1. Static Structure of aadl-qa

1.1 Test Suites [src/]

The aadl-qa/src/ directory contains a set of test suites.

A ***test suite*** is a directory that contains
- MANIFEST.TS and
- either a set of test cases or a set of test suites.

So there can be a hierarchy of test suites.

A ***test case*** is a directory that contains MANIFEST.TC and one or more aadl files. If common include directories/files are required, it is explicitly stated in MANIFEST file of a test case or a test suite. As a result, a set of aadl files passed to a tool under test consists of:
- all aadl files in the test case directory
- all aadl files from include directories mentioned in MANIFEST file of the test case
- all aadl files from include directories mentioned in MANIFEST file of all test suites the test case belongs to;
- all common aadl files mentioned in MANIFEST file of the test case and of all the test suites.

MANIFEST.TC file contains attributes in format of NAME = VALUE. Predefined attributes includes:
- EXPECTED_RESULT : VALID if aadl specification is syntactically and semantically valid and INVALID otherwise;
- EXPECTED_* - *TBD*
- REQ [optional] : an identifier of a requirement or a test purpose that is covered by the test case;
- *NYI* - expected_errors: error-list (???) some descriptors of expected error messages in tool-agnostic form???
- *NYI* - instance_roots : instance_root-set

All relations between test suites and test cases are defined just by location of the corresponding directories in a file system.

In the archive src/ contains a fake hierarchy of test suites to play with.

An example can be something like this:
- src
  - test-suite1
    - MANIFEST.TS
    - subtest-suite1_1
      - MANIFEST.TS
      - test-case1_1_1
        - MANIFEST.TC
        - file1_1_1.aadl
      - test-case1_1_2
        - MANIFEST.TC
        - file1_1_2.aadl

        - subtest-suite1_2
            - ...
    - test-suite2
        - ...

1.2 Requirements[requirements/]

It is a text file based DB that will keep a hierarchy of requirements with explicit linkage to the text of AADLv2.1 specification. Also it will contains test cases that covers the requirements.

The goal the database is to store all requirements of AADL specification in a hierarchical form and to keep linkage of the requirements to the text of the specification. Test cases can be stored in the DB as well. But test cases from src/ also can be traced to the requirements.

In the archive it contains just several examples marked up in AADL specification and by one test purpose for each of the example.

aadl-qa/gensrc/ is generated by 'aadl-qa.pl gensrc' command from requirements/ db.

To generate gensrc/ it is required to have an external tool Requality and JRE(any >= 6.0 except for gcj). To install Requality you have to untar an archive [1] and to make sure that {untarred_dir}/bin is in PATH before you run 'aadl-qa.pl gensrc'.

A GUI to work with the db is Requility plugin to Eclipse. It can be installed from
http://forge.ispras.ru/repo/requality/site/
There is a known bug in xulrunner that leads to very long opening of AADL specification in Eclipse [up 10 minutes]. But if Eclipse uses WebKIT engine for internal browser it work well.

[1] http://forge.ispras.ru/attachments/download/1804/requality-0.13.144-alpha-120224.tar.gz
from http://forge.ispras.ru/projects/reqdb/files

1.3 Tools [tools/]

- manager/ - manages actual execution of the tests and report generation
    - Adapters
        - ToolAdapter.pm – generic implementation of ToolAdapter
        - OcarinaAdapter.pm – adapter for ocarina inherited from ToolAdapter (requires to have ocarina in the PATH)
- aadlgen/ - generate AADL test suite from the requirements/ DB (it is a part of aadl-qa because it is specific for the aadl-qa)

1.4 aadl-qa.pl

The main script to run tests.

```
aadl-qa.pl run Ocarina - Run all tests the Ocarina and generate report
aadl-qa.pl gensrc     - Generate AADL tests from other sources
aadl-qa.pl clean      - Remove all generates files
aadl-qa.pl --help     - Print this message
```

After running the tests see reports/index.html and links to requirements from it.