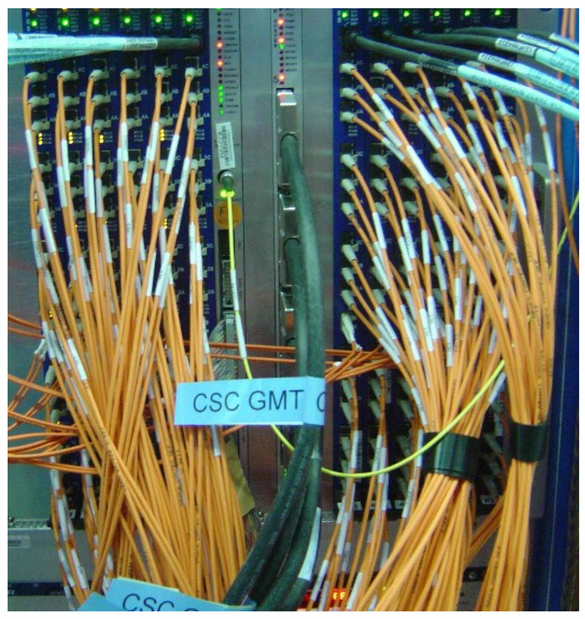
Interconnection Test Framework

Level-1 Trigger

Josef Hammer · CERN

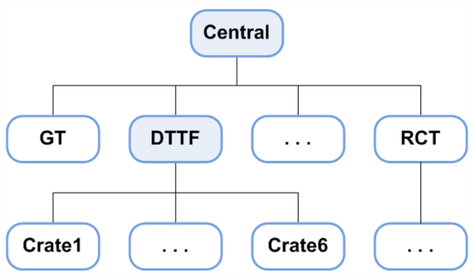


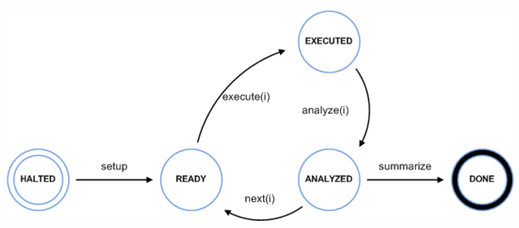
Wrong cable?

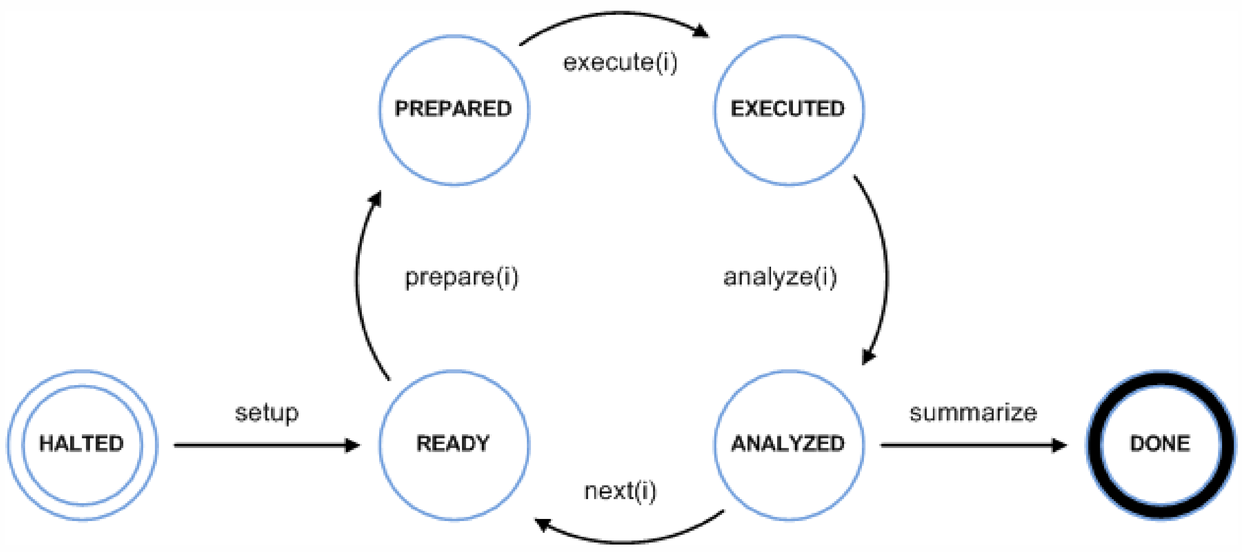
Damaged link?

Incorrect code?

Test to ensure correct physics data!







Architecture · FSM · Shared Memory

The communication hierarchy is strictly top-down (request-response via SOAP; figure upper left). Each node runs one *Interconnection Test operation* per test type – a specific sub class containing the concrete test functionality for that node. A *test case* – an XML document – defines test type, participating nodes, parameters and the order of execution.

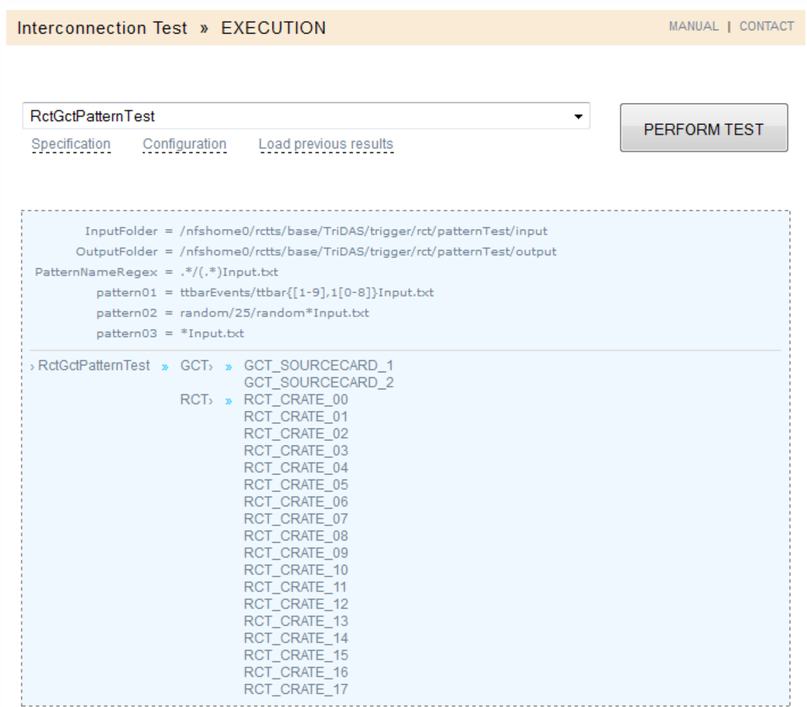
Each test operation runs the same *finite state machine* (figure above) which is kept synchronized across all nodes. The number of loops may be specified in the test case or at runtime. For improved performance the FSM is modified on-the-fly to remove empty transitions (figure upper right) – cutting off single nodes or entire sub-trees.

A name/value pair-based *shared memory* is provided to allow flexible communication between nodes. A name may also refer to a file on the local hard disk – it will be synced automatically to any node that needs that file.

Service Discovery

Simplicity is king. Wherever possible, the *convention over configuration* paradigm was applied. The *Service Discovery* component scans the network for available test operations (and their parameters) and creates the test cases on-the-fly.

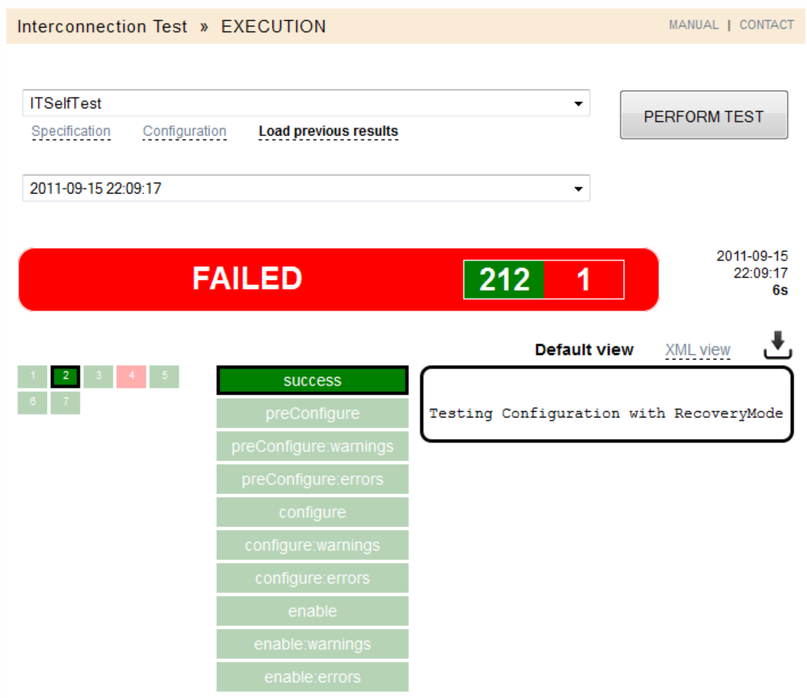
Usually, pressing the *PERFORM TEST* button is all there is to do for the user. The test case shown below was created automatically – without any input from the user.



Default result view

In order to visualize the test result in a user-friendly way, a generic result view is provided. Utilizing AJAX even huge test results with thousands of items can be browsed through easily.

Custom result views / analyzers allow interpreting and/or visualizing the result in a specific way for a particular test type.



Custom result view

for DTTF