

TEMA vs fMBT

Scope

TEMA: very domain-specific: UI-testing, multi-application, single active app.

fMBT: generic. From APIs to distributed systems and UI testing.

Modelling language syntax

TEMA: visual modelling language

fMBT: (we'll concentrate to AAL/Python, yet there are alternatives)

- Mostly Python.
- Automatically visualised, viewable in different perspectives.

Modelling language expressive power

TEMA: finite state machines, parallel composition, some data extensions

- limited handling of data
- limited communication between model components
- restrictions (such as two layers, strong connectivity)
- safe

fMBT: Python

- any Python data can be handled
- an action can be enabled or disabled based on observations (existence of a file, an icon on the display, database...)
- no restrictions, but using external data affects visualisation
- unsafe

Adapters

TEMA: limited to the extreme, still maintaining ability to make observations

- executing an action either succeeds or fails
- all actions are "input"

fMBT: no limitations

- all Python libraries are available
- AAL/Python adapter can directly read & write model variables
- "input" and "output" actions:
 - adapter can report execution of actions that cannot be triggered by test generator.
- extras: with more limitation it's possible to execute separate test steps on separate hosts.