Call for Offers:

P_2023_04 – ASAM Quality Checker Framework

ASAM calls for offers to support an ASAM project group for implementing a quality checker framework based on existing open-source contributions. Please send your offers for this project to offers@asam.net .
Offers shall be submitted no later than 05.04.2024.

Project

ASAM is continually working on improving the quality of its standards, such as removing ambiguity and enriching them with examples and usage guidelines. This project is taking an additional step. It develops a comprehensive "Quality Checker Framework". This framework lets users verify the conformity of files and implementations against ASAM standards, fostering greater adoption and understanding of the standards and significantly improving interoperability. The framework shall be standard agnostic, allowing the execution of a wide variety of both ASAM and user-defined checks for different standards.

The development is based on the existing QC4OpenX Framework (Quality Checker for OpenX Standards). The source code is already available as open source on GitHub within the ASAM organization.

Besides the framework, this project shall provide checker libraries for three ASAM Standards. For OpenDRIVE and OpenSCENARIO XML, there is also an existing, Python based, implementation published as open-source software on GitHub within the ASAM organization. To demonstrate the applicability of the framework for further standards, the existing rules defined for the Base Standard (BS) for ASAM OTX 3.2.0 shall also be integrated as a compatible checker library for the framework.

The detailed requirements analysis and specification for the framework, the checks to implement and the rule integration process with the standardization projects is done within the ASAM project group and not part of this call for offers.

Here you can find the original proposal:
P_2023_04_ASAM Quality Checker

The complete development of the product is done as a public open-source project on GitHub.
See GitHub repositories:
https://github.com/asam-ev/qc-framework
https://github.com/asam-ev/qc-pyFramework
https://github.com/asam-ev/qc-opendrive

https://github.com/asam-ev/qc-openscenarioxml

The members of the ASAM project group will support the implementation of the product by organizing the change control board (CCB), performing pull request reviews, enhancing the documentation and providing example files for the checker libraries.

Tasks

A. Framework

1. Generalize the current implementation
    I. Remove OpenDRIVE and OpenSCENARIO XML schema Check Bundles from the current framework implementation, to make them independent
    II. Possibility to include/call Check Bundles from arbitrary paths with respect to framework installation directory
    III. The implementation shall handle the following generic issue location types
        a. “FileLocation”, pointing to a line/column in the ASCII input file
        b. “CartesianCoordinate”, pointing to a position a virtual world (e.g. to the simulation map)
        c. “XmlLocation”, pointing to a tree element (only applies for XML based input files)

    The additionally implemented “RoadLocation” shall be handled as a domain specific information and simply passed to the reporting.

2. Extend result file information
    I. Log rules addressed by a check
    II. Version range for which a check is valid (textual information)
    III. Approval information (textual information)
    IV. Runtime metadata (textual key-value pair information)

3. Improve the Configuration GUI
    I. Cross platform, with Microsoft Windows > 10.0.0 and Linux > 5.0.0 as Tier-0 targets.
    II. Possibility to change the order of checks execution within a Checker Bundle
    III. Separate the Checker Bundles execution scheduler in a different standalone executable

4. Improve the Result Pooling
    I. Read the configuration file to
        a. order the results
        b. evaluate minimal/maximal issue level and adjust output results according to those levels
        c. filter results, so that the result after the pooling only contains issues from user requested checks, even if the bundle reports more
        d. optionally filter out consequential errors (i.e., shorten the report, when a check fails)

e. validate for name clashes (in case different Checker Libraries use same names for Checker Bundles) and suggest a solution

5. Improve Report GUI
   I. Support to show arbitrary mime-types ASCII based input files (with optional syntax coloring for XML based file types)
   II. Improve text widget to remove the current limitation of 5MB file size
   III. Increase usability (drag and drop to open, refresh after loading a new result file)
   IV. Show domain specific issue location (e.g. Time, RoadLocation) types in the textual description on an issue
   V. Implement a viewer plugin, which shows the OpenDRIVE map, if an OpenDRIVE or an OpenSCENARIO XML input file is processed
      a. Generate map geometry based on an existing open-source solution
      b. Add marker for issue locations
      c. Remote control of camera → jump to location of selected issue

6. Improve Text Report Module
   I. Write all information from result file to report
   II. Resolve domain specific issue location types to text and write to report
   III. Generate a list of all rule UIDs, which are addressed by the ran checks and a list of rule UIDs, which are violated.

7. Base libraries
   I. Provide a base library in the framework for Python to create and process configuration and result files based on the available implementation
   II. Adapt the existing C++ base library to the changed requirements
   III. Create one simple Checker Bundle and/or Report Module example for C++ and one for Python

8. Create a runtime executable or script, which is directly able to call Python based Checker Bundles from a configuration

9. Create a documentation generator script, which generates a text-based documentation for a Checker Bundle containing all included checks, the addressed rules, the approval information, their parameters and run-time metadata out of an empty result file

10. Adapt and improve the existing documentation

B. Checker Libraries for ASAM OpenDRIVE and OpenSCENARIO XML

1. Adapt the existing implementation to the framework

2. Remove checks, which are not approved by the standardization project

3. Implement missing checks

Here can be found examples of the checks and more information:
ASAM OpenDRIVE
https://github.com/asam-ev/qc-opendrive/tree/main/checks
https://github.com/asam-ev/qc-opendrive/tree/main/doc

ASAM OpenSCENARIO XML
https://github.com/asam-ev/qc-openscenarioxml/tree/main/checks
https://github.com/asam-ev/qc-openscenarioxml/tree/main/doc

C. Checker Library for ASAM OTX

Implement a wrapper for the existing base OTX checks to include them into the Quality Checker Framework. There are 55 Checker Rules on the specification. The specification is available on demand subject to an NDA.
ASAM OTX Extensions

D. Demo CI/CD pipeline

Create a demo pipeline on GitHub, where an arbitrary user can upload his OpenDRIVE file. The pipeline should trigger the framework to run the OpenDRIVE Checker Library with a standard configuration and create a text report. The user can then download the machine-readable result file and the human-readable text report.

E. AsciiDoc based user documentation

The service provider shall implement the ASAM documentation framework in the project's GitHub repository with Antora and ASCiiDoc, a lightweight markup language that supports all structural and semantic elements necessary for writing technical documentation. Setup an automated CI pipeline that combines AsciiDoc/Antora files (source) with styling templates in order to achieve a fully featured html output. in the project's GitHub repository. This framework shall contain the general structure of the documentation with all mandatory chapters as defined by ASAM.
All authored content shall be aligned with the ASAM Editorial Guide. This is the responsibility of the contracted partner. Where necessary, restructuring or editorial changes shall be made to comply with the Editorial Guide.

Deliverables

All deliverables will be expected in open-source repositories on GitHub within the ASAM organization, published under the MPL-2 license.

- Final working source code of the framework and the checker libraries including their documentation and examples
- Build environment to create the products in a runnable version, which can be downloaded by users in their development environment
- Tests environment performing regression tests to ensure the basic functionality of the framework and the Checker Libraries. All tests shall be executed based on test input files to

the executables/scripts and a comparison to reference output files. The tests for the Checker Libraries shall use the provided examples for the implemented checks.
- Demo CI/CD pipeline on GitHub integrating the framework and one of the checker libraries in a standard configuration.

Qualification

In order to be able to carry out the tasks, the service provider shall meet the following conditions:

- Expertise in C++, Python and the processing and generation of XML
- Practical work experience in agile software development on GitLab/GitHub
- Experience with ASAM OpenDRIVE and/or ASAM OpenSCENARIO XML would be beneficial
- High proficiency in technical English

General Conditions

Please note the following general conditions for the project:

- The project period is from 08.04.2024 to 30.10.2024
- The project lead, ASAM office and service provider will jointly determine the need for attendance of the service provider in meetings.
- Offers have to be preferably work packaged based (categorization according to the different tasks presented in this document). The offers can be made addressing one, more or all the tasks.
- In case of variable-price offers, please indicate upper limits for the costs
- In case a deliverable is found to be defective or not up to customary industry quality standards, the service provider will correct the deliverable without charging ASAM for the extra work. Not completed work is exempted from the service provider's warranty, i.e. if a deliverable is found to be incomplete, then its completion will not be paid by ASAM
- Modification to deliverables, should they be required, will be jointly reviewed and the order adjusted accordingly
- All work results of the project (including the sources to create them) will be handed over to ASAM and will become the exclusive property of ASAM. If the general terms & conditions of the service provider contain rules that are in conflict herewith, then they must be identified and explicitly declared as invalid for this offer
- Offers have to be submitted in writing to ASAM within the above stated bidding period, including a clear description of the provided services and deliverables, terms & conditions, time period and costs for the service
- Please note that offers will be reviewed by the members of the project group, the Technical Steering Committee of ASAM and the ASAM Office. The project group members will not receive pricing information. For this purpose, please put all pricing information on a separate page