Automation Desk: A Universal Tool for Creating & Managing Automation Tasks

Abhishek Shukla, Lina Nath

Abstract

The increasing complexity of control software means that testing has to be automated. Control Desk can issue signals to the ECU under test when a test Engineer operates the panels, this could have been fine in a scenario where only one time testing is to be done, but in the Automotive world HIL is used for testing of the ECU's under development, it calls for running the test cases several times and hence requires that the test mechanism should be repeatable. Automation Desk serves this major requirement.

Keywords: **Automation Desk, Python, Hardware in Loop, Engine Control Unit.**

Introduction to Automation Desk

Automation Desk is a universal tool for creating and managing automation tasks. The test process can be made more efficient by making use of Automation Desk. It provides a graphical editor for describing control flows and parameterizes them using programming language "python".

Test process without Automation Desk

In parallel to the ECU development process, there is the ECU test process. The activities are *Test development*, *Test execution*, *Result analyzing* and *Report generation*. Below figure represents the Test Process without Automation Desk which requires and creates a lot of data. Managing this amount of information normally requires more than one software tool.

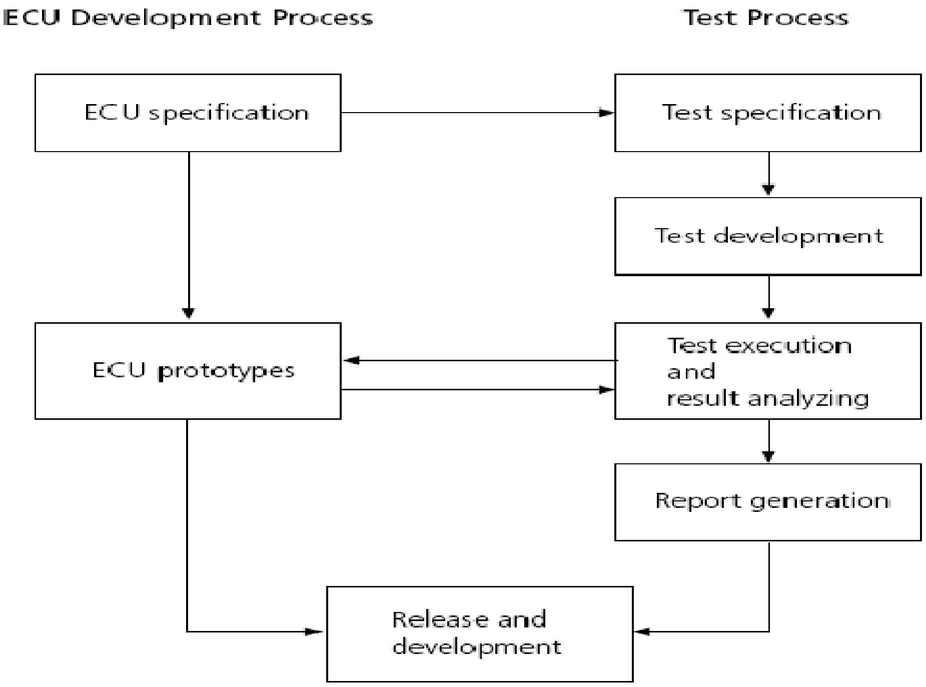


Figure 1.Test Process without Automation Desk
Courtesy: ADGuide.pdf (Automation Desk Reference Manual)

Test process with Automation Desk

The handling of these activities e.g. Test development, Test execution, result analyzing, report generation is now combined in one software tool called Automation Desk. With Automation Desk one can manage all the data one needs for testing. One can create and specify control flows and test parameters, execute tests and log the results. All execution results are stored internally. They can be exported as XML data and stored in an HTML or PDF file on the fly. It provides structured view of the whole project. It is guaranteed that one can identify and reproduce tests.

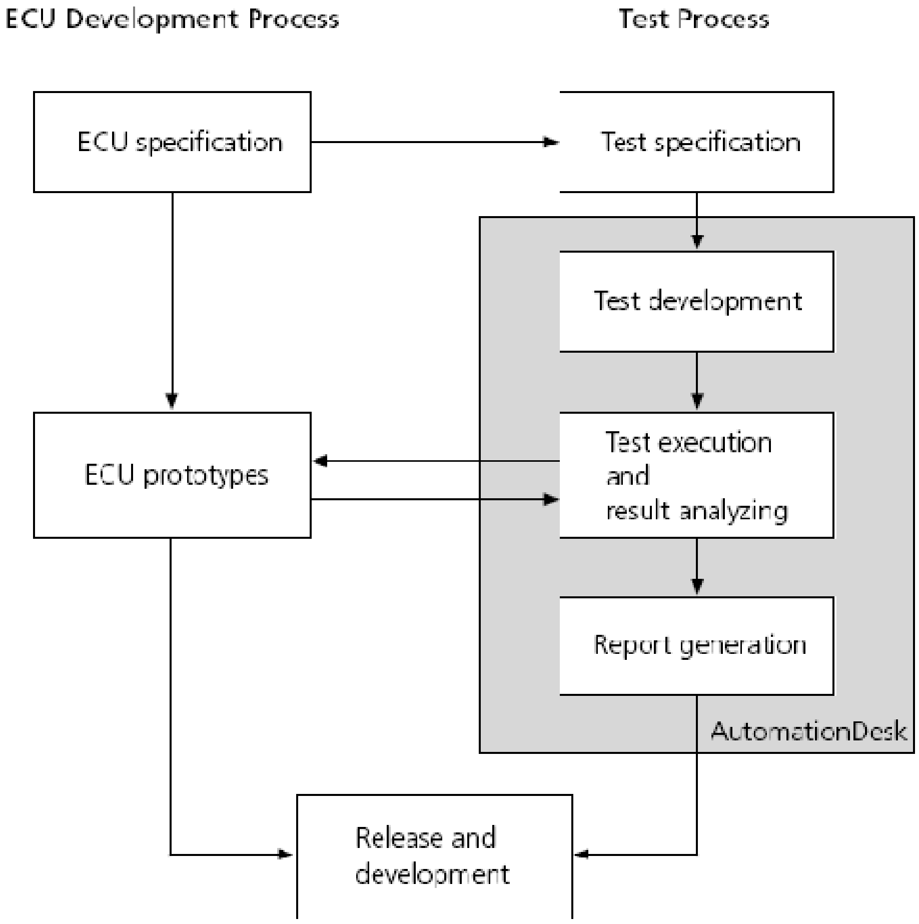


Figure 2.Test Process with Automation Desk
Courtesy: ADGuide.pdf (Automation Desk Reference Manual)

Testing scripts are written using “Python” scripting language under execution block.

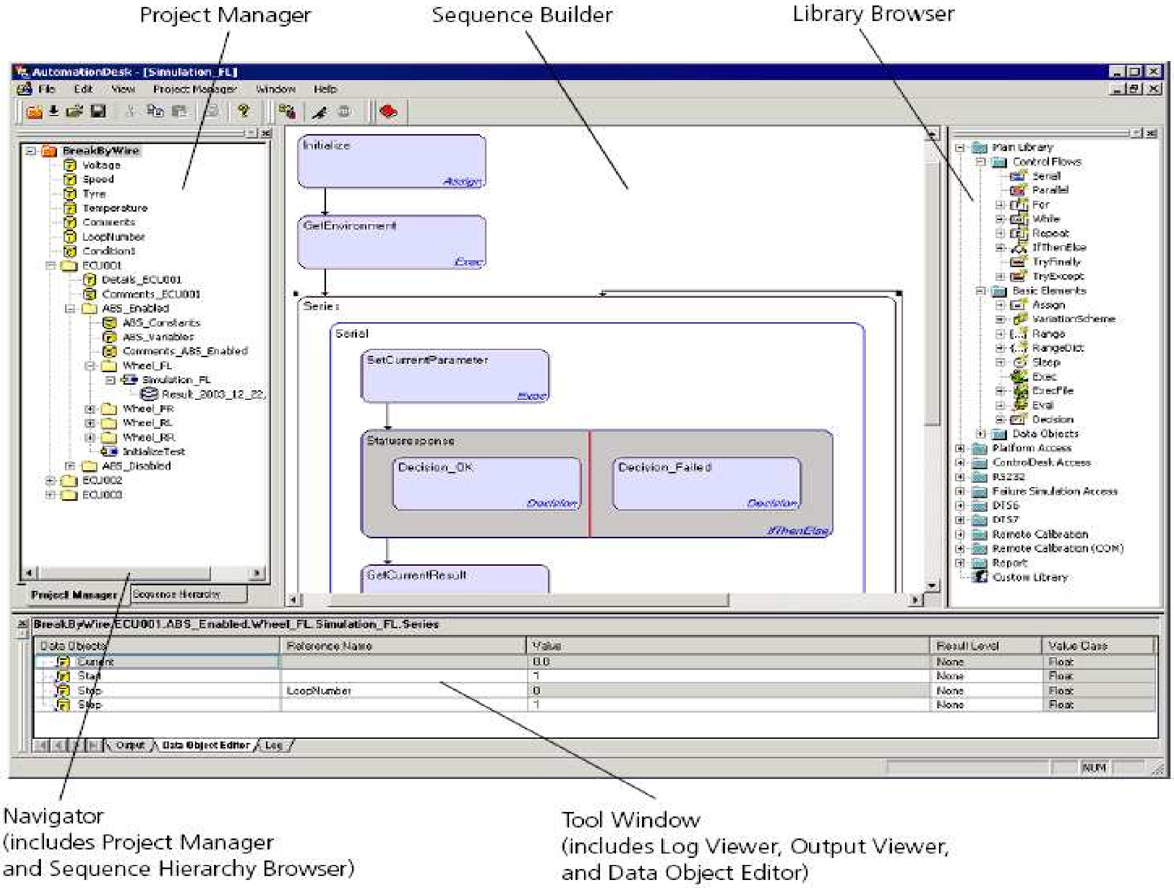
In this project, script is written to run bunch of test cases. So automation desk executes bunch of test cases at single go!!!

User Interface of Automation Desk

The following section describes individual elements of Automation Desk in brief.

Navigator: The Navigator window provides the Project Manager and the Sequence Hierarchy browser.

Project Manager: The Project Manager gives a hierarchical view of the project elements. One can use its functions to manage automation project.



Data Objects	Reference Name	Value	Result Level	Value Class
Current		0.0	None	Float
Start		1	None	Float
Stop	LoopNumber	0	None	Float
Step		1	None	Float

Figure 3.Automation Desk Windows

Courtesy: ADRefguide.pdf

Sequence Hierarchy Browser: Clicking the Sequence Hierarchy page of the Navigator displays the structure of the sequence that is open in the Sequence Builder. One can expand and collapse the sequence tree by clicking the tree nodes. This window can also be used to edit the properties of automation blocks and create custom library elements by dragging a sequence element to the Custom Library folder of the Library Browser. When one uses the Data Object Editor in single mode, its content is automatically synchronized to the selection in the Sequence Hierarchy Browser. Depending on the settings in the General Properties dialog, the Sequence Hierarchy Browser contains the structure of the selected sequence with the blocks' data objects displayed or hidden.

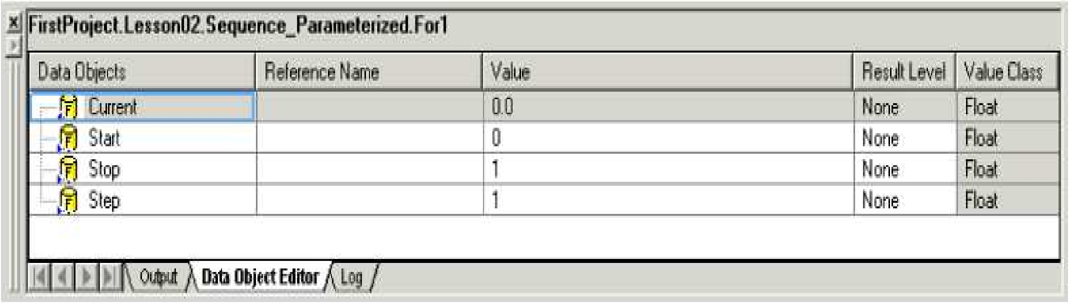
Library Browser: The Library Browser provides a wide range of elements for automation tasks. There are automation blocks for creating control flows with basic elements, and data objects for defining project-specific parameters or variables. Several standard libraries are included for accessing a platform, accessing the serial interface or working with diagnostic tools. The Custom Library folder can be used to integrate user-defined library elements.

Tool Window: The Tool Window provides the Log Viewer, the Output Viewer, and the Data Object Editor, if one uses it in single mode.

Log Viewer: In the Log Viewer one can see Automation Desk's log messages, for example, infos, warnings and error messages.

Output Viewer: In the Output Viewer one can see the specified output from print commands, but also error messages of the Python Interpreter. It can be used for debugging purposes.

Data Object Editor: With the Data Object Editor, one can view and edit the data objects of the selected element.



FirstProject.Lesson02.Sequence_Parameterized.For1

Data Objects	Reference Name	Value	Result Level	Value Class
Current		0.0	None	Float
Start		0	None	Float
Stop		1	None	Float
Step		1	None	Float

Figure 4.Data Object Editor
Courtesy: ADRefguide.pdf

Limitations of Automation Desk

- If the model changes, the "path" to that variable changes, and then test case needs to be updated.
- Tracing of the graphical program is tedious with increase in the number of variables.
- The maximal number of automation blocks executable in parallel depends on the operating system's maximal number of threads.

References

1. dSPACE GmbH, "Automation Desk Guide" Release 5.0, 15 – 212, November 2005.
2. dSPACE GmbH, "Control Desk Experiment Guide" Release 5.0, 17 – 75, November 2005.
3. Mark Hammond, Andy Robinson, "Python Programming on Win32" First Edition, O'Reilly, PP 15 – 45, 67 – 79, 351 – 369, January 2000.
4. Alex Martelli, David Ascher, "Python CookBook" First Edition, O'Reilly, PP 4.1 – 4.27, 11.1 – 11.12, July 2002.
5. Society of Automotive Engineers, "J1939_73_SAE_Mar04_Diagnostics" Rev March 2004, PP 11-23.