

Technical Report

New xml output

Author: *Jérôme Legrand*

1 Introduction

CheddarKernel is a special version of cheddar without the aadl parser and the gui. It is used mainly by ellidiss technologies through AADLInspector or the taste toolchain. CheddarKernel uses the cheddar framework but needs specific framework modifications. These packages are copied to the ellidiss directory and, thanks to the gpr capabilities, overload the original ones. These modifications mainly consist in adding a new xml output but it can be used to make framework modifications (an attempt to optimize the simulation engine was also done) without interfering with the cheddar source code.

The main problem with this approach is that each time a modification is done in the original framework packages, a manual difference merging has to be done to the copied ones. This operation can take a couple hours and is prone to mistakes.

After meeting with the main cheddar's developers, it has been decided to replace the previous xml output, which is not used and not maintained, by the one from cheddarKernel. Furthermore, some package from the ellidiss directory were obsolete. They were used by the optimization of the simulation engine.

2 Work done and reflection

In this section, the work done is presented as well as some thought about its current state and possible future modifications.

2.1 xml validation :

It is not related to the xml output but by default the xml projects must be validate. For cheddarKernel, it was needed to remove this verification. The option was there in the concerned functions of the framework but always set to true. A parameter has been added to allow the validation to be set to false.

2.2 Replace the current xml output

- remove the old one :

all the switches referencing `xml_output` has been removed (generally the `set_tag` function was called) as well as all tags from the `xml_tag` package.

- add the ellidiss xml output :

The xml formatting from the cheddarKernel packages is far from perfect and only partial. It consists in a switch between the textual/raw and the xml output for the services used by cheddarKernel. A better way to make a new xml output would be to design what a service result should be and, if possible, based on that design, automatically generate the output functions. This work could be huge, time consuming and should involved all cheddar developers, so we (ellidiss) decided to stay on the simple version for now.

2.3 Remove the obsolete packages and references to the simulator engine optimization :

The objective was to add some heuristics for jumping to the next relevant event. Due to the fact that the cheddar simulator engine is based on next event calculation and is not dynamic, the optimization was useless. There were also a preliminary work on a socket based communication with the simulator engine in order to add more dynamic behavior.

Therefor, the *heuristics*, *marzhin_utils* and *sockets_overlay* packages were archived.

2.4 Regression tests :

After modifying the code, tests must be conducted.

For Cheddar, an automatic regression tests process exists. The main objective, for this work, is to check if the textual output is still the same.

For cheddarKernel, no automatic test exists for checking the xml output, so a manual verification is done. Tests can be added in the Cheddar regression test process :

- add an output argument to “call_unit_test” (which can be set to `xml_output`) and pass to “sequential_framework_request”
- add a command line to the “unittest” program to set the output to xml.
- and, of course, add the tests in the “regression_tests” one.

2.4.1 Cheddar/unittest

In order to check if there is no regression with all these modifications, the regression tests have been done.

- `./test_memory_footprint1.xmlv3` and `./test_memory_footprint2.xmlv3.R3.4`

There is a problem in the file loading, the parser complains about the missing period of the task A2s.

- `./basic_scheduling_simulation/multiprocessor/test_run2.xmlv3`

The main result is identical but, in the current version, the event table is added.

- `./basic_scheduling_simulation/uniprocessor/test_resource_wait_post.xmlv3`

The results are identical except for the line of the “constraint error”.

- `./basic_scheduling_simulation/user_defined_schedulers/test_parametric6.xmlv3`

There is a lot differences in the end of the simulation. This is probably due to the random numbers found in the “task_activation_section” section in the `activations.sc` file.

- `./basic_scheduling_simulation/user_defined_schedulers/test_parametric15.xmlv3`

There is a lot of differences in the beginning of the files, in the “Line” part. This is probably due to the random numbers found in the “priority_section” section in the `laplace_gauss.sc` file.

- `./test_memory_footprint1.xmlv3`

There is no result for this test in the current version. I do not know why.

There is also a missing reference file for the 3,3 reference version :

“./basic_scheduling_simulation/uniprocessor/test_ppcp_tasks_sems.xmlv3.R3.3”.

I've compiled a unittest version with the previous version of the source and here is the result with the problematic tests :

- ./test_memory_footprint1.xmlv3 and ./test_memory_footprint2.xmlv3.R3.4: same error
- ./basic_scheduling_simulation/multiprocessor/test_run2.xmlv3 : same differences. The 2 results from the previous unittest and the new one are identical.
- ./basic_scheduling_simulation/uniprocessor/test_resource_wait_post.xmlv3 : same differences. The 2 results from the previous unittest and the new one are identical.
- ./basic_scheduling_simulation/user_defined_schedulers/test_parametric6.xmlv3 : the 3 results (reference 3.3, previous modification and after modification) are different.
- ./basic_scheduling_simulation/user_defined_schedulers/test_parametric15.xmlv3 : the 3 results (reference 3.3, previous modification and after modification) are different.

So the added source code does not seem to change the results than the previous one and the only difference in the tests seems to be coming from the random generation numbers from the 2 parametric tests.

2.4.2 CheddarKernel

For cheddarKernel, only the xmloutput has been checked through the AADLInspector prototype and seems to be ok.