VDCT Status

Nick Rees

(with lots of help from the Cosylab guys: Igor Verstovsek, Matej Sekoranja, and Jaka Bobnar)

Introduction

- Goals
- What has happened
- Current problems
- What's going to happen
- Next set of requirements

Introduction

- Goals
- What has happened
- Current problems
- What's going to happen
- Next set of requirements

Goals

- VDCT to become THE database configuration tool
- Don't turn people off by burning them with a substandard product
- VDCT to be an “Open” product

Goals

- VDCT to become <u>THE</u> database configuration tool
- Don't turn people off by burning them with a substandard product
- VDCT to be an "Open" product

VDCT status

- Yes – it’s all a sham – John MacLean gave this talk at the Abingdon EPICS meeting 2 years ago!
- I took over VDCT support in the beginning of April 2005.
- What has happened?

Had we met the goals?

- I didn't know, but my impression was that we maybe weren't there yet.
- Tried a simple survey on tech-talk
  1. Site name.
  2. Approximate number of (EPICS) developers
  3. Approximate % of databases developed recently that use VDCT.
  4. Likelihood of contributing to VDCT development - either through cash or effort.
  5. Any complaints, praise, wishes etc...

Survey results

Site	#EPICS developers	% VDCT	Help?	Notes
IPNS, ANL	4	100%	No	Runs slowly over X, so use vi at times
USPAS Training	25/course	100%	No	
GRETINA, LLBL	1	100%	No	
APS	9	80%	No	GDCT users. Improve documentation. Enumerated many bugs
Obs. Sciences	4	20%	No	Improve documentation. They use whatever tool the contract specifies
BESSY	12-15	<10%	No	Uses Capfast. VDCT needs more Capfast like features
JAC	5	<10%	No	Uses Capfast
SPEAR	4	8%	No	20% of db's are GDCT and need to change
PPPL	2	5%	No	Mainly GDCT, plan to switch sometime
ALS	?	Evaluating	No	

Survey conclusion

- VDCT had plateau'ed in a state where it was an adequate GDCT replacement, but an inadequate Capfast replacement.
  - Was a good basic tool but no match for a hierarchical schematic capture package for complex systems.
  - On the plus side, the "knowledge" of EPICS gave it a few capabilities that a generic schematic capture couldn't match.
  - Support for hierarchy was too buggy to be used.
  - Wiring and printing support was quirky.
  - No-one was willing to help
- In other words:
  - It was not really <u>THE</u> database development tool
  - It did turn people off by being a bit buggy
  - It was not being adopted as an "open" product

What to do?

- Drop it in favour of something else?
  - But what?
  - Capfast? - too expensive for many sites
  - Only other suggestion was an open source package investigated by Andrew Johnson a few years ago.
- Address the issues directly?
  - Don’t focus on new features, but fix bugs (boring) and address schematic capture issues.
  - Encourage open nature by ensuring build and documentation is improved and maybe doing some internal work.
- *We need to change the approach from previous work packages and focus on solidifying, not developing*

What did I do?

- Collected a comprehensive list of bugs an features
- Eliminated those that may be funded by others.
- Ordered the rest in the following priority order:
  1. Bugs or functionality essential to a schematic capture tool (i.e. would be seen as a bug in a schematic capture tool).
  2. Useful functionality that improves VDCT as a database configuration tool
  3. Useful functionality, but may require greater thought or co-ordination with EPICS V4 development
  4. Potentially useful functionality, but not too essential to the core function of a DCT. Also irreproducible bugs
  5. Functionality that is not required since the solution is already available or provided in another way.
- Found funding for a work package to address items 1 and 2, with 20% contingency for more bugs.

Work package details

- 22 bugs and 15 features
- Example features were:
  - Drawing borders
  - Improve ease of wire re-routing - eliminate wire handles so you can grab anywhere along a wire.
  - Have dots where wires join
  - Allow morphing of templates as well as records
  - Highlight which record is being inspected
  - Enlarge record under mouse focus at low zooms
  - Control over which side of the record wires emerge – allow wires to connect to both sides of a field
  - Eliminate small steps in wires.
  - Enable file saving when traversing the hierarchy
  - Make building from source easier

More Work package details

- Examples of bugs
  - Printing problems - large flatbed printers were unavailable
  - Wire routing information connected to a template instantiation wasn’t saved
  - Dragging multiple objects resulted in their relative positions changing
  - Lots of graphics quirks - small steps in wires, macros sometimes changed orientations unexpectedly
  - Undo and redo don’t behave consistently.
  - Imports of databases just didn’t work
  - Pasted objects appeared in strange places
  - The addpath database function wasn’t consistent with the dbLoadDatabase definition

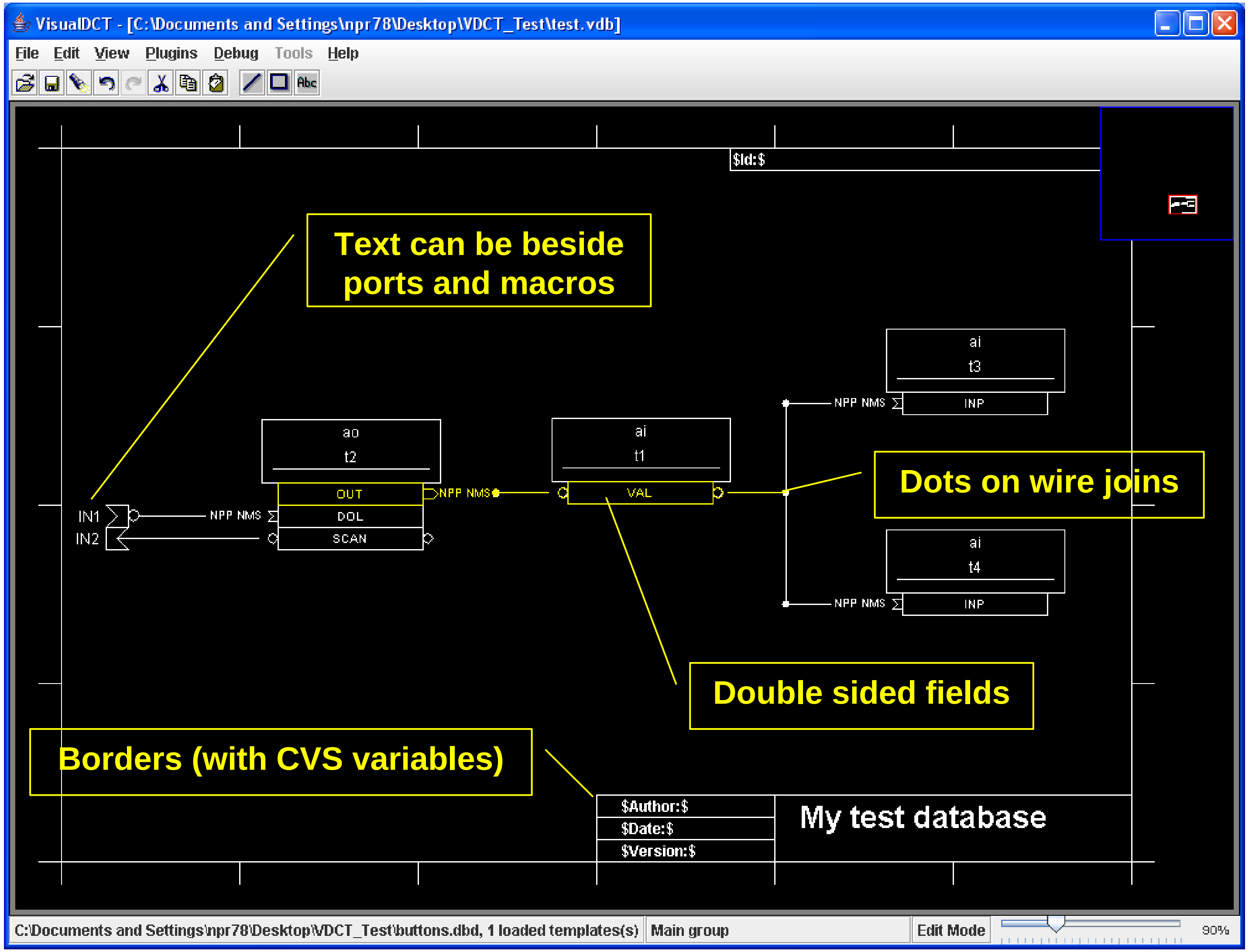
Not the Work package details

- Some significant ideas were ***not*** included
  - Ability to edit multiple record properties at once
    - we hoped this would be addressed by a SLAC package that seems to have fallen through – Bob?
  - Different shapes for different records
    - This is easy in Capfast where shapes are static, but VDCT has dynamic ‘just large enough’ shapes. It is difficult to see how to do this whilst retaining the GDCT look and feel.
    - Only idea was to have an area in the record reserved for a distinctive icon that is different for different records, but it is hard to see how to scale the icon nicely.
  - Colour customization, detachable navigator
    - Didn’t make the cut in priority.
  - On-the-fly add and delete to live soft IOC, debugging hierarchical databases, direct database interface
    - Just a bridge to far…

Work package status

- Purchase order issued in July
- 41 days effort estimated, with 9 additional days funded for more bugs – 10 weeks total.
- All except two items are addressed, but not fully tested by Diamond.
  - Some solutions are not entirely satisfactory, but there may have to be some compromise
- About 18 new bugs have surfaced in testing

Example results



C:\Documents and Settings\npr78\Desktop\VDCT_Test\buttons.dbd, 1 loaded templates(s)

Text can be beside ports and macros

Dots on wire joins

Double sided fields

Borders (with CVS variables)

Where to now (short term)?

- Matej is working on some of the last difficult problems (wire routing, cut and paste between windows).
- Diamond has to test the delivered items for conformity
  - Done once, a few weeks ago, but it is an ongoing process
- Assess the effort spent with an eye to seeing how many new bugs can be fixed within the budget.
- Document and release

Where to now (longer term)?

- A number of bigger issues have been left unaddressed
  - "Spreadsheet view" and/or editing properties on multiple records.
  - Database interface – a relational database is good for properties, VDCT is good for structure.
  - Debug view for templates
  - Metadata support
  - Formal definition of graphics format (i.e. via metadata).
- To be THE database configuration tool it must keep abreast of base developments
  - How can we represent version 4 ideas in a way that is understandable and maintainable?
  - How do we fund ongoing support?
  - Can we standardise on a way of instantiating templates?

How do we do this?

- Difficult to fund
  - Managers don’t like open ended external contracts
- Difficult to keep pace of developments
  - How to we track and anticipate V4 developments
  - I know, I just have to offer to help…
- Thinking of doing some work in house
  - Develop the open source nature of VDCT, and broaden the developer base.
  - Maybe a possibility is to do some plug-ins in house and have significant new developments done by Cosylab

Conclusions

- **We need a decent database configuration tool and VDCT is the only one I see that is realistic**
  - **Emacs is not a good DCT**
  - **Without support for hierarchies (or some knowledge structuring) complex databases are very difficult to design and build, and impossible to maintain**
  - **Other alternatives (like Capfast or maybe other different programming models) are either not mature or too expensive or outside out ability to develop along with EPICS.**
- **We have to take the development of database configurations tools as seriously as we take the development of base.**
- **Any ideas, comments, questions or suggestions?**