look like. The RFCs are taking advantage of the extensions offered by GitLab Flavored Markdown, including the *front matter* [13] which is used for embedding metadata like editor's name or document's status.

Some aspects of the Tango Controls specification like naming schemas must describe complex textual patterns. A well-defined formal notation is required to precisely describe the structure of such patterns. The RFC authors adopted the ABNF language [14] for this purpose. Listing 1 shows an excerpt from the Device Property specification using ABNF to specify the Property name.

Listing 1: Device Property name specification.

```
alphanum = ALPHA / DIGIT
underscore = %x5F
device_property_name =
    1*1ALPHA
    0*254(alphanum / underscore)
```

THE RESULT

Documents' Structure and Organisation

The project provides a template to prepare an RFC about a given part of the specification. Each topic to specify has its own document. Always starts with some metadata to help automatic processing followed by a short preamble that focus on the licensing of this documentation and the use of the ietf RFC2113. It is important then to collect the goals of the document and some user cases to describe what it is being to be specified.

The specification itself is the most agnostic as possible to the current implementations. The focus stay in describe

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-TUBL03

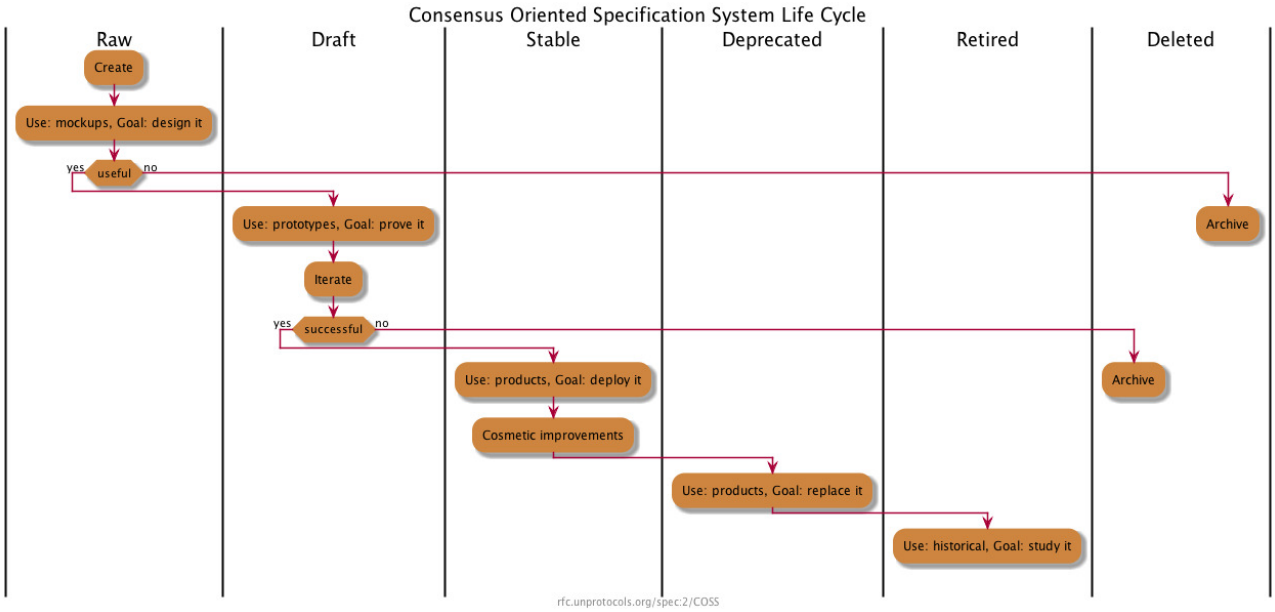

Figure 1: The complete COSS lifecycle.

and define the best way possible how things shall be made for interoperability and compatibility but giving enough freedom to avoid to be a rigid corset.

The topics in the tango-rfc are [15]:

1.Tango Control System	10.Request-Reply protocol
2.Device Object model	11.Request-Reply with CORBA
3.Command model	12.Publisher-Subscriber protocol
4.Attribute model	13.Publisher-Subscriber with ∅MQ
5.Property model	14.Logging service
6.Database system	15.Dynamic Attrs & Cmds
7.Pipe model	16.Cache system
8.Server model	17.Memorized Attributes
9.Data types	18.Authorization system

Publication

The RFCs are published on the ReadTheDocs.io service: `https://tango-controls.readthedocs.io/projects/rfc/en/latest/`. It is compiled from the MarkDown sources with use of *Sphinx* [16] and *myst-pareser* [17] library and renderd to HTML and PDF formats.

The ReadtTheDocs supports versioning. The service can be configured to use Git tags and branches as sources for different versions of the documentation. This feature allows to relate the RFCs to a specific version of Tango Controls. The versioning is already exploited to pre-release RFC-10 *Request-Reply protocol* specification [18] which is well advanced in writing but still in-progress.

USE OF THE RFCS

Added Value

Writing formal specifications for the existing Tango 9 software had some interesting side benefits.

Trying to describe some Tango 9 features revealed some inconsistencies in the way they are currently implemented in the different languages. For instance, some code generated by POGO [19], the Tango code generator, is not always behaving the same way when generated in the different supported languages. The possibility to override a device property with a class property is currently not implemented the same way in the different languages [20]. This will be improved in a future version of POGO.

This exercise helped to spread the knowledge among the involved developers on some not so well known Tango features and also pointed out the parts of the documentation which required some improvements.

This work triggered as well interesting discussions to agree on a common vocabulary because different words were used in different institutes to name the same Tango concepts. For instance, the administration device is sometimes named “Management Device”, “Admin Device”, “DServer” or “Device Server” in different institutes [21].

Finally, this activity was a great opportunity to discuss new features [22], backwards compatibility of the future versions and to propose some improvements at the concepts level.

Future Use

The RFC will play an essential role in the future development of a new major version of Tango. The model of Tango has proven to be a powerful concept for the last 20 years. Many existing and new projects have successfully

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-TUBL03

based their control system Tango and therefore need to ensure that new versions of Tango provide the same or compatible features. The RFCs describe the behaviour of Tango as it is today seen by clients and servers. The RFCs open the door to a new development of the Tango core with a new code base while ensuring that the features of Tango which are provided by the current LTS version (V9) are maintained. The RFCs can be seen as a specification for future versions of Tango. Having a clear specification agreed by all means new implementations are not required to implement all features of a full Tango-based system. Possible implementations could be made for only the server part or even micro-servers which implement only events for example. This approach allows a step-by-step approach to testing new ideas without taking a large risk that the new developments do not satisfy the needs of a typical Tango system.

The RFCs are also a guarantee that even if key members of the kernel team leave the behaviour of Tango has been captured and documented. This gives another guarantee that in the future of Tango could be entirely redeveloped without taking too much risk for existing installations even if source compatibility is not guaranteed.

CONCLUSIONS

The most important Tango Controls concepts and protocol behaviour are formally documented now. The RFCs are published on ReadTheDocs: `https://tango-controls.readthedocs.io/projects/rfc/en/latest/`. These documents have status *draft*. This means one can use them as a specification for prototype development.

Final work is in progress for RFC-10/The Request-Reply Protocol [18]. The RFC-13/Publisher-Subscriber protocol implementation with ZMQ is waiting for review [23]. Three other RFCs:

- RFC-11/The Request-Reply protocol - CORBA implementation,
- RFC-17/Memorised attribute service,
- RFC-18/Authorisation system

are planned to be written if need.

Cache system specification is now part of RFC-10/Request-Reply, and it is not yet decided if it will be moved to RFC-16.

The Tango Community plans to organise a workshop for the final edition of the RFCs and plan the next steps.

The RFCs will be further developed to follow new or changed features of Tango Controls. As the Tango Community plans to build a prototype of Tango V10, the RFCs will be used and verified for completeness and any ambiguity. This will also trigger either supplementation or corrections of the specifications.

Following COSS, after verification by usage and when a new version of Tango Controls is ready, the documents will be marked as stable.

ACKNOWLEDGEMENTS

The authors acknowledge the support of the Tango Controls Collaboration for funding and contributions. The RFCs were written with the engagement of all Tango Controls Community. Authors would like to thank all who contribute by writing, commenting, reviewing, discussing and encouraging to do the work, among others (in last name alphabetical order): Gwenaelle Abeillé (SOLEIL), Reynald Bourtembourg (ESRF), Thomas Braun (byte physics), Antoine Dupre (MAX-IV), David Erb (MAX-IV), Andrew Götz (ESRF), Igor Khokhriakov (IK), Olga Merkulova (IK), Sergi Rubio (ALBA), Graziano Scalamera (Elettra).

REFERENCES

[1] Chaize J.-M., Götz A., Klotz W.-D., Meyer J., Perez M., Taurel E., "TANGO - An Object Oriented Control System Based on CORBA", in *Proc. ICALEPCS 1999*, Trieste, Italy, Oct. 1999, paper WAI2I01.

[2] P.P. Goryl, A. Götz, V.H. Hardion, M. Liszcz, and L. Pivetta, "Towards Specification of Tango V10", in *Proc. ICALEPCS'19*, New York, NY, USA, Oct. 2019, pp. 331–333. doi:10.18429/JACoW-ICALEPCS2019-MOPHA051

[3] ZeroMQ, `https://rfc.zeromq.org/`.

[4] `https://github.com/unprotocols/rfc/blob/master/1/README.md`

[5] `https://guides.github.com/introduction/flow/`.

[6] A. Götz, G. Abeillé, S. Brockhauser, M.O. Cernaianu, J.M. Chaize, T.M. Coutinho, *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, `http://jacow.org/icalepcs2015/papers/wea3o01.pdf`, `doi:10.18429/JACoW-ICALEPCS2015-WEA3O01`, 2015.

[7] `https://github.com/unprotocols/rfc/blob/master/2/README.md`

[8] `https://www.writethedocs.org/guide/docs-as-code/`.

[9] M. Liszcz *et al.*, "Migration of Tango Controls Source Code Repositories", presented at ICALEPCS 2021, Shanghai, China, Oct. 2021, paper MOPV034, this conference.

[10] `https://gitlab.com/tango-controls/rfc`

[11] `https://tango-controls.readthedocs.io/projects/rfc/en/latest/`.

[12] `https://datatracker.ietf.org/doc/html/rfc7764`

[13] `https://gitlab.com/gitlab-org/gitlab/-/blob/master/doc/user/markdown.md#front-matter`

[14] `https://datatracker.ietf.org/doc/html/rfc5234`

[15] `https://gitlab.com/tango-controls/rfc/-/blob/draft/README.md`

[16] `https://www.sphinx-doc.org`

[17] `https://myst-parser.readthedocs.io/en/latest/`.

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-TUBL03

[18] https://tango-controls.readthedocs.io/projects/rfc/en/raw-rfc-10-request-reply/10/RequestReply.html

[19] https://gitlab.com/tango-controls/pogo

[20] https://gitlab.com/tango-controls/rfc/-/issues/128

[21] https://gitlab.com/tango-controls/rfc/-/merge_requests/58

[22] https://gitlab.com/tango-controls/rfc/-/issues?label_name%5B%5D=v10

[23] https://gitlab.com/tango-controls/rfc/-/merge_requests/39