

RELEASE OF VISUAL WORKSHOP 2, A MODEL VIEWER, EDITOR, RUN AND RESULTS DISPLAY PACKAGE FOR THE ANSWERS CRITICALITY AND SHIELDING CODES

A. J. BIRD, T. C. FRY

Serco Technical Consulting Services
Kimmeridge House, Dorset Green Technology Park,
Winfrith Newburgh, Dorchester, Dorset DT2 8ZB, UK
adam.bird@serco.com, tim.fry@serco.com

ABSTRACT

The goal of Visual Workshop is to be an 'Integrated Development Environment' for preparing, modifying, checking, running and analysing results from the criticality code MONK [1], the shielding codes MCBEND [2] and RANKERN [3] and the reactor physics/criticality code WIMS [4]. The functionality is being introduced in stages and this paper announces the release of version 2. The key features of Visual Workshop version 1 included interactive 2D and 3D ray trace display of the model using the same routines as the analysis code, wire frame display of the model, built in editing, running and diagnostic capabilities and a simple results display.

The main additional capabilities of version 2 are:

1. Results display using 2D cell plots, 2D contour plots and 3D iso-surface plots overlaid on the model geometry for the following result types:
 - Scatter/Reflect sampling probabilities (RANKERN).
 - Importance map values and particle inflows (MCBEND/MONK)
 - Flux and Response/Dose values (MCBEND/MONK)
 - Action Tally values (MONK)
2. Display of event type, location and particle tracks overlaid on the model geometry:
 - History Log and Collision log data (MCBEND/MONK)
 - Source Log, Forced Flight log and Leakage Log data (MCBEND)
 - Birth Store data (MONK)

Visual Workshop is written in JAVA with the 2D and 3D ray trace capability provided by a 'Java Native Interface' link to FORTRAN code from MONK, MCBEND and RANKERN.

This paper describes the features of Visual Workshop as they are applicable to a criticality assessment.

Key Words: Results Visualisation, Graphics, MONK

1 INTRODUCTION

The long term goal of Visual Workshop is to be an 'Integrated Development Environment' for preparing, modifying, checking, running and analysing results from the ANSWERS criticality code MONK [1], the reactor physics/criticality code WIMS [4] and the shielding codes MCBEND [2] and RANKERN [3]. The functionality is being introduced in stages and this paper announces the release of version 2.

2 DESCRIPTION OF VISUAL WORKSHOP

Visual Workshop is written in JAVA using SWING for the user interface components with the 2D and 3D ray trace capability provided by a 'Java Native Interface' link to FORTRAN code from MONK, MCBEND and RANKERN. The wire frame graphics display is provided by the Visualisation Toolkit (VTK) [5].

Visual Workshop consists of four tabbed components; 3D Wire view, 3D Ray view, 2D Ray view and Launchpad view with an overview component on the left. Optionally a sub-view component can appear below the four main components. Figure 1 shows Visual Workshop with the sub view containing the input editor and the wire frame tab selected.

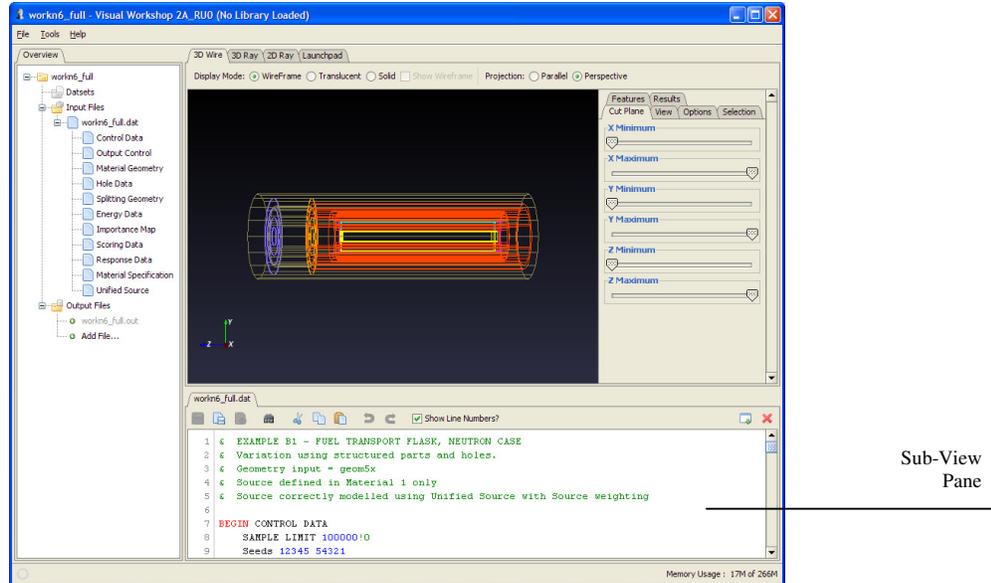


Figure 1. Visual-Workshop with the sub view containing the input editor and the wire frame tab selected.

The **Launchpad** view handles selection of the MONK code version and selection of data library versions – this is largely automatic and based on the requirements of the input data and running calculations. A run monitor is provided to ‘watch’ a running calculation and catch any messages that would appear on a console. A queuing tool is available allowing submission of a series of calculations in batch mode.

The **3D Wire** view displays the geometric bodies used to construct the model as described in the Fractal Geometry (FG) package of MONK. The display supports solid, translucent and wire frame modes, cut-away planes and selection of components to display. Additional code version features can be viewed, for example a MONK Unified Tally mesh.

The **3D Ray** view displays the model using the FORTRAN tracking routines that MONK uses to run a calculation. This means it is the most accurate representation of what the model really is. Also ‘HOLE’¹ geometries are accurately displayed. The 3D ray trace display supports cut away planes and selective exclusion or transparent rendering of components. Errors in the model can be automatically detected and displayed. Many code version features can be viewed as overlays on the display. The display is initially rendered at a low resolution then progressively refined, allowing the 3D ray trace display to be fully interactive and rendered in real time.

¹ Hole geometries in MONK use Woodcock tracking which enables potentially complex shapes to be efficiently modelled, the disadvantage with respect to visualisation is that not being defined by simple bodies can not be displayed by the wire frame.

The **2D Ray** view displays a slice through the model. In other respects it is similar to the 3D ray capability.

The **overview** component lists the files associated with a calculation and provides shortcut links to key sections of the input data.

The **Input Editor** provides a full-featured text editor including syntax colouring. Changes that are saved using this editor will automatically result in updates to the displays.

The **sub-view** component can contain the input editor, output file viewers or graphed results, among other things. These may optionally be detached as separate windows.

3 CODE DEVELOPMENTS

A lot of development work has gone into the production of Visual Workshop 2 not only to add important new features, but also to improve the features in version 1. First, improvements to existing features are covered:

The user interface has been improved to make the controls for the three display types; 3D Wire Frame, 3D Ray trace and 2D ray trace; more consistent. Feedback from version 1 has helped guide changes made for version 2.

The parser, which is used for reading and interpreting the input data for MONK, MCBEND and RANKERN, has been refined and developed to understand some of the more complex syntax available in the ANSWERS assessment codes, for example:

- Looping parameters.
- Multiple File Input.
- Processing from a 'datasets' file.

Visual Workshop uses a system of self testing, version 2 has seen improvements and extension to this set of 'auto-tests'. These are described in a section 5.

In anticipation of the next release of the ANSWERS codes MONK and MCBEND Version 2 of Visual Workshop understands and can display features which will be new with these codes.

The major new developments are as follows, and are described in more detail below:

1. Use of the Visualisation Toolkit (VTK) to provide the 3D wire frame display.
2. Results display in the form of cell plots and surface contours.
3. Event location and particle history display.

3.1 VTK graphics

Version 2 of Visual Workshop uses the VTK toolkit to provide the wire frame display. This has improved the quality of the graphical display and enabled development of the results display features. However, these improvements are not for free, version 2 has larger memory requirements and works better with more powerful graphics hardware than was required for version 1.

3.2 Results Visualisation

This is a major new feature in version 2. Visual Workshop is able to scan an output file and extract results data which can then be displayed overlaid on the model in the wire frame display.

Additional display styles have been provided; 'Neutral' displays the model in grey shades leaving the results in colour to enable easier visualization and 'Report' which is 'Neutral' with a white background for use in documents.

The following result types are supported in version 2.

- Scatter/Reflect sampling probabilities (RANKERN).
- Importance map values and particle inflows (MONK /MCBEND)
- Flux and Response/Dose values in Unified Tally (MONK /MCBEND)
- Action Tally values in Unified Tally (MONK)

These results can be plotted as a 3D contour or iso-surface (Figure 2), a 2D contour plane plot (Figure 3) or a 2D cell plot (Figure 4). The cell plot displays the results by cell as they are tallied in MONK (or MCBEND), whereas the contour plots interpolate the contour based on the mid point of a cell.

The Figures 2, 3 and 4 show total neutron flux from a simple MONK calculation with fissile material distributed equally in four drums. Figure 5 shows neutron dose inside a transport flask from a MCBEND calculation.

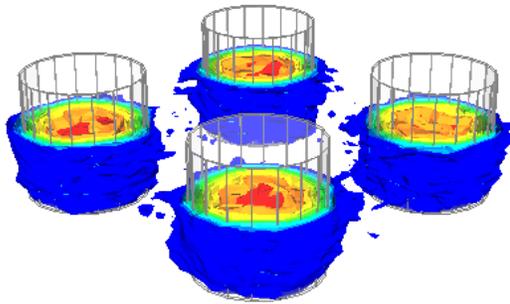


Figure 2: 3D Contour

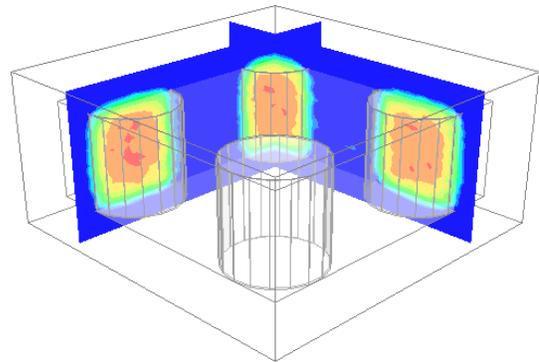


Figure 3: 2D Contour

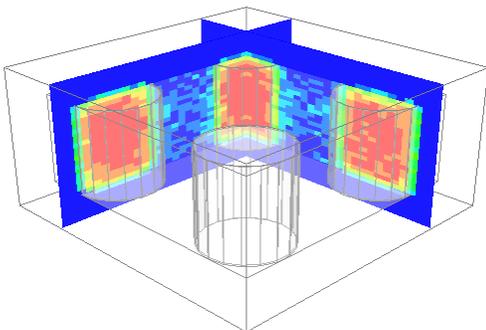


Figure 4: 2D Cell

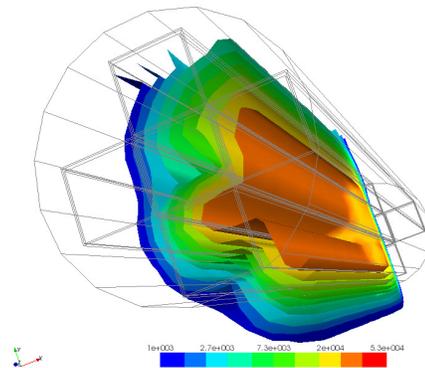


Figure 5: Neutron Dose inside a transport flask

3.3 Event Location and track visualisation

The ANSWERS codes MONK and MCBEND are able to write event locations and particle history information to a range of files. Visual Workshop version 2 is able to read these files and display the particle histories on the 3D wire frame, 3D ray trace and 2D ray trace images.

The event files supported by version 2 are:

- History Log and Collision log data (MONK/MCBEND)
- Source Log, Forced Flight log and Leakage Log data (MCBEND)
- Birth Store data (MONK)

The events and tracks can be selectively displayed by sample number, energy, weight, time, location, reaction type and nuclide or material the reaction occurs with.

Figures 6 to 8 show examples of event log displays.

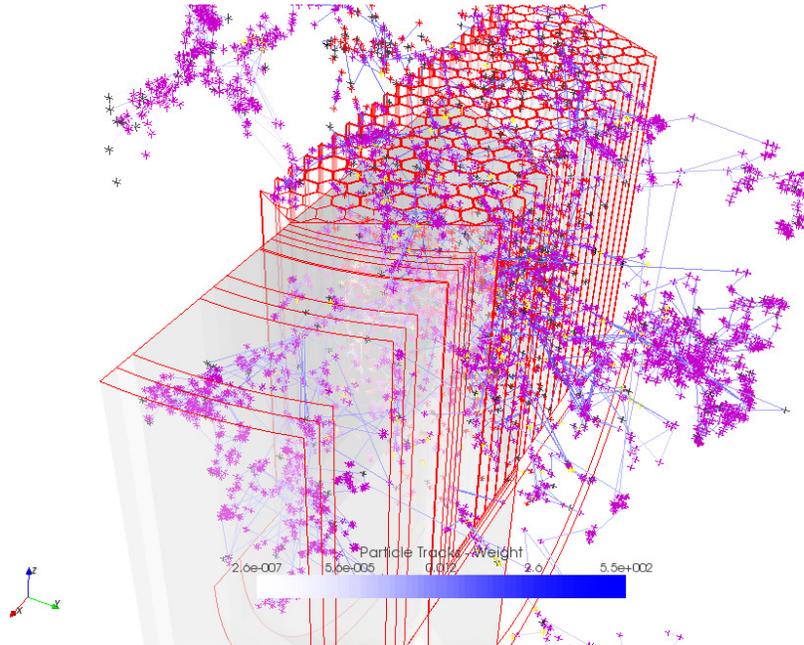


Figure 6. History log colored by event weight on translucent wire frame of reactor with geometry cut away.

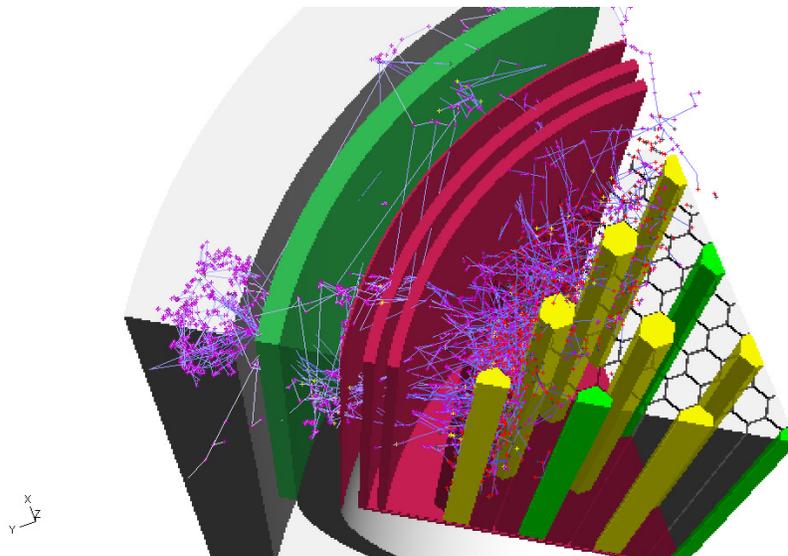


Figure 7. History log colored by event weight on 3D ray trace with cut away geometry, transparent materials and hidden line removal.

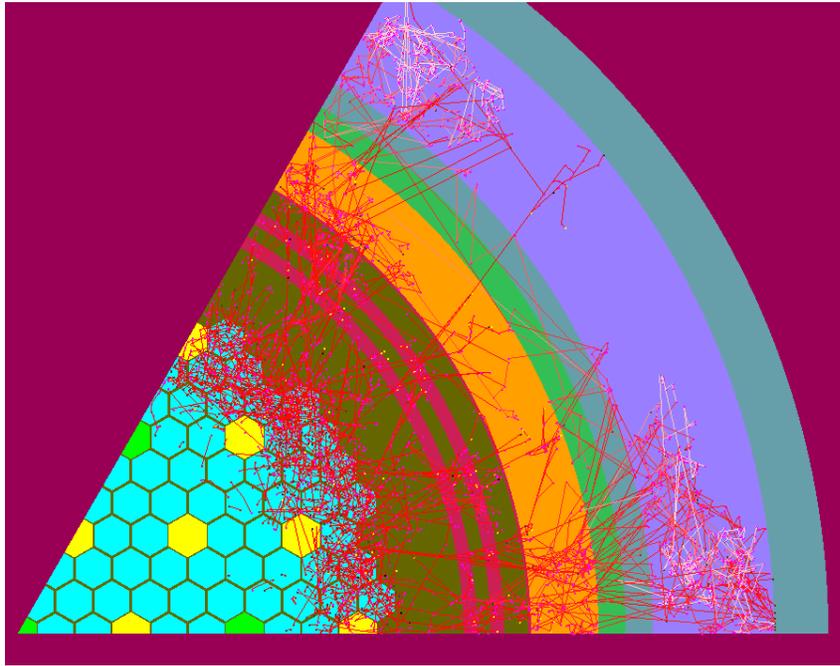


Figure 8: History log coloured by event energy on 2D ray trace.

The MONK Birth Store display provides an animate feature so that the evolution of the source distribution between stages of the MONK calculation can be monitored. Figure 9 shows a series of pictures from such an animation, highlighting the effectiveness of the MONK superhistory powering option when converging an inappropriate source guess to the steady state source distribution.

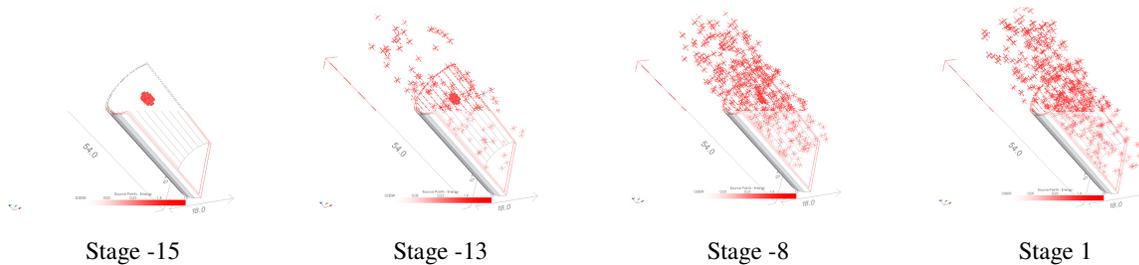


Figure 9. MONK Birth store display showing evolution of the source distribution.

4 SUPPORT FOR DEVELOPMENT FEATURES

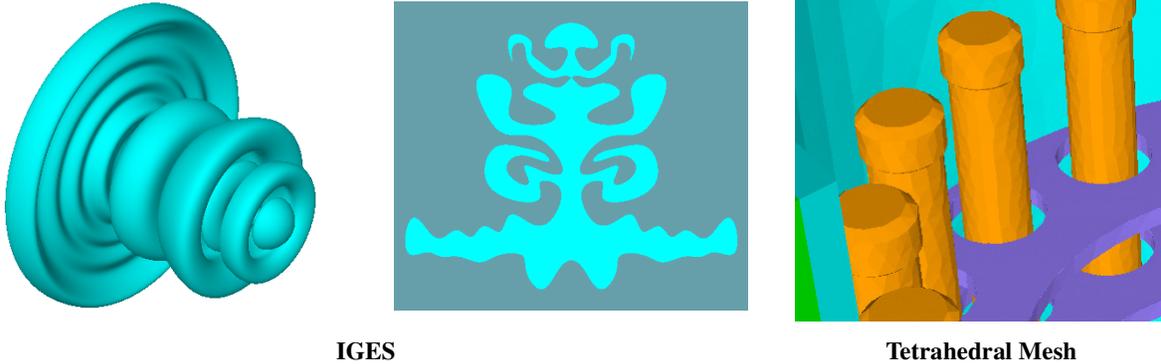
Visual Workshop provides ray traced images of the model using the particle tracking FORTRAN used by the ANSWERS codes MONK, MCBEND and RANKERN. The correct FORTRAN dynamically loaded library is chosen at runtime based on the code version selections in the Launchpad tab. The de-coupling of the GUI and the ray trace functionality allow Visual Workshop version 2A to support and test some development features.

The development features include:-

Import of CAD geometry in IGES format: This provides support for complex surfaces with no approximation.

Import of MESH geometries: This provides high performance support for complex surfaces with some geometric approximation.

Figure 10 shows examples of these imported geometries.



IGES

Tetrahedral Mesh

Figure 10: examples of imported geometries

4.1 3D Wire fame Stereo viewing

Visual Workshop version 2A includes functionality to allow the contents of the 3D Wireframe View to be displayed in stereo using a compatible combination of graphics card, monitor and stereo glasses.

The functionality uses OpenGL Quad Buffered Stereo to display stereo images. While a number of manufacturers produce hardware suitable for viewing this type of stereo display, ANSWERS uses and suggests customers use hardware compatible with the NVIDIA[®] 3D Vision[™] Pro system [6].

This is currently a development status feature which can be enabled using a command line argument if users with suitable hardware wish to try the functionality.

5 TESTING

The testing of a tool such as Visual Workshop is important given the accuracy with which the MONK code is expected to be run. One of the major components of a MONK calculation is the geometry, and the need to check that the supplied geometry is correct is vital. The testing of Visual Workshop has thus been considered an important part of the development and release process to ensure that it correctly displays MONK geometry models.

Testing a graphical user interface poses additional difficulties above those involved when testing a Monte-Carlo code such as MONK. These difficulties relate to the fact that a user is required to drive the user interface during the testing process. When you consider that a GUI needs testing not only on different platforms (Windows, Linux etc) but also requires testing with different graphics capabilities and display options that are available on these systems a tester can quickly become ‘snow blind’. Tools are available for recording mouse and keyboard events such that these can be re-played for testing purposes, however, these are not convenient when the application is under development and, for example, the arrangement of the GUI components is changing.

The approach chosen is to test as much of the functionality as possible using built in self testing routines. This means that the user testing can be kept to a minimum and is associated with ‘does the button work?’ rather than ‘does the process invoked by the button work?’ This approach has allowed ANSWERS to increase the coverage and include more obscure options as part of a regular test cycle.

Visual Workshop can be instructed to run in test mode using command line arguments. All of the test inputs and reference files form part of the standard Visual Workshop package so they can be used as installation tests. When each test set is complete the option is provided to produce an HTML test report.

Most of the tests function in the following way:

1. Initialise the test environment.
2. Run the functionality to be tested.
3. Check the expected outcome; this can be by examining contents of program data structures or comparing with reference files.
4. Clean down the test environment ready for the next test.

As a specific example: The ray trace tests compare images generated during the test with reference images. The test images are generated in a 'backing store' making them independent of screen resolution and colour depth. The event log display ray trace tests consist of 220 image comparisons.

Rather than comparing images, which can depend on capabilities of the graphics hardware that is required to run the wire frame, the wire frame tests compare the data that defines the graphical objects.

Figure 9 shows the wire frame tests running.

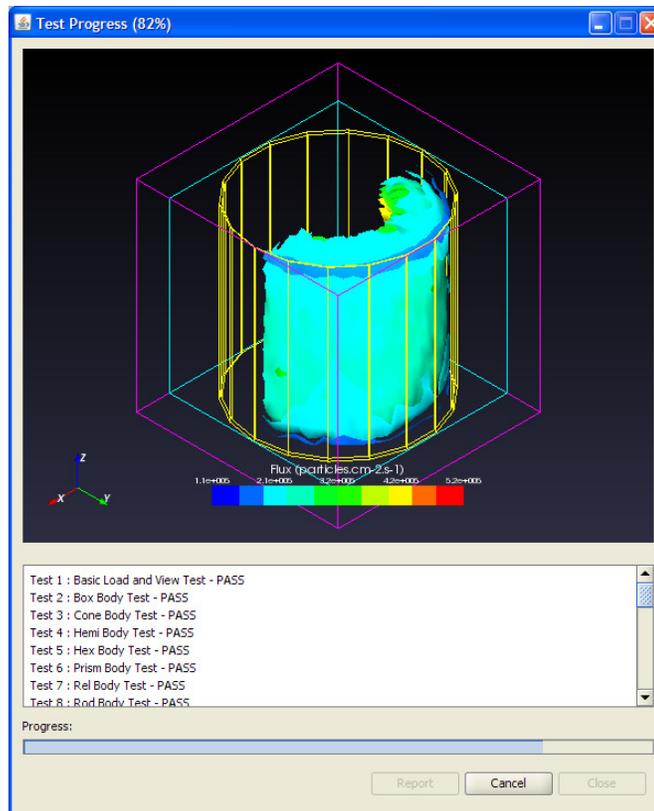


Figure 9: Screenshot of the wire frame self test.

6 CONCLUSIONS

Visual Workshop has undergone significant developments since the release of version 1. It has made large steps towards the long term goal to be an 'Integrated Development Environment' for preparing, modifying, checking, running and analysing results from the ANSWERS codes MONK, WIMS, MCBEND and RANKERN. Feedback from beta test customers has been very positive saying have found the results and event log display features particularly useful.

7 ACKNOWLEDGMENTS

This acknowledges the input of the ANSWERS team who have all contributed to the development of Visual Workshop. Also, the contribution from Sellafield Ltd to the development of Visual Workshop as part of the NCD collaboration with Serco.

8 REFERENCES

1. M. Armishaw and A.J. Cooper, "Current Status and Future Direction of the MONK Software Package". *Proc 8th International Conference on Nuclear Criticality Safety (ICNC 2007), St. Petersburg, Russia (May 2007)* (See http://www.sercoassurance.com/answers/resource/pdfs/icnc2007_monk.pdf.)
2. P. Cowan, E Shuttleworth, A Bird and A.J. Cooper, "The Launch of MCBEND 10". *Proc. 10th International Conference on Radiation Shielding (ICRS-10) and 13th Topical Meeting on Radiation Protection and Shielding (RPS-2004), Funchal, Madeira Island, Portugal (May 2004)* (See http://www.sercoassurance.com/answers/resource/pdfs/icrs2008_mcbend.pdf.)
3. S CHUCAS, I CURL, "Streaming Calculations using the Point-Kernal Code RANKERN". *Proc. 9th International Conference on Radiation Shielding (ICRS-9), Tsukuba, Japan. (October 1999)* (See <http://www.sercoassurance.com/answers/resource/pdfs/paprank.pdf>)
4. P Smith, J Lillington and C Middlemas, "Radiation transport modeling and the ANSWERS code suite", *Nuclear Future, Vol. 7, Issue 2, March/April 2011.*
5. Visualization Toolkit, "An open-source software system for 3D computer graphics (See <http://www.vtk.org>)
6. NVIDIA® 3D Vision™ Pro, "3D Vision Pro for Professionals" (See <http://www.nvidia.com/object/3d-vision-professional-users.html>)