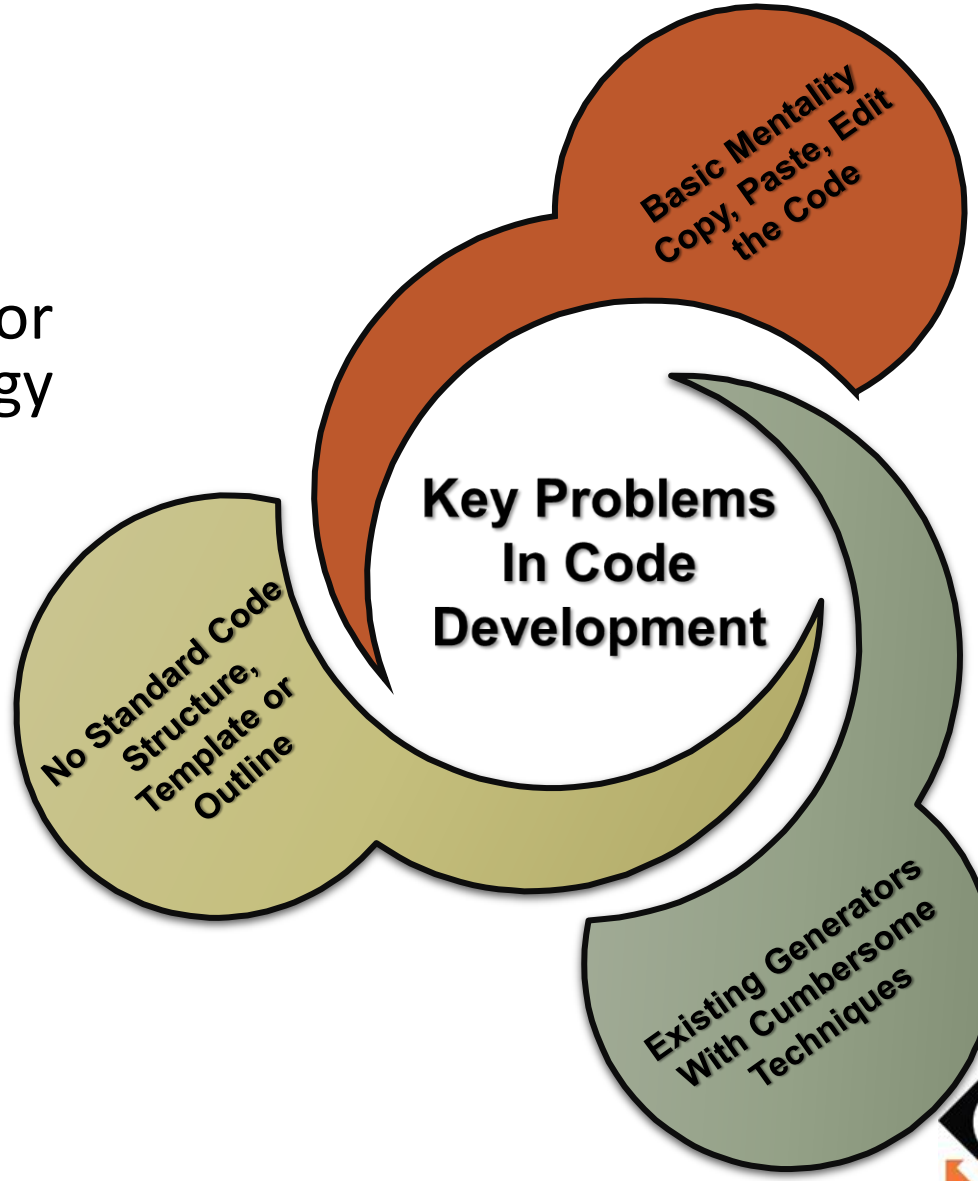
Novel GUI Based UVM Template Builder

Vignesh Manoharan

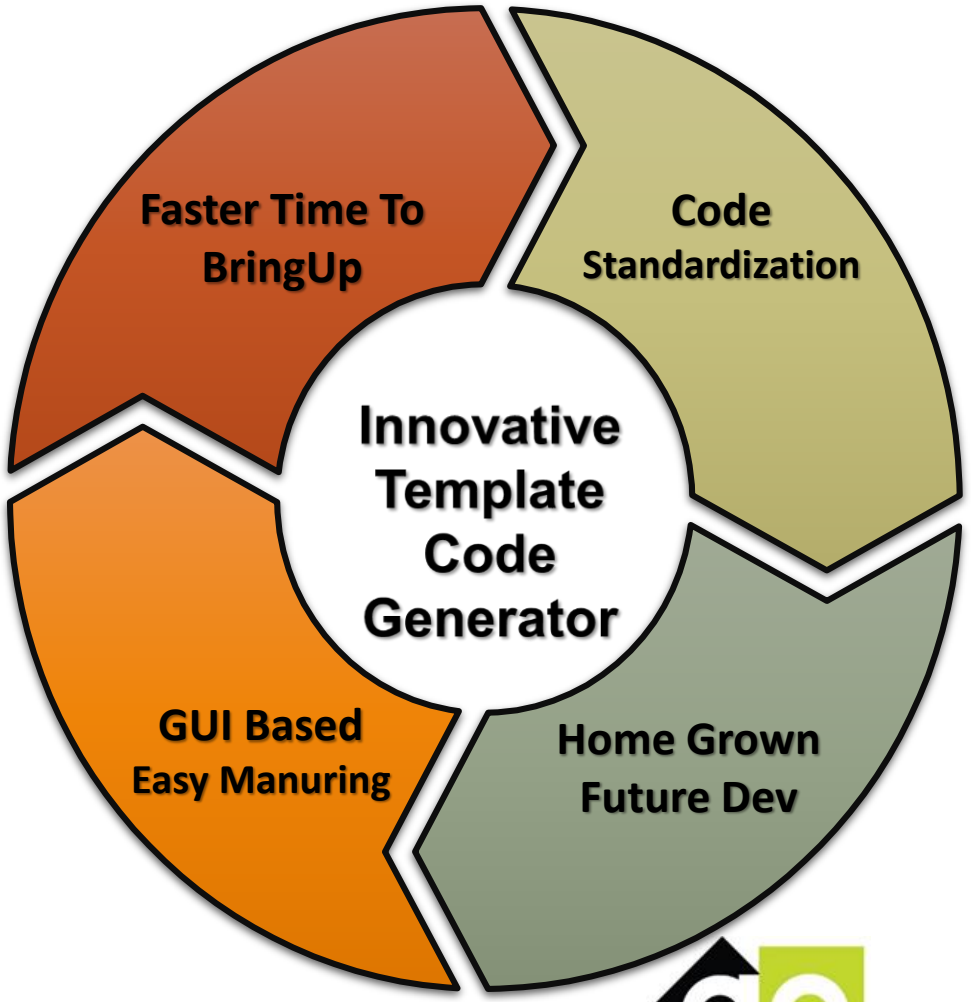
Problem Description

- Ever increasing Adoption Rate for Universal Verification Methodology (UVM).
- Cumbersome nature in writing effective re-usable code.
- Can the template code development be automated ? Any innovative approach ?



Innovation Description

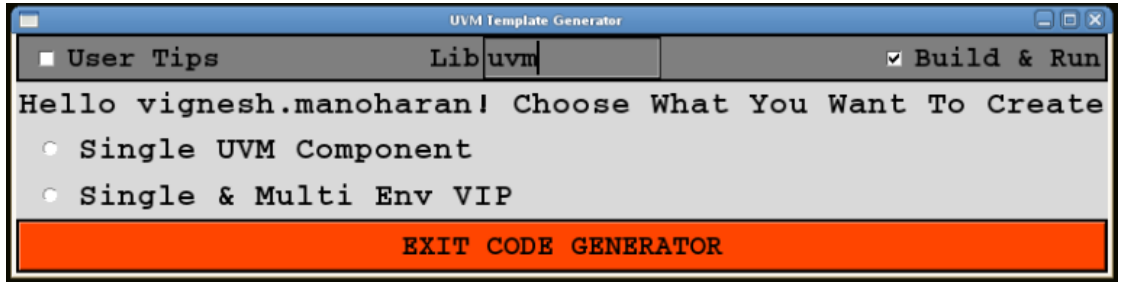
- Able to develop/generate all the required testbench shells in fraction of seconds instead of days.
- Helps in integrating verification collaterals across projects and thus helps in faster re-usability.
- Build structured UVM template codes with proper comments, indentation, environment configuration dump, list files etc.



Tool Operation

How The Tool is Being Built:

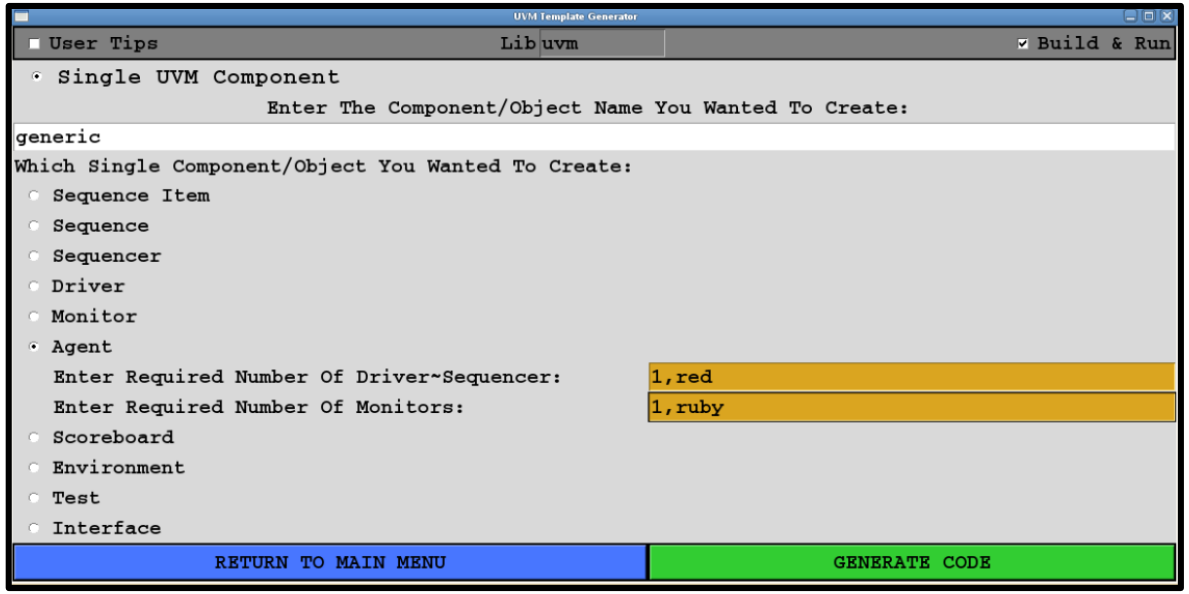
- The GUI portion of the tool is built using Tkinter library from Python.
- Exhaustive text/string editing is done using Python.
- File generations are done using Shell script.



Tool's Opening/Initial GUI Interface

- The tool provides the user with options to create single UVM components or complete UVM VIP.
- The moment user launches the tool, the following right side GUI is launched.
- The user has to click the required radio buttons to choose what user wanted to create, either it can be single components or entire test bench.

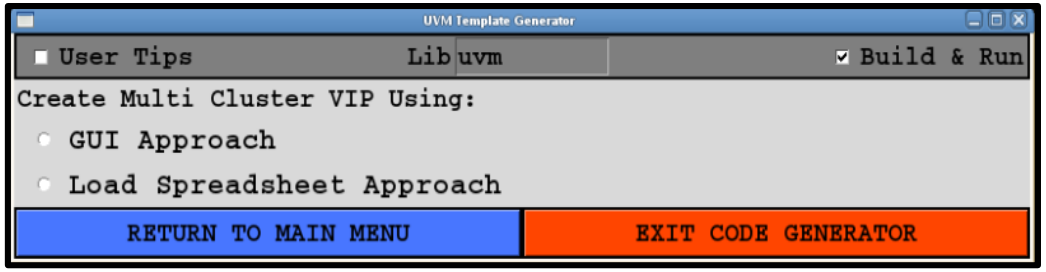
Single UVM Components



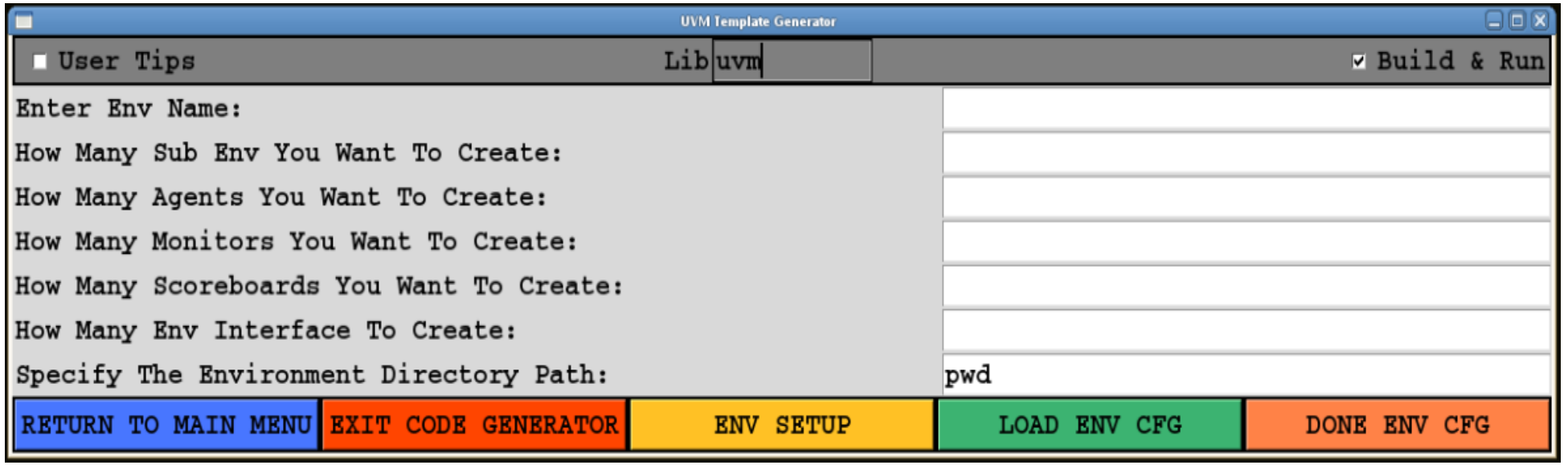
- Left snapshot shows the details of the tool when user choose single UVM component creation.
- User needs to enter the required name of the component, choose the corresponding component to create and press the ‘Generate Code’ button to develop the code template.

Complete UVM VIP

- Right side snapshot shows the details of the tool when user choose Single & Multiple env UVM VIP.

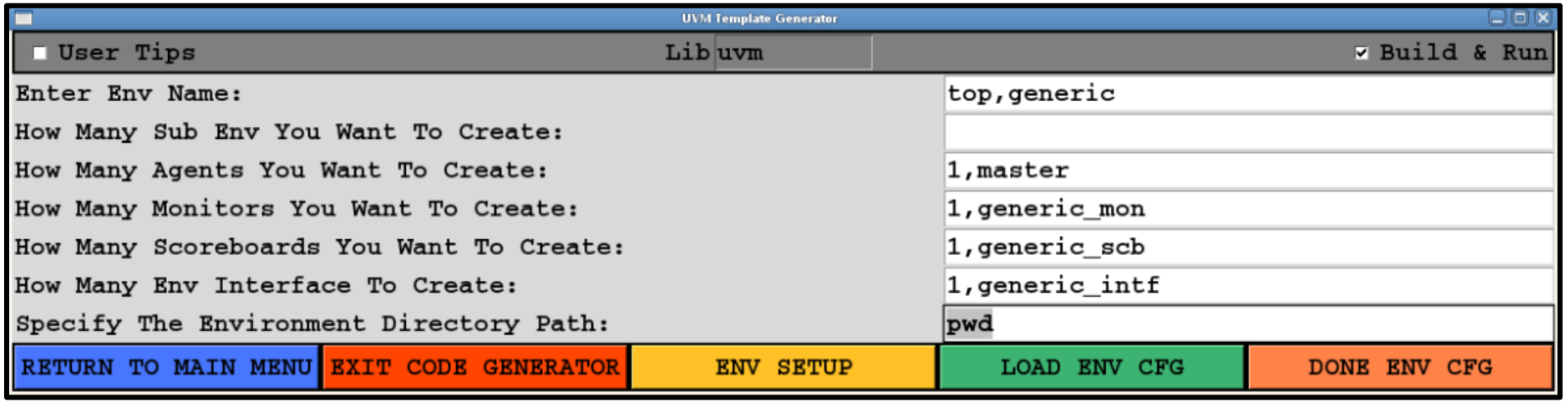


- User can choose ‘GUI Approach ’to enter the VIP details in the tool provided GUI layout as shown below.

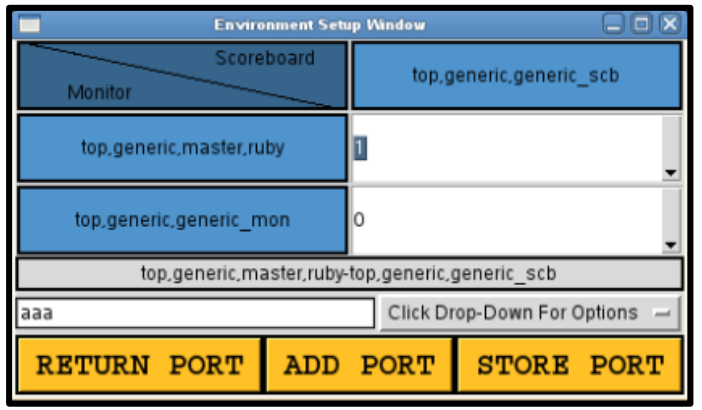


Complete UVM VIP Continued..

- On the below snapshot you can find the tool filled with the required environment details based on user input.

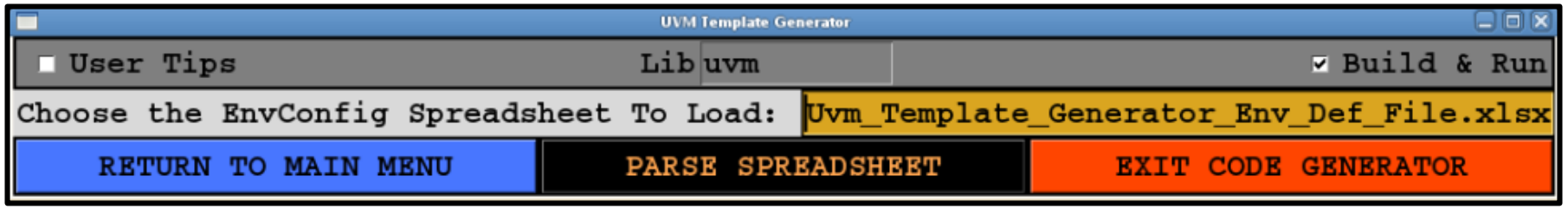


- Once user completes entering the details, they can click the 'FINAL ENV SETUP' to make required connectivity between the specific monitor and scoreboard.



Complete UVM VIP Continued..

- The other option is, user can choose the ‘Load Spreadsheet Approach’ to feed in the tool with the entire environment details provided in spreadsheet as shown below.



- Below snapshot shows the format of the “Spreadsheet” format which the user can load into the tool to generate the code.

	A	B	C	D	E	F	G	H	I	J	K	L
1	EnvironmentNo	Environment	ParentEnvironment	SubEnvironment	Monitor	Scoreboard	Agent	AgentConfig	MonScbCon	EnvIntfDetails	Directory	EnvCfgFilePath
2	0	aloha	top		1,amon	1,ascb	2,master,slave	master,5	aloha,slave,bla-aloha,ascb-red,2/blue,1	3,blu,bla,ble		
3								2,red,ruby		1-Jan		
4								1,mon				
5								1-1				
6								slave				
7								1,blu				
8								1,bla				
9								1				

Results

- The tool helps in massive (99.88%) time reduction in bringing up the initial UVM VIP template development cycle which is compile clean, properly commented and ready to use as compared to the code developed by an engineer (Refer Table below)

Developed By	Time
Engineer	8 Hours [1 Business day]
GUI Based UVM Automation Tool	0.0083 Hours [30 seconds]

- Apart from that, the tool brings in
  - Standardization of code development and re-usability of the code across the projects and helps in complete integration of the verification collateral.
  - With its unique ‘Create & Stitch’ feature, the tool can add new components or add connection between component's to the already existing codes.

Questions!