

Software Trends in Automation

By Rob Harkness, Process Analysis & Automation, Falcon House, Fernhill Road, Farnborough, Hampshire, UK, rob.harkness@paa.co.uk

Introduction

Laboratory automation software used on biochemical assays require flexibility. As with any software package, increased flexibility typically means there is an increased level of complexity. The main problem is pre-dominantly the interface with the user, the Graphical User Interface (GUI). Other aspects such as the scheduling method, static, dynamic or event driven are important. However, it is the interface that will ultimately determine if chemist or biologist will be able to use the software.. If the software is complicated, it is only expert users that will be able to use the software . A simple to use GUI will encourage all users to add their assay onto the automated system.

Solution

The *Overlord2* software that has been developed by PAA uses a flowchart based interface. A flowchart clearly defines the process flow for even the most

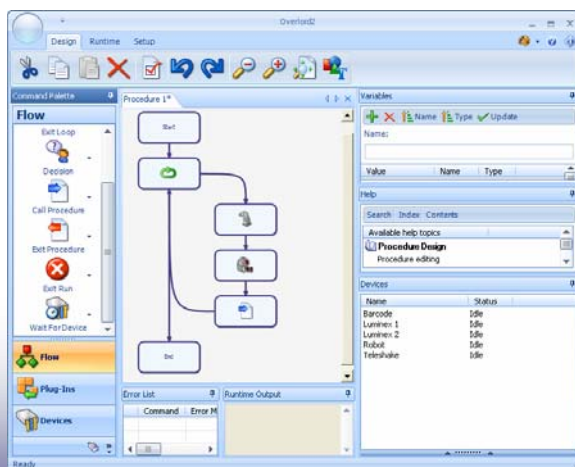


Figure 1—Overlord2 software

complicated assay. The flowchart is used in many fields for describing a process and is instantly recognizable to anyone working in a scientific field. This makes the flowchart an ideal choice for forming the core UI of any laboratory automation software .

A flowchart is a simple way of describing the scheduling constraints. By using flow operators such as “Decisions” and “Loops”, an event driven process can easily be defined and modified as part of any process optimisation that is carried out once the automation platform is setup.

Architecture

The flowchart is a key component of the *Overlord2* software package. It is the mechanism for representing what series of tasks will be carried out to automate a laboratory process. The flowchart has a dual functionality.

- It is the link between the user and the software application to allow the user to define the process.
- It is the representation of what will happen when the user decides to run the flowchart.

The Graphical User Interface (GUI) architecture is an important aspect of the software design. A good architecture structures the application in a way where it is clear what role each section of code undertakes.

The Model-View Presenter (MVP)¹ design pattern has been used in the *Overlord2* software and has been extended to include the software components that run the process and control the instruments.

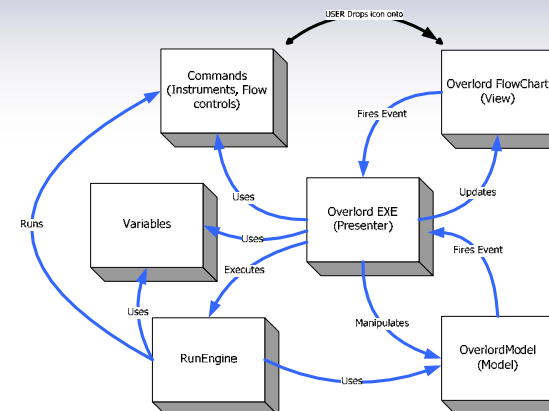


Figure 2—Overlord2 MVP Architecture

Extensibility

The software architecture that has been used for *Overlord2* means the product can be modified to fit the exact requirements if the basic version is not suitable. Known as the *Overlord2 Framework*, the key *Overlord2* components can be used within a customized front-end application (The Presenter in the MVP architecture) whilst still incorporating the Flowchart interface.

There is a trend with all automation packages to provide extensibility. Being able to customize key components of the product is an extremely powerful tool. This functionality provides a way to deliver laboratory automation solutions that fully meet the customer requirements.

References

1. MVP: Model-View Presenter The Taligent Programming Model for C++ and Java <http://www.wildcrest.com/Patel/Portfolio/mvp>

