



Automated Test & Release Environment for GEDAE

by Martin Lang and Andreas Bogner

GEDAE Users Conference March 2003



Overview

- Motivation
- Test Philosophy
- Test Tool
- Release Tool
- Summary



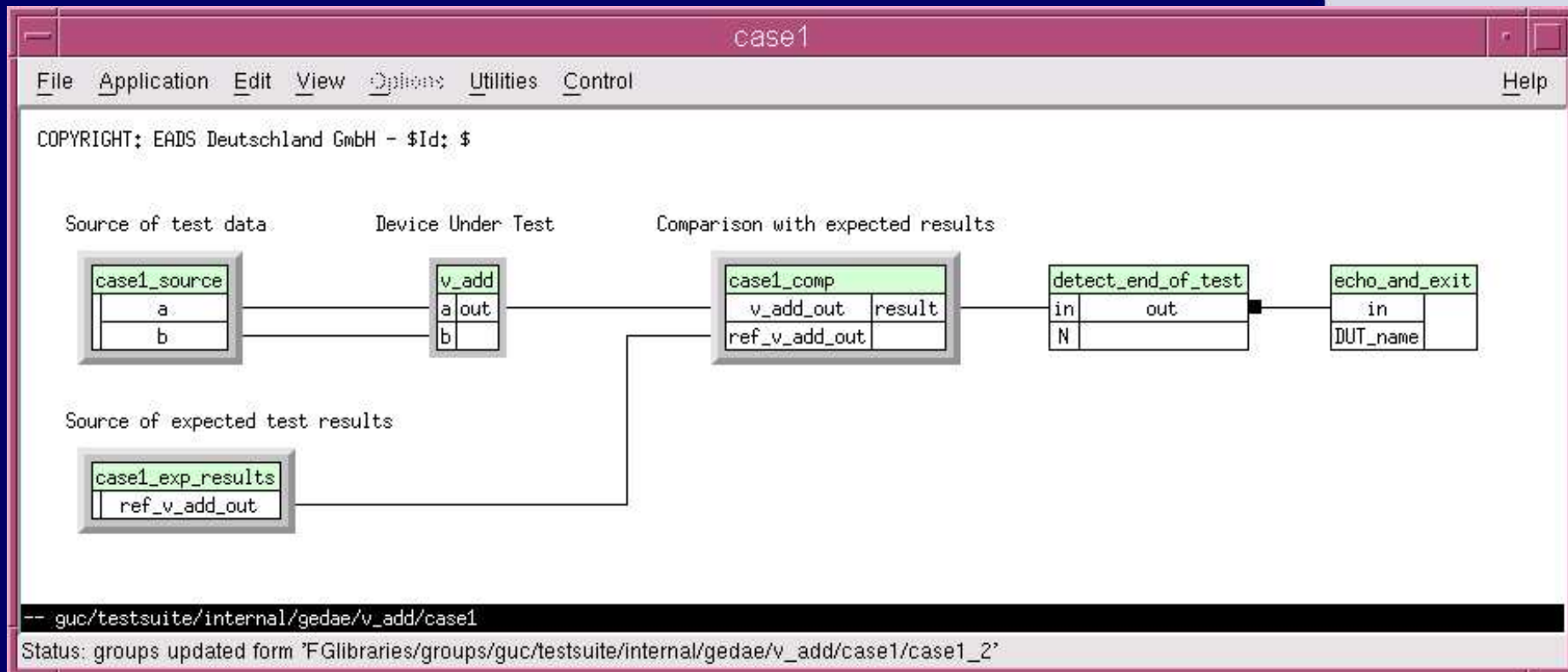
Motivation

- Current Project dimensions:
 - multiple modes build up with common algorithms
 - hundreds of common algorithms
 - thousands of flow graphs and primitives
- Multi-site development without a direct TCP/IP connection

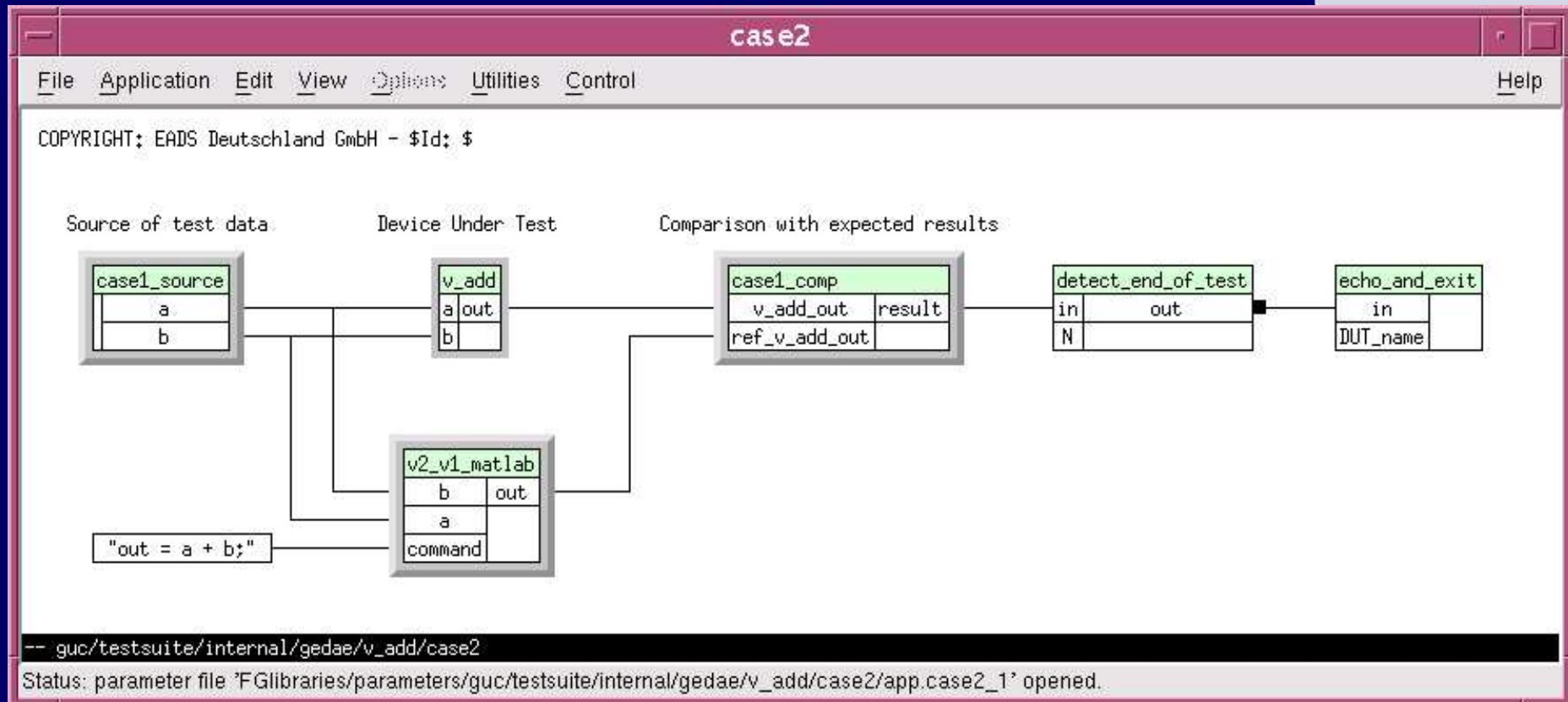
Test Philosophy

- Unit Under Test (UUT) can be:
 - a primitive
 - a flow graph (algorithm, sub mode, mode)
 - a complete application with external interfaces
- Most important test principles:
 - bottom-up approach
 - four eye principle
 - automate all tests
- The tests itself can be organised as
 - black box tests
 - white box tests

Test Bench with External Reference Data



Test Bench with Internal Reference Data Generation



Test Support Tool – Key Features

- Filtering of algorithm and modes according to
 - Packages
 - Responsible developers
- Selection of algorithms and modes within list
- For selected algorithms and modes
 - Selection of available test cases
- Generation of reference data
- Control of the most common GEDAE options
- Execution of GEDAE test bench on
 - Host
 - Target(s)

Test Shell

The screenshot shows the GEDAE Test Shell interface with the following components and callouts:

- GEDAE Command Line:** Points to the top terminal window containing the command: `gedae -file guc/testsuite/internal/gedae/v_add/casel -param casel_1`
- Reference Command Line:** Points to the second terminal window containing the command: `cd FGlibraries/boxes/guc/testsuite/internal/gedae/v_add;matlab -nosplash -nodesktop -r 'casel_1,quit'`
- List of algorithms and modes:** Points to the left sidebar menu listing: `internal/gedae/bench1/bench1`, `internal/gedae/v_add/casel`, and `internal/gedae/v_add/case2`.
- List of all tests for the selected algorithm or mode:** Points to the central list of test parameters: `-param casel_1`, `-param casel_2 -group casel_2`.
- Options to control GEDAE:** Points to the control panel on the right, including checkboxes for `run`, `timeout 10`, `enable trace`, `dump trace`, `open trace`, and `rsh to skyl`.
- Execute GEDAE / Reference:** Points to the `Execute` button.
- Select Package:** Points to the `Package filter:` dropdown menu.
- Select User:** Points to the `rsh to skyl` field.
- Run on Target:** Points to the `remove` button next to the `tt_events` field.
- Edit Testsuite Configuration File:** Points to the `Edit testsuites.tab` button.

Underlying Directory Structure

`boxes/<project>/algorithm/<group>/<name>/`

`boxes/<project>/testsuite/algorithm/<group>/<name>/case<X>`

`boxes/<project>/testsuite/algorithm/<group>/<name>/in/`

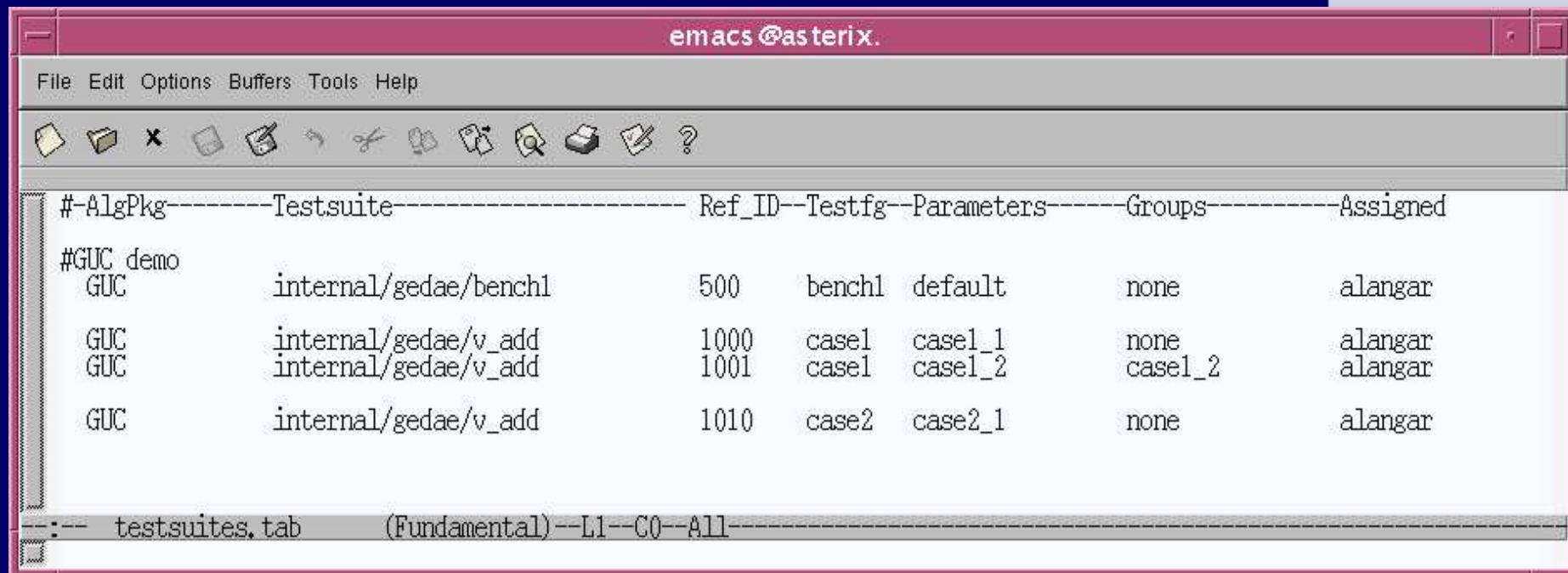
`boxes/<project>/testsuite/algorithm/<group>/<name>/out/`

`boxes/<project>/testsuite/algorithm/<group>/<name>/ref/`

`parameters/<project>/testsuite/algorithm/<group>/<name>/case<X>/
app.case<X>_<Y>`

`groups/<project>/testsuite/algorithm/<group>/<name>/
case1/case<X>_<Z>`

Testsuite Configuration File

The image shows a screenshot of an Emacs editor window titled "emacs @asterix.". The window has a menu bar with "File", "Edit", "Options", "Buffers", "Tools", and "Help". Below the menu bar is a toolbar with various icons for file operations. The main text area contains a table with the following content:

```
#-AlgPkg-----Testsuite-----Ref_ID--Testfg--Parameters-----Groups-----Assigned
#GUC demo
  GUC      internal/gedae/bench1      500    bench1  default      none      alangar
  GUC      internal/gedae/v_add             1000   casel1  casel_1      none      alangar
  GUC      internal/gedae/v_add             1001   casel1  casel_2      casel_2   alangar
  GUC      internal/gedae/v_add             1010   case2   case2_1      none      alangar
```

At the bottom of the window, a status bar shows "testsuites.tab (Fundamental)--L1--C0--All".

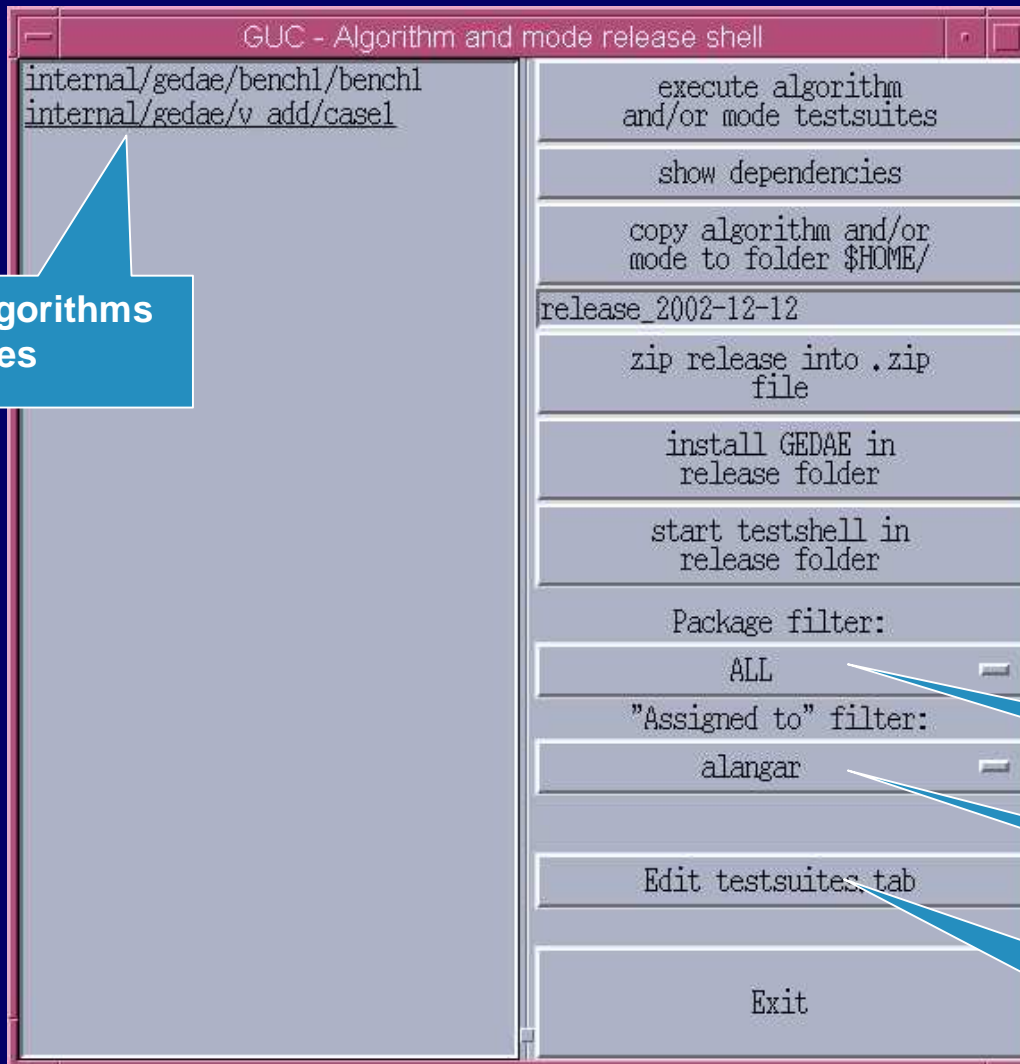
Automated Tests

- Special command line version of the Test Shell
 - Nightly tests of all test benches on
 - Host
 - Target(s)
 - Tests are performed from a clean environment
 - Overall results are automatically forwarded to a test manager
 - Detailed results of failed test benches are mailed to the responsible developer
 - Test results are logged permanently for
 - customer needs
 - metrics

Release Tool – Key Features

- “Enhanced FGU” utility
 - Filtering of algorithm and modes according to
 - Packages
 - Responsible developers
 - Execution of all selected test benches on
 - Host
 - Target(s)
 - Positive side-effect:
 - Collection of every file access (read, write)
 - Archive all accessed file
 - Verification of generated release in a separate directory

Release Shell



List of algorithms and modes

Consecutive Commands to generate Release

Select Package

Select User

Edit Testsuite Configuration File

Summary

- Test philosophy has been applied to all generated algorithms and modes
- Integration time of algorithms in modes is reduced
- Test Shell is the standard GEDAE access tool for the whole development team
- Release Shell simplifies data exchange between multiple sites
- Benefit of automated tests:
 - Improved reliability of algorithms and modes
 - Problems due to changes are detected on a daily basis