Last edited by 🔲 **Ondřej Vašíček** 1 day ago

Requirements of Automated Analysis Tools

Types of tools

Stateless dynamic analysis (Anaconda, Valgrind, grep)

- Use launch command and build command
- Build the tool

Static analysis (Facebook infer)

- Use build command
- Do not build

Other (hilite - test case geenration)

- Does not need an SUT, just a payload or an artifact
- _hus no build or launch command, maybe use the title

Statefull analysis or combination of tools (perun)

- Multiple analysis requests on the same SUT - easiest and the most easy to use
- *Alternative is capturing state in artifacts and then transferring those around*

General steps for tools

Before analysis

- setup the tool
  - install tool [outside of the adapter]
  - Create auto plan using XML config

When requesting analysis

- Get SUT to the server and build
  - Post to compilation adapter with build command and launch command
- Configure the tool
  - Conf files with the SUT
  - Or add conf files as input parameters
  - Or add an option to create a file in the SUT directory before the analysis
- Specify tool input parameters and execute
  - Defined in the auto plan with commandline positions
  - SUT launch command can be fetched by the adapter
  - Set environmental variables
  - *Maybe an input parser that builds the command string to execute?*
    (could remove command line positions from the inputs) [FUTURE IDEA]

During analysis execution

- Check status
  - Get the result or request
- Cancel execution
  - Update or delete the request or result
- *Make execution a priority [FUTURE IDEA]*
  - *Update the request or result?*

After analysis

- Get analysis outputs
  - Currently get all that match regex including their contents
  - Or using plugin output filters
- Persist analysis outputs
  - Currently same as output parsers
  - *Future through a "datastorage adapter" [FUTURE IDEA]*
- Get info about execution run
  - State and verdict – if it finished successfully
  - *Add some statistics – oslc monitoring? Run duration, etc. [FUTURE IDEA]*
  - *Maybe implement a way for users to set the vedict – eg. in the output parsers [FUTURE IDEA]*

- Optional clean up
  - Right now create a new workspace (SUT)
  - *Could add a post run clean up script? or a way to reset an SUT / workspace [FUTURE IDEA]*
- *Get the whole SUT workspace from the server [TODO]*
  - *GET the SUT with accept octet-stream*
- Run a follow up analysis on top of the current results
  - POST a new request referencing the same SUT