

EVOLUTION OF VISUALDCT

J. Bobnar, M. Sekoranja, I. Verstovsek, Cosylab, Slovenia

Abstract

VisualDCT (Visual Database Configuration Tool) became the most advanced and popular graphical EPICS database configuration tool for creating, editing and debugging EPICS databases. EPICS is a widely used control system based on a real-time database configured via ASCII files. The most recent development achievements in Eclipse IDE, which is also a RCP application portable to many operating systems since it is written in Java, brought another perspective to development of VisualDCT. Using Eclipse GEF (Graphical Editing Framework) for graphical features and EMF (Eclipse Modelling Framework) for database code generation makes it possible for VisualDCT to become a part of the Eclipse IDE. Using Eclipse as the framework for application automatically adds common features as plug-in support, debugging tools and many others. In addition, VisualDCT could be used as a part of the CSS (Control System Studio) allowing easy handling of EPICS databases using MB3 (mouse button 3) and drag and drop functionalities.

OVERVIEW

Visual Database Configuration Tool (VisualDCT) has been designed to create and maintain EPICS record instance (.db) database files. EPICS is a widely spread control system which can be nowadays used to control either a simple system with only a few devices as well as a system as complicated as an accelerator. Such distributed control systems typically comprise tens or even hundreds of computers, networked together to allow communication between them and to provide control and feedback of the various parts of the device from a central control room, or even remotely over the internet. In order to maintain such a complex system, powerful development software is required; VisualDCT's main objective is to offer intuitive development environment to the EPICS database engineer. With its sophisticated graphical design VisualDCT offers an instant intuitive view over the complex database structure and is a great step forward from text-based database editors.

HOW EVERYTHING STARTED

VisualDCT began as a project funded by Swiss Light Source (SLS). Initially, the idea behind it was to develop a tool providing missing features of CapFast and Graphical Database Configuration Tool (GDCT). In the beginning VisualDCT very much resembled GDCT. Following this initial development, soon additional features like adding comments to graphical object, postscript printing, etc. and the most important among them the hierarchy support were added. Introduction of hierarchies brought a new aspect to EPICS database configuration and has in the last few years become de

facto of database configuration as it is much easier to maintain a hierarchically constructed database than a flat one.

In the following years, in collaboration with various laboratories from around the world the JCA Debug Plugin was introduced. With use of JCA and CAJ Java libraries Debug plug-in enables displaying actual values in the control system inside the control boxes for records displayed within VisualDCT. This enables user to debug the database already at the time of creation.

Following the needs of various labs, who used CapFast prior to VisualDCT, a plug-in that enables conversion of CapFast databases to VisualDCT format was developed. The introduction of this plug-in overcame the barrier between the two applications and made it possible that even existing databases could be ported to and maintained with VisualDCT, which as being primarily developed for EPICS database configuration, became more and more appealing.

The latest updates of VisualDCT were made possible in collaboration with Diamond Light Source (DLS). In a desire to extend the application not to be just The EPICS database configuration tool, but also a tool which would allow generating and maintaining other configuration files, additional plug-ins have been introduced. Alarm Handler plug-in enables user to easily configure EPICS Alarm Handler configuration files, which are required for proper propagations of alarms inside the control system. Using this plug-in the wide structure of configuration files become very easy to maintain. In addition to Alarm Handler, Channel Archiver plug-in has also been developed. Using this plug-in one can easily create a new one or parse and fix an existing xml configuration file.

In addition to these plug-ins, numerous visual and functionality upgrades have been introduced in 2005.

FUTURE PLANS AND DEVELOPMENT

Spreadsheet View and Editor

The majority of EPICS databases contain a large number of records. In these cases, the graphical view is not very logical and intuitive and therefore, it is difficult to maintain the databases with ordinary VisualDCT's tools, not to mention ordinary text editors. Bob Dalesio from SLAC proposed to introduce a new view of records – spreadsheet style. Instead of graphical representation, records would be presented in the table. The table would allow easy configuration of multiple records at the same time, while the graphical layout would be updated in the background, therefore allowing visual inspection of introduced changes. The functionality of the spreadsheet view could also be extended to support import/export of data from/to excel.

Integration of VisualDCT into Eclipse

Eclipse Rich Client Platform (RCP) is an extensible open tools platform to build just about any client application. Eclipse IDE, which is also a RCP application, has become one of the most used IDEs. It is portable to many operating systems since it is written in Java, but still provides native user interface look and feel and speed. In the EPICS community there is a strong interest to use it as a base of an EPICS Office, which would include also VisualDCT.

One of the main advantages of using Eclipse as the platform for VisualDCT is the possibility of using Graphical Editing Framework (GEF), which could in combination with Eclipse Modelling Framework (EMF) replace its core part.

GEF provides a powerful framework for creating customized visual editors while being model agnostic [1]. There are many options how to provide an efficient model and once this is done it allows one to easily develop graphical representation. All graphical visualization in GEF is done via the Draw2D framework based on Standard Widget Toolkit (SWT). The framework already provides the most needed features, such as effective layouts, multiple layers, printing, etc., which makes it

efficient and well tested solution for the renewed VisualDCT.

In addition to graphical representation, GEF also supports the use of graphical editors for nearly every model. Editors can be used to make simple modifications like changing element properties as well as more complex operations. Many of these modifications can be done in graphical editor using common functions like drag and drop, copy and paste and actions invoked from menus and toolbars. GEF employs a model-view-controller (MVC) architecture, which enables simple changes to be applied from the view.

Furthermore, GEF already supports some of the features, which proved useful in VisualDCT, such as undo/redo, direct editing via property sheet (which could be a nice base for the spreadsheet view and editor), rulers, snap to grid, etc.

The model independency of GEF brings the advantage that one can specify the model he desires. The majority of models are usually based on Plain Old Java Objects (POJOs). However, an alternative and the most appealing solution would be to use EMF, which can easily be adapted to use an existing POJO model or it can provide its own. From a model specification described in XMI, EMF provides tools and runtime support to produce a set

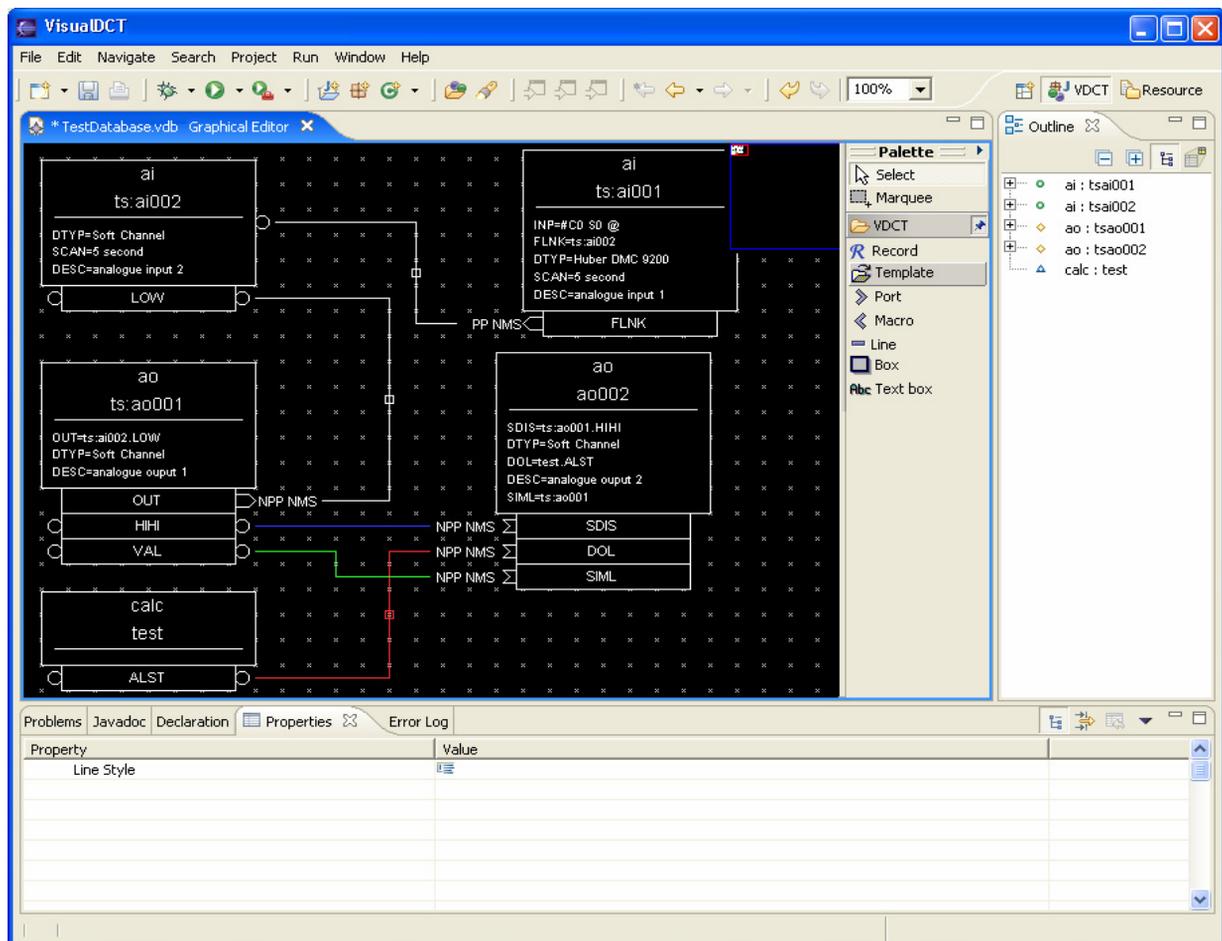


Figure 1: VisualDCT integrated into Eclipse framework.

of Java classes for the model and a set of adapter classes that enable viewing. Command-based editing of the model and a basic editor are also provided by EMF.

EMF consists of three separate modules: the core EMF, which allows describing a model and offers runtime support, the EMF.Edit framework includes generic reusable classes for building editors for EMF models and the EMF.Codegen is a generation facility capable of generating everything needed to build a complete editor for an EMF model. The EMF also employs different levels of code generation, which can extensively be used for automatic EPICS database generation.

Another aspect that might contribute its share to development of VisualDCT is also the Graphical Modelling Framework (GMF), which is the bridge between aforementioned EMF and GEF. However, the downside of GMF is that it creates far more complex code, which is not easy to understand and edit, thus making it not very acceptable for the community developed project. On the other hand GMF offers a handful of UI resources which could be extended to serve the needs of VisualDCT.

Integration with Control System Studio

Control System Studio (CSS) is a powerful application based on the Eclipse framework. It was primarily designed as a reach client platform for EPICS, offering all the necessary tools for maintaining EPICS configurations as well as offering operator the control over the system. Since VisualDCT is one of the most important tools in the development of such control system it is desired to adapt it to run with the CSS. The aforementioned upgrade to Eclipse framework would introduced a great benefit for CSS.

In addition to that, some alternative solutions have also been investigated. Introduction of Drag and Drop (DnD) would offer easier handling of databases and records, since VisualDCT could be used in combination with other EPICS tools that provide DnD. This feature becomes even more pronounced once VisualDCT becomes an Eclipse plug-in, as complex object could then be easily transferred between different plug-ins.

Another proposed feature was also the use of mouse button 3 (MB3), which would similar to DnD enable easier transfer of data between applications. VisualDCT will act as a receiver, allowing import of records and databases via MB3, as well as a source, which would enable loading the database and start the IOC or launching an application such as JProbe on a particular record for example.

Integration with APS's VCCT

In the recent years an application called VCCT (Visual Connection Configuration Tool) has been developed by the Advanced Photon Source (APS). The purpose of this tool is to present the structure of the EPICS control system in a more human friendly form. For this reason a relational database containing the entire set of EPICS records has been created. By using VCCT one can

construct the hierarchical tree of all devices and their connections in the system. Such representation becomes very useful when inspecting the system with several thousands of records.

Because VCCT is written in Java as is VisualDCT, there have been ideas to combine the two applications. VCCT would provide the list of records in the system. When records from the VCCT would be dragged and dropped into the workspace they would be recognized by VisualDCT and graphical representation of those records and their relations would be generated automatically. This would allow easy inspection and debugging of a desired part of the database. All changes made to the records and their fields would then be reflected in the relational database upon saving the newly made configurations.

CONCLUSION

Taking into account all aforementioned features we can conclude that VisualDCT is becoming not only database configuration tool but The configuration tool for EPICS control systems. With world-wide support it can be improved even more, when the requirements of various users are taken into account and therefore, its position in the community could be made even stronger.

ACKNOWLEDGEMENTS

We would like to express our gratitude to all the people involved on Visual DCT project, especially the funders of all the development packages. We would also like to encourage all the labs to contribute to the community and sponsor further development of the application.

REFERENCES

- [1] <http://www.eclipse.org>
- [2] J. Bobnar, I. Verstovsek, and M. Sekoranja, VisualDCT's Latest Hits (ICALEPCS 2005).
- [3] M. Sekoranja et. al., VisualDCT User's manual.
- [4] M. Sekoranja, VisualDCT design, GEF, EMF, GMF feasibility study.
- [5] M. Sekoranja, VisualDCT design, CSS Additions.
- [6] N. D. Arnold, D. A. Dohan, Connection-Oriented Relational Database of the APS Control System Hardware.