AutomationDesk: Positive Feedback

Behr-Hella Thermocontrol GmbH develops and produces electronic control units and manual controls for vehicle air conditioning. To ensure that the development of control strategies runs smoothly, the company chose to integrate dSPACE tools into its development process. The latest addition to Behr-Hella's tool chain is AutomationDesk, the dSPACE software for handling ECU tests that was launched in August last year. Stefan Wanoschek from Behr-Hella Thermocontrol GmbH spoke to dSPACE NEWS about his experiences with AutomationDesk.

▲ *Stefan Wanoschek, Test Engineer at Behr-Hella Thermocontrol GmbH: "The tool is chic, and gives us lots of potential."*

What do you use AutomationDesk for?

To construct software tests with as full a coverage as possible. We have requirements specifications and test our control functions against them. AutomationDesk is mainly used for regression tests, often overnight or on weekends, to check the functionality of specific software versions according to our release plans.

What's your view of the working procedures that AutomationDesk involves?

The solution is generally very good and well thought out. For example, the visibility rules for the data objects mean that general data can be stored higher up in the project tree, with more specific data further down.

AutomationDesk's features range from graphical description to Python module integration. How do you utilize these?

Obviously, graphical programming is also programming. Test creators still need knowledge in programming control and data flows. But with AutomationDesk, you can simply use graphical blocks for actions such as embedding Python functions. This makes it easy to keep track of test sequence structures, even enormously complex ones.

How do your developers learn to use AutomationDesk?

You have to distinguish between developers who have experience with modeling and simulation techniques in testing, and developers who do not. The ones with no experience are often a bit nervous at first about working with simulators without real loads – because you also have to understand and be familiar with the peripherals. But once they've overcome this inhibition, learning runs smoothly. No compilation is needed, and each block can be executed individually from the editor, so you can get to know the tool, the model, and the test bench interactively.

What are AutomationDesk's greatest strengths?

You can just switch it on and go. You can produce your first executable test sequences very fast, using AutomationDesk's graphical description features. Even so you still have to ensure that sequences are well-structured so that individual subsequences can be reused. Another of AutomationDesk's strengths is the Custom Library, with all its potential for reusing tests – even across different projects. Further features we would like to have in the future are the ability to import and export parts of projects and libraries, distributed teamwork, and integration of a versioning tool to manage test sequences.

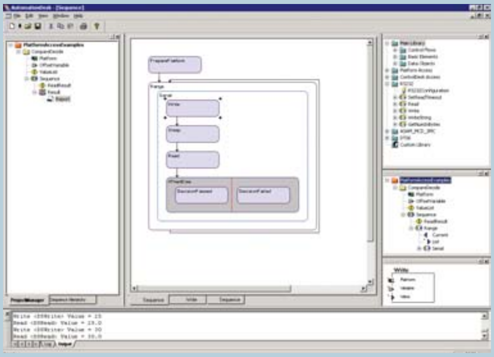
Does AutomationDesk interact smoothly with your other tools?
That mainly depends on how well a developer knows the other tools. Where developers are familiar with the tools – such as the Diagnostic Tool Set DTS 6 from Softing – and the peripherals, everything runs very smoothly.

Do you plan to use AutomationDesk in other areas?
Yes. We're interested in using it in preproduction and production development. We are already using dSPACE's TargetLink for automatic code generation in these areas. These are also potential application areas for AutomationDesk. For example, we might use MTest, the AutomationDesk extension, for systematic test development.

And finally: What's your overall impression of AutomationDesk?
We wouldn't have gotten where we are today if we hadn't had AutomationDesk. The tool is simple to use, and is still evolving. It's exactly what we need. Cooperation with dSPACE during the beta phase was also excellent. We could make suggestions which were then correctly implemented in the tool. Moreover, AutomationDesk is open and extendable. For example, it's easy for us to integrate our own Python modules. The tool is chic. It opens up plenty of potential. And provides ways of arriving at solutions.

Mr. Wanoschek, thank you for talking to us.

About AutomationDesk



```
Write <DSWrite> Value = 25
Read <DSRead> Value = 25.0
Write <DSWrite> Value = 30
Read <DSRead> Value = 30.0
```

AutomationDesk is used to handle ECU software tests. Its main features are graphical test description, test management, and a function library that is a "storehouse of know-how".

Intuitive, Graphical Test Description

Graphics-based working procedures make test description intuitive. The tool provides access to Simulink® and real-time simulations, so testing can be done both during hardware-in-the-loop simulation (HIL) and in early phases of function design. This offers great potential for reusing existing test sequences.

Project Manager for Good Organization

Structuring test projects is simple with AutomationDesk's Project Manager. It is also easy to handle several test projects, together with data and results. Moreover, the Project Manager automatically generates test reports on an XML basis.

Libraries Store Know-How

The integrated library contains a wide range of functions (for example, for all kinds of calculations) and can easily be adapted to individual requirements. Access to external software tools via DLL or COM interfaces is also provided – for example, to MATLAB®, Microsoft Office tools, and the Diagnostic Tool Set DTS 6 from Softing.