20th International Conference on Computing in High Energy and Nuclear Physics (CHEP2013)

Contribution ID: **459** Type: **Poster presentation**

Round-tripping DIRAC: Automated Model-Checking of Concurrent Software Design Artifacts

Monday, 14 October 2013 15:00 (45 minutes)

A big challenge in concurrent software development is early discovery of design errors which can lead to deadlocks or race-conditions. Traditional testing does not always expose such problems in complex distributed applications. Performing more rigorous formal analysis, like model-checking, typically requires a model which is an abstraction of the system. For object-oriented software, UML is the industry-adopted modeling language, offering behavioral views that capture the dynamics of the system with features for modeling code-like structures, such as loops, conditions, and referring to existing interactions. We present an automatic procedure for translating UML into mCRL2 process algebra models, amenable to automatic model-checking. Our prototype is able to produce a formal model, and feed model-checking traces back into any UML modeling tool, without the user having to leave the UML domain. We apply our methodology to the newly developed Executors framework, part of DIRAC's WMS system responsible for orchestrating the workflow steps of jobs submitted to the grid. Executors process any task sent to them by a Dispatcher, each one responsible for a different step (such as resolving the input data for a job). The Dispatcher takes care of persisting the jobs states and distributing them among the Executors. Occasionally, tasks submitted in the system would not get dispatched out of the queue, despite the fact that their responsible Executors were idle at the moment. The root cause of this problem could not be identified by certification testing with different workload scenarios, nor by analysis of the generated logs. We used our toolset to generate an mCRL2 model of this system, based on the reverse-engineered sequence diagrams. Model-checking the generated mCRL2 model discovered a trace violating the desired progress requirement, the bug being localized in the Dispatcher component implementation. The trace was automatically translated back to the UML domain, showing an intuitive view of the communication between components which leads to the faulty behavior.

Primary authors: CASAJUS RAMO, Adrian (University of Barcelona (ES)); REMENSKA, Daniela (NIKHEF (NL)); TEMPLON, Jeff (NIKHEF (NL))

Presenter: REMENSKA, Daniela (NIKHEF (NL))

Session Classification: Poster presentations

Track Classification: Software Engineering, Parallelism & Multi-Core