|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| D:\usr\campos\TSB-Reference\Logos\ITU\sigleITU.gif | INTERNATIONAL TELECOMMUNICATION UNION  **TELECOMMUNICATION STANDARDIZATION SECTOR**  STUDY PERIOD 2017-2020 | | | TSAG-C.4 | |
| **TSAG** | |
| **Original: English** | |
| **Question(s):** | | N/A | Geneva, 1-4 May 2017 | | |
| **CONTRIBUTION** | | | | | |
| **Source:** | | Canada | | | |
| **Title:** | | Open Source and Reference Implementation | | | |
| **Purpose:** | | Proposal | | | |
| **Contact:** | | Oscar Avellaneda ISED Canada Canada | | | Tel: +1 613-851-2682 E-mail: [oscar.avellaneda@canada.ca](mailto:oscar.avellaneda@canada.ca) |

|  |  |
| --- | --- |
| **Keywords:** | Open Source; Reference Implementation |
| **Abstract:** | This contribution responds to several questions in TSAG’s living list regarding open source and provides a recommendation on how to proceed |

#### Introduction

This contribution illustrates the experience of some ITU-T working groups in developing source codes for ITU-T recommendations in order to answer the following questions of the TSAG living list:

1. What are the experiences of study groups and focus groups with open source?
2. Is there a requirement of ITU-T study groups to collaborate with open source communities?
3. How can open source foster the implementation of ITU-T Recommendations?

#### Discussion

1. Reference implementations in ITU:

To answer the question: “What are the experiences of study groups and focus groups with open source?” we first looked into existing ITU-T reference implementations, for which a source code is available. We note that the competences to develop these reference implementations are internal to the ITU working groups.

Annex A provides a non-exhaustive list of (8) ITU-T Reference implementations.

While many ITU-T recommendations do not have a companion reference implementation, we should note that:

1. Standardisation experts are more and more involved in source code development (e.g. in W3C, IETF, 3GPP, ITU-T SG16 Q6). ITU-T should start to see these experts and software developers attending ITU study groups.
2. A process exists in ITU, and could be generalise to other groups in ITU-T.
3. Not all ITU-T recommendations need to have a reference implementation.
4. **Development process of a reference implementation (H.265):**

The development of the H.265 reference implementation is taken here as an example. H.265 also known as High Efficiency Video Codec (HEVC), is one of the most recently developed reference implementation and its development process shares some similarities with various open source development. It was jointly developed between ITU-T SG16 WP3 Q6 and ISO/ JCT1/SC29 WG11 (MPEG).

Working group members have provided the necessary resources. The reference software is accessible to all but only working group members can contribute to it by providing a contribution to the working group. Certain tools (such as the bug tracker) can be used by anyone. Membership of neither ITU-T, nor ISO is required to view or submit bug reports.

This reference implementation is provided under the ITU/ISO/IEC copyright and under a 3-clause BSD licence with specific provision excluding other rights (e.g. patent rights).

Annex B provides more details on how the process works and what resources are needed. Simpler development process can be defined, depending on the needs of the working group.

1. **Relation between open source implementation and reference implementation**

To answer the questions:

“Is there a requirement of ITU-T study groups to collaborate with open source communities?” and   
“How can open source foster the implementation of ITU-T Recommendations?” we looked into ITU-T recommendations having a reference implementation and checked if open source implementations were also developed outside of ITU, why and for what purposes.

The table below lists only a few open source implementations developed outside of ITU that are related to ITU-T reference implementations/recommendations:

|  |  |  |
| --- | --- | --- |
| **ITU reference implementation** | **External open source development** | **Note** |
| **G.722** | <https://github.com/cisco-open-source/ffmpeg/blob/master/libavformat/g722.c> | There was no relation between the ITU/ISO working group and any of these open source developers/community |
| **H.264.2 : Reference software for ITU-T H.264 advanced video coding** | <http://www.videolan.org/developers/x264.html>  <http://www.openh264.org/> | There was no relation between the ITU/ISO working group and any of these open source developers/community |
| **H.265.2 : Reference software for ITU-T H.265 high efficiency video coding** | <http://x265.org/>  <http://www.videolan.org/developers/x265.html>  <https://github.com/strukturag/libde265>  <https://github.com/OpenHEVC/openHEVC>  <http://turingcodec.org/> | There was no relation between the ITU/ISO working group and any of these open source developers/community |

We note that these open source implementations started after ITU releasing its reference implementations. The existence of ITU-T reference implementations **triggered and sped-up** the development of external open source implementations by easing the test of functionalities against the reference implementation. As such, ITU-T reference implementations participate heavily in an easier market adoption of ITU-T recommendations.

The objectives underlying a working group's test model and any reference implementation generally differ from those of external open-source implementations. For instance:

|  |  |  |
| --- | --- | --- |
| **Area** | **WG objective** | **External open source software group objective** |
| Standard development | To study and evaluate techniques in a standard's development | None: generally implemented outside the SD process |
| Completeness | Aims to have significant coverage of all features and parameters. | May be constrained to implement the most common or immediate standard features only |
| Speed / Optimization | Generally, not specifically optimized. Readability and ability to later modify code, concepts or operating assumptions is more important. | Often highly optimized for speed, using if necessary low-level techniques and taking care over data use |
| Order | Leads development of standard | Developed in response to a standard |
| Authoritative | The recommendation is authoritative, not the reference implementation | N/A |

**Conclusion**

It seems therefore that reference implementations and external open source implementations of a standard have different objectives which lead to very different implementations. One model cannot simply be replicated for the other.

It is further noted that the existence of reference implementations usually triggers the development of external open source implementations and is a major factor in speeding up market implementation of ITU-T recommendations.

**Recommendations:**

It is recommended that:

1. ITU-T groups consider developing reference implementations for their Recommendations and provide the resources needed for such development.
2. Further contributions on these elements as well as on other aspects of open source be invited.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Annex A**

The table belowprovides a non-exhaustive list of ITU-T Reference implementations

|  |  |
| --- | --- |
| ITU recommendations | Link to the reference implementation (source code) |
| **G.722 : New Annex E - An alternative implementation of stereo superwideband extension using floating point** | <http://www.itu.int/rec/T-REC-G.722-201410-I!Amd1> |
| **G.722.2 : Wideband coding of speech at around 16 kbit/s using Adaptive Multi-Rate Wideband (AMR-WB)** | <https://www.itu.int/rec/T-REC-G.722.2/en> |
| **G.729 : Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)** | <https://www.itu.int/rec/T-REC-G.729-201206-I/en> |
| **G.729 : Reference implementation of G.729 Annex B DTX functionality for Annex D** | <https://www.itu.int/rec/T-REC-G.729-200002-S!AnnF/en> |
| **H.264.2 : Reference software for ITU-T H.264 advanced video coding** | <https://www.itu.int/rec/T-REC-H.264.2> |
| **H.265.2 : Reference software for ITU-T H.265 high efficiency video coding** | <https://www.itu.int/rec/T-REC-H.265.2> |
| **P.862 : Revised Annex A – Reference implementations and conformance testing for ITU-T Recs P.862, P.862.1 and P.862.2** | <http://www.itu.int/rec/T-REC-P.862-200511-I!Amd2/en> |
| **T.835 : Information technology - JPEG XR image coding system - Reference software** | <https://www.itu.int/rec/T-REC-T.835> |

**Annex B**

**Development process of a reference implementation (H.265):**

#### 1) Overview

H.265, the ITU recommendation has authority over H.265.2, the ITU reference implementation. The reference implementation is accessible to all. Only members of the working group can contribute to it by providing a contribution to the working group. When a technical contribution is proposed in the working group; the corresponding code is cross-checked by other members of the working group, to confirm the results provided with the technical contribution.

Only when technical contributions are agreed by the working group, does the companion code find its way in the reference implementation. This ensures the standardisation process remains in control and validates which features are added both to the recommendation and to the reference implementation. It also ensures that the software and the recommendation remain consistent one with another.

#### 2) Resources

ITU-T or ISO members of the working group are in charge of providing the necessary support (servers, human resources, etc.) for hosting the source code and tools during the reference implementation development process on an ad-hoc basis as well as for its maintenance phase. Quite a few of these information is available at: <https://hevc.hhi.fraunhofer.de/>

##### Human resources:

**HM software coordinators**

* Karsten Suehring (HHI, Germany)
* Karl Sharman (Sony, UK)

**Former HM software coordinators**

* Frank Bossen *(NTT docomo, Japan)*
* David Flynn *(BlackBerry, Canada; formerly BBC, UK)*

Software coordinators do not decide which codes will be added or not to the reference software but they follow the decisions made by the working group.

Human resources is a key parameter as software development continuity needs to be guaranteed even when members of the group change affiliation or employment both when the reference implementation is developed and for its maintenance phase.

##### Servers and development tools

Reference software:

The reference software / test model for H.265/HEVC is called HM (**H**EVC Test **M**odel) and licensed under a 3-clause BSD licence with specific provision excluding other rights (e.g. patent rights). The authoritative copy is publicly maintained in a Subversion repository and accessible at:

* [HM software repository (SVN master at HHI)](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/)
* [HM software repository (SVN mirror at BBC)](http://hevc.kw.bbc.co.uk/svn/jctvc-a124/)
* [HM software repository (source browser)](https://hevc.hhi.fraunhofer.de/trac/hevc/browser)

Two independent contributing organizations (members of either ISO or ITU) provide infrastructure for hosting the software repositories, allowing a synchronized mirror to provide a full on-line backup in case of server maintenance and to protect against data loss.

Facilities for the use of the Git distributed version control software is also provided through a mirror: <<http://hevc.kw.bbc.co.uk/git/w/jctvc-hm.git>>

For IPR and liability reasons, the coordinators of the software do not provide any compiled, executable, binaries. Anyone wishing to use the software must compile it themselves.

Bug reporting

Specification and software bugs can be publically reported using the HEVC bug tracker:

* [HEVC bug tracker](https://hevc.hhi.fraunhofer.de/trac/hevc)

The bug tracker is a tool open to anyone. Membership of neither ITU-T, nor ISO is required to view or submit bug reports.

#### 3) Development methodology

A master branch maintains the current stable or near-stable development version. Major release cycles coincide with the Working Group (joint JCT-VC group between ISO and WG16 Q6) meeting schedule. For each major release cycle, one or more development branches are created. Technical contributions approved by the Working Group are submitted to a development branch by the author of the technical contribution, with access control managed by the software coordinators. The software coordinators manage any code review (to maintain quality), the merging of any parallel development branches, and the resolution of any merge conflicts and the testing of the software prior to release.

Additional automation tools, such as continuous integration build reports, are used by the software coordinators. For instance, [automatically generated build results](http://hevc.kw.bbc.co.uk/gitbuilder/) identify accidental changes that prevent the software being compiled in a particular configuration.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_