TOWARDS SPECIFICATION OF TANGO V10*

P. P. Goryl, M. Liszcz, S2Innovation, Kraków, Poland
L. Pivetta, Elettra-Sincrotrone Trieste S.C.p.A., Basovizza, Italy
A. Götz, ESRF, Grenoble, France
V. Hardion, MAX IV Laboratory, Lund, Sweden

Abstract

More than 40 laboratories use Tango Controls as a framework for their control systems. During its 18 years of existence, Tango Controls has evolved and matured. The latest 9.3.3 release is regarded as the most stable and feature-rich version of the framework. However, it makes use of what is today considered as outdated CORBA technology which impacts all the stack, from the low-level transport protocol up to the client API and tools. The Tango Community decided to move forward and is preparing for so-called Tango Controls v10. Tango v10 is meant to be more a new implementation of the framework than a release of new features. The new implementation shall make the code easier to maintain and extendable as well as remove legacy technologies. At the same time, it shall keep the Tango Controls objective philosophy and allows the new implementation to coexist with the old one at the same laboratory. The first step in the process is to provide a formal specification of current concepts and protocol. This specification will be the base for the development and verification of new source code. Formal specification of Tango Controls and its purpose will be presented along with tools and methodologies used.

CONTEXT

After the first release in 2001, the Tango [1] control system framework has been continuously evolving and improving, triggered by the request of new features and the need for better performance.

Each new release has been developed guaranteeing full backward-compatibility. Currently, Tango 9 uses CORBA synchronous and asynchronous communication and ZeroMQ protocols for publish-subscribe data transport. Both CORBA and ZeroMQ are well documented protocols, with clean open-source implementation libraries that provide complete APIs. Thus, the need for a complete product specification in a formal language was not mandatory. The existing documentation, in the form of the Tango Controls manual and the API documentation, and the close cooperation of core developers has been sufficient to keep knowledge and compatibility between versions.

However, aging of certain technologies and libraries used by the framework together with the turnover in the developers team, led to a non optimal understanding of some Tango kernel implementation concepts within the growing community of the Tango Controls collaboration.

TANGO RFC

The Tango Request For Comment (RFC) is the name of the project which aims to define the most important aspects of Tango without being tied to any implementation. This is an attempt to separate what Tango adds compared to CORBA and ZeroMQ.

The idea of specifying Tango came after the Tango Kernel meeting held in 2019 at Solaris, Kraków, Poland (Fig. 1). Tango version 10 is a recurrent discussion in the community. Although everyone agrees to remove the obsolete technologies, the analysis of the code showed that Tango is tightly linked to CORBA, making it very hard to reuse the existing C++ implementation called "libtango9". A complete re-implementation is always a risk especially when the current stable version is heavily used and depended on by many sites.

Figure 1: The Tango Kernel group meeting in Kraków, Poland.

Two attempts were made to advance on v10. The first way was to propose another architecture with a different level of abstraction in a form of a plugin. The migration plan would consist of implementing a CORBA plugin first, to check the compatibility with the former "libtango9".

The second attempt was a prototype where CORBA was replaced with gRPC [2] done by the MAX IV Laboratory. It demonstrates the feasibility by making a PyTango* client communicating with a PyTango server. The prototype was breaking not only the backward compatibility with Tango v9 in terms of wire protocol but also changed some CORBA-specific behaviours.

The conclusion was that Tango needs a new implementation in any case which then raises the question "What makes Tango so Tango?". The answer is the Tango RFC project. The Tango RFC project got inspired by the ZeroMQ RFC [3] after investigating [4] how the other open source projects solve this problem.

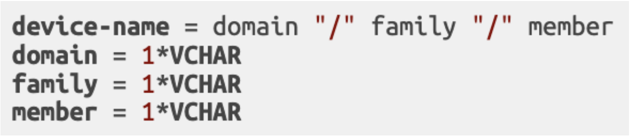
* Work supported by Tango Controls Collaboration

THE ZEROMQ RFC

The ZeroMQ RFC is a formal specification intended to define the communication protocol in order to ensure that different implementations of ZeroMQ stay compatible.. The resulting text and document organisation follows some very simple collaborative principles.

Augmented Backus-Naur Form

Augmented Backus-Naur form (ABNF) [5] is a formal descriptive language, often used to describe network protocols. Its main advantage is the possibility to directly use the specification written in ABNF in order to check the conformity of an implementation. An example of ABNF syntax is shown in Fig. 2.



```
device-name = domain "/" family "/" member
domain = 1*VCHAR
family = 1*VCHAR
member = 1*VCHAR
```

Figure 2: A example of ABNF syntax that specifies the structure of a device name in Tango.

Collective Code Construction Contract

Collective Code Construction Contract (C4) is simple organisation inspired by the git flow in which each specification is managed by an editor and written by contributors. Specific rules are applied regarding the status of the specification.

Consensus Oriented Specification System

Consensus Oriented Specification System (COSS) is a simple, open, shareable, collaborative way to specificy software systems. It ensures the specification does not become proprietary by introducing key component (protected by patent). It allows to track changes in specification. COSS defines a strict lifecycle for the formal documents. This lifecycle is shown in Fig. 3.

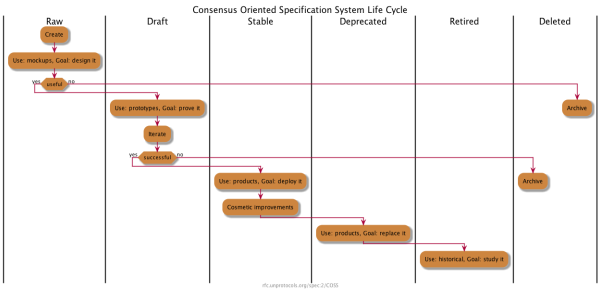

Consensus Oriented Specification System Life Cycle
Raw
Draft
Stable
Deprecated
Retired
Deleted
Create
Use: mockups, Goal: design it
yes
useful
no
Use: prototypes, Goal: prove it
Archive
Iterate
yes
successful
no
Use: products, Goal: deploy it
Archive
Cosmetic improvements
Use: products, Goal: replace it
Use: historical, Goal: study it
rfc.unprotocols.org/spec:2/COSS

Figure 3: The COSS lifecycle defined by the original specification.

ONE SPECIFICATION, SEVERAL IMPLEMENTATIONS

The main goal behind writing the Tango RFC is to guarantee the strict interoperability in terms of behaviour between client and server which may use e.g. different wire protocols. The aim of the RFC is to act as a guideline for any new implementation of Tango in order to ensure the compatibility with other implementations. For example Tango can be implemented in the same programming language but may target a different architecture and still stay compatible or vice versa.

The project started in June 2019 with the objective after 4 months to write the first basic specifications and then evaluate the benefits of the selected mode of work. The goal is to focus on Tango v9 to understand what has to be preserved for the future Tango v10. New features can be added once the specification covers the essential of the Tango model.

Methodology and Tools

The Tango Controls Community, including the institutes which constitute the Tango Controls Collaboration, commercial companies and individuals linked to Tango are involved in writing RFCs. Anyone from the Community can contribute by writing a specification or commenting on the pull-requests.

RFC editors are selected from the institutes participating in the Tango Consortium board.

To keep the Community synchronised, a commercial company (S2Innovation) has been engaged to document the process and animate the team meeting by teleconference every second week. In addition to teleconferences, substantive discussions take place as comments on GitHub [6] pull-requests or issues. On daily basis, a Slack [7] channel supports communication.

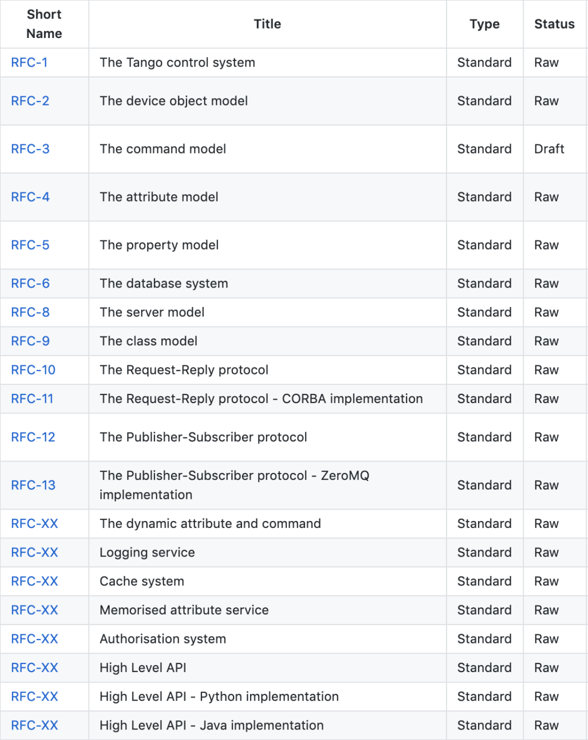
CURRENT STATUS

The documents are based on a common structure, that starts with a meta-data section and a preamble. The purpose of introducing the feature is described and explained by typical use cases. These two paragraphs provide an informal introduction to the following detailed formal specification. Mark-down is the language chosen to describe the document.

Several documents will build up the full specification. Each of the documents is itself a specification of a certain feature (ex. attribute, command, pipe) on a certain level (semantic, behaviour, implementation) that can be, read or applied separately. However, in the current design, there are interconnections between the RFCs.

In order to define a clear and unambiguous specification, the use of key words Indicating Requirement Level [8] was adopted. The capitalized word *MUST* indicates a strict requirement while the word *MAY* indicates an optionsl feature.

Tango RFCs are a work in progress. Figure 4 shows the list of currently proposed specifications. The updated list can be



Short Name	Title	Type	Status
RFC-1	The Tango control system	Standard	Raw
RFC-2	The device object model	Standard	Raw
RFC-3	The command model	Standard	Draft
RFC-4	The attribute model	Standard	Raw
RFC-5	The property model	Standard	Raw
RFC-6	The database system	Standard	Raw
RFC-8	The server model	Standard	Raw
RFC-9	The class model	Standard	Raw
RFC-10	The Request-Reply protocol	Standard	Raw
RFC-11	The Request-Reply protocol - CORBA implementation	Standard	Raw
RFC-12	The Publisher-Subscriber protocol	Standard	Raw
RFC-13	The Publisher-Subscriber protocol - ZeroMQ implementation	Standard	Raw
RFC-XX	The dynamic attribute and command	Standard	Raw
RFC-XX	Logging service	Standard	Raw
RFC-XX	Cache system	Standard	Raw
RFC-XX	Memorised attribute service	Standard	Raw
RFC-XX	Authorisation system	Standard	Raw
RFC-XX	High Level API	Standard	Raw
RFC-XX	High Level API - Python implementation	Standard	Raw
RFC-XX	High Level API - Java implementation	Standard	Raw

Figure 4: The Tango RFC list.

found in the README file in the project repository [9]. Two of the documents, RFC-3 and RFC-4, are already accepted as drafts according to COSS. Five other specifications (RFC-2, RFC-5, RFC-6, RFC-12 and RFC-13) are in raw state, scheduled for review by the editors.

CHALLENGES

Writing specification of the already existing protocol is mostly a matter of describing the facts. However, existing documentation describes Tango Controls from the end-user perspective and essential details of the protocol, like handshaking or keeping the connections alive, are not documented. Besides, the existing documentation is not formal, and its text could be interpreted differently by different readers. Thus, providing a precise and formal specification requires analysis of the source code and testing the different use cases to determine the actual behaviour. Moreover, after initial analysis a few ambiguities were discovered in the current implementation. A simple example is naming of the Tango Properties—different constraints and restrictions are implemented by different parts of the system.

Another challenge is to make the specification readable while keeping it precise and accurate.

CONCLUSION

The Tango RFC project is the result of the Tango Collaboration requirement to distill a complete formal specification for the current Tango 9 design. The formal specification, aimed at providing a complete and comprehensive coverage, will bring benefits to all the institutes that joined the Tango Collaboration during its lifetime, allowing for a better understanding of Tango. The specification will guarantee that the main features of Tango v9 are maintained in future versions of Tango so that sites using Tango or planning to use Tango can be sure to have a stable base on which to base their control system. Moreover, the Tango 9 formal specification will be used as the foundation for defining the future Tango 10. The same methodology will be used to write the v10 specification before starting the development.

ACKNOWLEDGEMENTS

The current progress wouldn't be possible without the engagement of all Tango Controls Community. Institutes, companies and individuals put extent amount of effort in writing and reviewing of the specification. 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), Sergi Blanche-Torne (ALBA), Reynald Bourtembourg (ESRF), Thomas Braun (byte physics), Antoine Dupre (MAX-IV), David Erb (MAX-IV), Andrew Götz (ESRF), Vincent Hardion(MAX-IV), Igor Khokhriakov (IK), Olga Merkulova (IK), Lorenzo Pivetta (Elettra), Sergi Rubio (ALBA), Graziano Scalamera (Elettra).

REFERENCES

[1] Tango, https://www.tango-controls.org

[2] gRPC, https://grpc.io

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

[4] Tango Specification and Experimentation, https://hardion.gitlab.io/tango-rfc-presentation

[5] IETF Network Working Group, *Augmented BNF for Syntax Specifications: ABNF*, https://tools.ietf.org/html/rfc5234

[6] Github, https://github.com

[7] Slack, https://slack.com

[8] IETF Network Working Group, *Key words for use in RFCs to Indicate Requirement Levels*, https://www.ietf.org/rfc/rfc2119.txt

[9] Tango Request For Comments (RFC), https://github.com/tango-controls/rfc