

Building a simulation application for blasting: Case study of software component re-use and customisation in a workflow framework

S.J. Cummins^a, D. Thomas^a, E. Sellers^b, E. Fathi Salmi^b and P.W. Cleary^a

^a CSIRO Data61, Melbourne, Australia

^b CSIRO Minerals, Pullenvale, Australia

Email: Sharen.Cummins@csiro.au

Abstract: In energy efficient blasting, numerical modelling techniques are a valuable tool for miners to identify how the material mixes post blast and therefore to ensure that retrieval of valuable metals in digging operations is as optimal and as safe as possible. As each blast region typically has a different geology and explosive distribution, it is highly desirable to develop an automated and easy to use software tool that can be used by miners to build the pre-blast geometry and run the numerical model to predict post blast geometry. Building such a tool, however, is quite challenging. The numerical model requires construction of a 3D geometry of the blast region, specification of the explosive load in each blast hole and an accurate representation of the surrounding geology; available data however is limited to XYZ positions of each blast hole. Therefore, a software tool to automatically construct a blasting numerical model can greatly assist mining engineers. In order to address this need, a blasting software application called Benchblaster is currently being developed at CSIRO using the Workspace platform. Re-use of existing Workspace infrastructure allows for a rapid construction of an advanced, stand-alone modelling software product with an application-specific UI. In this paper, the development and construction process of Benchblaster is presented as a case study to illustrate how re-use and leverage of a shared underlying IP pool can be used to create a bespoke high quality blasting simulation tool.

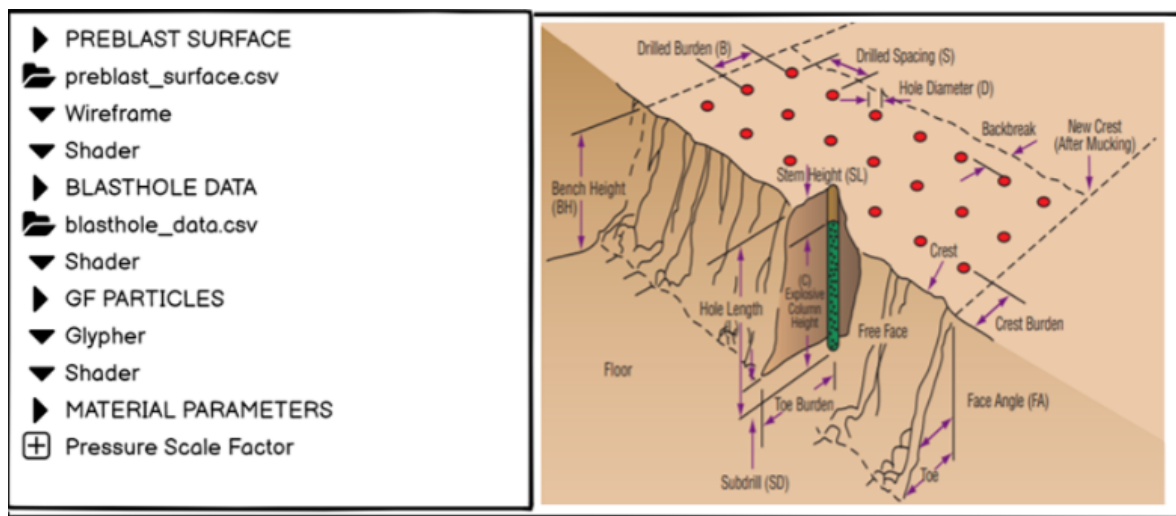


Figure 1. Key components needed to build the Benchblaster numerical blast model

Keywords: *Blasting, Workspace, workflow, software re-use, software customisation, commercialisation*

1. INTRODUCTION

Open-pit mines continue to be the main source of the metals and critical minerals being used to build the technology that enables our modern lives. As older mines become depleted, the remaining ores have a lower percentage of metal within the rock (grade) and are increasingly heterogeneous. The critical outcomes for energy efficient blasting in mining are achieving the desired fragmentation with sufficient movement of materials to allow a good dig rate, whilst minimising mixing and ore dilution. Because the mineralisation tends to be irregularly distributed, and there is often no visual distinction between ore and waste rocks, the first challenge for miners is to identify how the material mixes to ensure that digging operations maximise the metal containing ore sent to the plant for processing and discard waste that would reduce plant throughput and increase costs. One option is placement of transmitters within the blast volume prior to blasting and moving the pre-identified mineral blocks in line with the measurements (Thornton 2015). This approach is approximate, expensive, disruptive and has safety consequences due to the need to locate the sensors within the broken rock pile. Alternatively, numerical modelling techniques can help to provide a better insight into ore movement resulting from the complex processes of detonation, dynamic wave propagation, rock fragmentation and muck pile formation (Sellers et al. 2012). However, the requirements to calculate the physical interactions of millions of particles, including the segregation of the different sized particles have been a limiting factor in these models. The efficiency of the CSIRO Discrete Element Method (DEM) approach (Cleary, 2004, 2009, 2020b) enables the modelling of realistic sized blasts, in three dimensions, with a reasonable range of particle sizes.

The photographs in Figure 2 show a typical blast in progress. The explosives are detonated in sequence to ensure there is always a free surface for the material to move towards. Figure 2a shows how the broken rock moves forward with the largest movement at the mid height of the blast. Once the rock from each detonated hole settles in sequence, the final muck pile forms, as shown in Figure 2b. This example indicates how the adjacent grey and orange rocks have been mixed by the blast processes. Since each blast has a different geology, geometry and distribution of explosives, it is very useful to have an automated approach to building the numerical model that can make use of modern lidar scanning technologies to provide the pre-blast geometry for model setup and the post blast geometry for validation.



Figure 2. Example of a blast (a) in progress, and (b) the final muck pile

To prepare for a blast, holes are drilled in rows. The holes extend below the final ground level to induce upward and forward movement of the rock. The spacing of the holes is designed to ensure there is sufficient energy based on the amount of explosive in each hole per unit volume of rock. In the mine, the required amount of explosive is placed into each hole, along with a detonator that determines the time of firing ranging between 25 ms and 100 ms between holes. The top portion of each hole is filled with rock aggregate stemming to maintain the explosive energy in the hole for as long as possible. Finally, the detonators are connected on surface and the blast is fired. It is clear that this is a complicated 3D system, and it would be expected that all the associated data would be easily available to build a numerical model. However, the data is generally limited to single flat files tabulating each parameter relative to the XYZ position of each hole. This means that the

construction of a numerical model requires significant time for the modeller to create a 3D geometry for discretisation, add the loading for each hole and include the relevant material zones representing the geology. Therefore, automatic construction of the blast model using a software application can make a huge difference to the practical application of DEM models in mining.

This blasting software application, referred to as Benchblaster from here on, is built using the Workspace platform (Workspace, 2014) with a workflow software paradigm. This allows rapid construction of such software by a numerical modeller to a high software quality level providing robustness, ease of extension and the ability to distribute this as a stand-alone application, opening the possibility of its direct commercialisation. The construction of an advanced modelling software product for grinding mill simulation using Workspace has previously been explained in detail (Cleary et al. 2020b). Thomas et al. (2023) describes in more detail the methodologies available for rapid construction of such software from a shared underlying IP pool and how this can be customised to achieve both a bespoke application with a GUI designed specifically for the application but with high levels of software component re-use. Here, we use the construction of this blasting software as a case study to show how this can be achieved with the underlying simulation software and much of the software infrastructure being common. A final advantage of such an approach is that the resulting software application is of sufficient software quality that it can be provided to external customers enabling fairly straightforward commercialisation without the need for sufficient re-investment in the product.

2. WORKSPACE SUMMARY

Workspace (Cleary et. al (2020a), Bolger et. al (2016)) is a powerful, cross-platform scientific workflow framework that enables and software reuse and commercialisation of research IP, particularly modelling and simulation software. It has been used in numerous applications such as hydraulic fracture (Kear et. al (2019)), welding (Thomas et. al (2019)) and diving (Cohen et. al (2017)). Workspace allows for the creation of custom operations that can be used in conjunction with a large set of powerful built-in operations to create scientific workflows. These custom operations can also be used in existing applications that are built with the Workspace architecture, enabling low-cost extension of the application capabilities without major reworking.

3. BENCHBLASTER PRODUCT APPLICATION DESIGN

Figure 3 shows the key components that need to be incorporated into Benchblaster to build the numerical blast model. These include an ability to specify and visualise the pre-blast surface, the location of the blast holes and the initial position of the DEM (Discrete Element Method, see Cleary 2004, 2009 for method details) particles prior to the blast. An important requirement here for the user is to be able to toggle these key elements on/off, to adjust their transparency and to colour them by attributes such as height for the pre-blast surface, detonation time for the blast holes and material properties for the DEM particles (among others). This permits the user to easily check the initial conditions of the numerical model before running the blast simulation. An advantage of building Benchblaster using the Workspace platform is that existing shared underlying infrastructure can be heavily leveraged to build these modelling and visualization tools.

The process to build the numerical model, run it, visualise and post-process the results needs also to be straightforward for users. The Workspace standard GUI workflow based on a primary horizontal row of tabs is chosen (but could be anything else a developer wanted). The Balsamiq UI mockup design tool has been used to quickly prototype the design. Figure 3 shows one of the UI design mockups that Benchblaster was built from. The idea behind the design is that the user traverses the top tabs from left to right, starting with specification of the input files, see ‘Input Files’ tab in Figure 3. These input files are simple CSV files providing XYZ positions of the surface and blast holes and a template DEM particle location file. By clicking on the next ‘BENCH SETUP’ tab, pre-existing Workspace CSV file readers and tools will then be used to create a triangulated surface mesh to represent the blast surface and to create cylindrical glyphs to represent the blast zones. Any necessary co-ordinate transformations from the world frame to the DEM model frame will also be presented. Clicking on the next ‘Particle Creation’ tab will result in a lattice of DEM particles to be built under the blast surface. Existing Workspace solver file writers will then be used to create DEM solver input files. Finally, DEM simulation parameters such as gravity, particle friction and spring stiffness are specified in the ‘Material Parameters’ tab. At this point the simulation will be ready to be run and visualised which can occur by clicking on the final ‘Visualisation’ tab. A ‘Postprocessing’ tab will in future be added to provide important simulation diagnostic information.

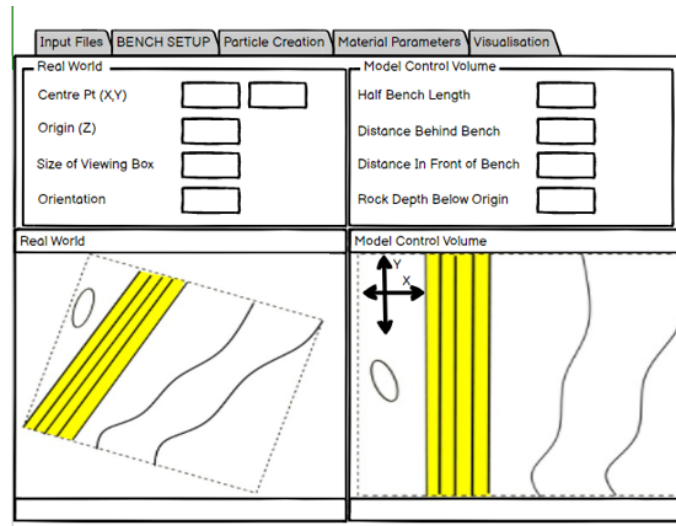


Figure 3. A UI mockup used in the design process of Benchblaster showing the proposed process to build, run and visualise the numerical blast model

4. DEVELOPMENT PROCESS

The Benchblaster development process can be summarized as follows. The Balsamiq UI mockup tool was first used to quickly prototype and design the required UI. Using these UI mockups, Qt Designer was then used to build the front-end UI. The driver Workspace workflow was next created, taking care to re-use existing functionality where possible. New components (such as building the lattice of DEM particles under the blast surface and writing out the blast zone data to a DEM solver input file) were identified and created as C++ code. Once the driver workflow was built, its relevant inputs and outputs were linked to the Qt designer UI elements (as explained in detail in Cleary et al. 2020b). At this point, the Benchblaster product could then be created. The product creation was done by leveraging the existing Workspace application builder which essentially links the Qt designer files, workflows and underlying workspace components to create a stand-alone application that can be run on multiple platforms.

5. THE BUILT PRODUCT – BENCHBLASTER

The stand-alone cross-platform Benchblaster product is described in Figures 4 to 9 below. Once executed, the model build process window is presented with the user required to traverse tabs from left to right as shown in Figure 4. The user first clicks on the ‘Input Blasting Files’ tab and enters the blast surface, blast hole and template DEM particle input files.

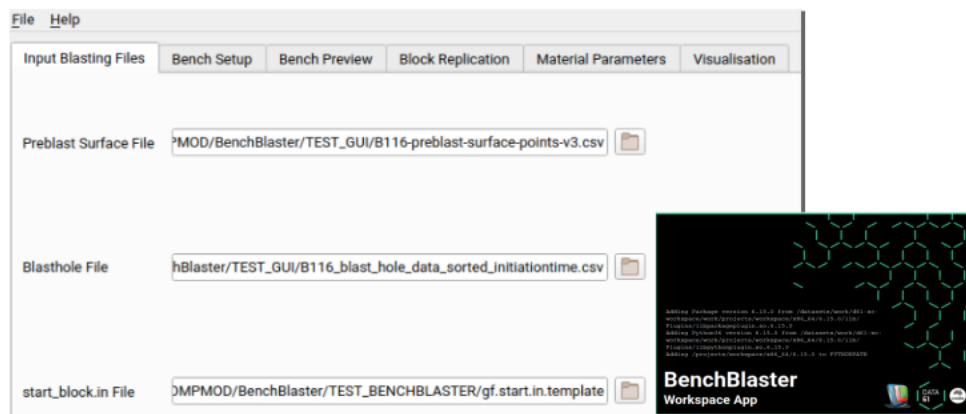


Figure 4. Benchblaster ‘Input Blasting Files’ tab

Referring to Figure 5, the user next clicks on the ‘Bench Setup’ tab to specify the domain (control volume) and origin of the blast simulation and transformations from world frame to model frame. Height contours of the blast surface in the world frame and model frame are then presented. The user next clicks on the ‘Bench

Preview' tab which permits the 3D blast surface and blast holes to be previewed, see Figure 6. The blast surface is a traditional Workspace surface mesh model with triangle elements, coloured by the height of the surface mesh nodes. The blast holes are a standard Workspace particle mesh model with cylindrical glyphs that can be coloured by blast hole attributes such as detonation time or duration. In Figure 6 they are all coloured red.

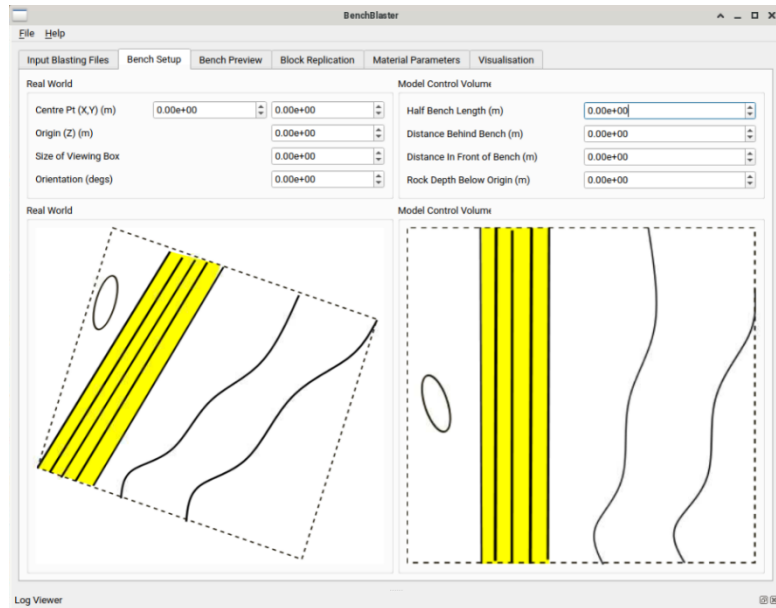


Figure 5. Benchblaster 'Bench Setup' tab

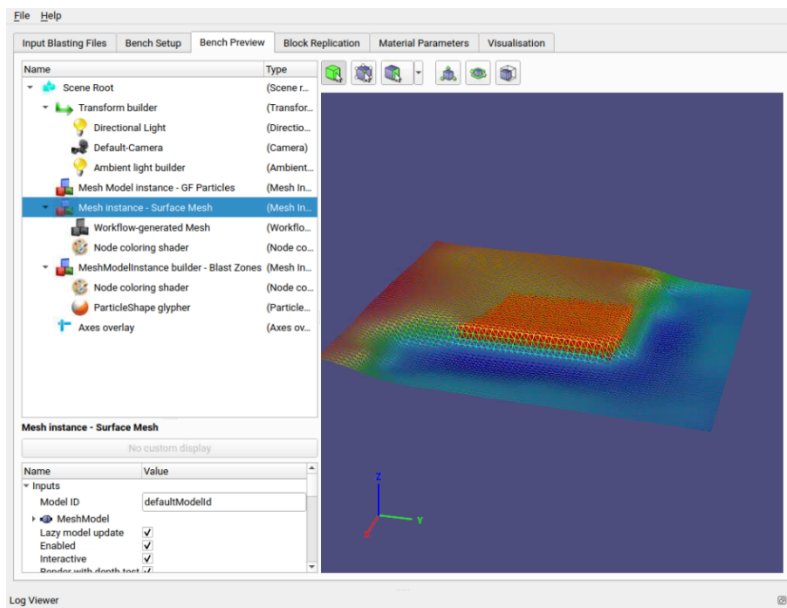


Figure 6. Benchblaster 'Bench Preview' tab

With the control volume, blast surface and blast holes specified, the user must next create the DEM particles. This is done by clicking on the 'Block Replication' tab. In this step, the user takes the template DEM input file (which contains a block of DEM particles) and defines the desired particle size, shape and aspect ratio. Pressing the 'Build Bench' button then builds a lattice of evenly spaced particles by replicating the template block multiple times until the portion of the control volume under the blast surface is evenly filled with particles. The resulting set of DEM particles and blast surface are viewed by clicking on the 'Visualisation' tab, as shown in Figure 7. The visualisation tab consists almost entirely of pre-built re-used components. These are based on the advanced 3D rendering and visualisation which is native to Workspace and a hierarchy of advanced features built as combinations of C++ code modules wrapped in Workspace workflows. The ability to rapidly deploy advanced 3D graphics is critical to most physics-based simulation software.

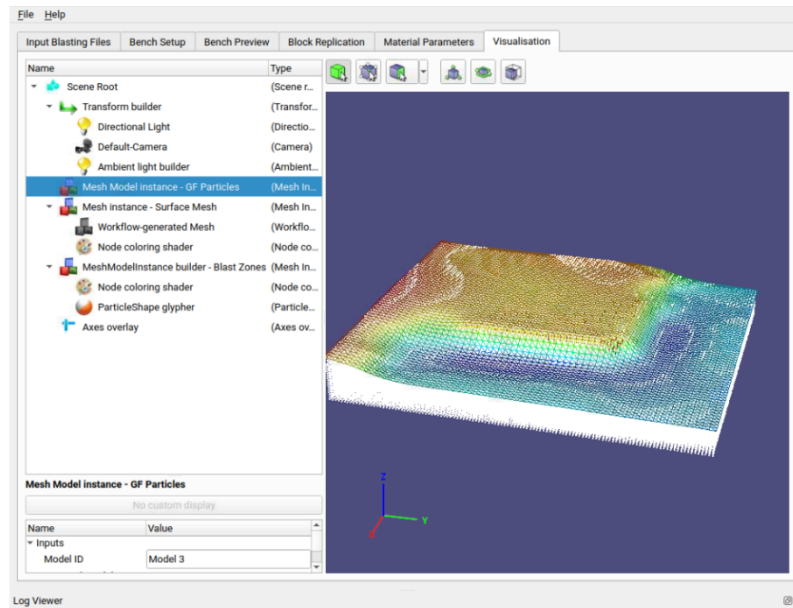


Figure 7. Benchblaster ‘Visualisation’ tab

Figure 8 shows a simulation mid-blast in the visualisation tab. Animations of the DEM particles being ejected from the blast zones are provided with the user able to colour particles by attributes such as speed, force or material. This ability to render the particles is another example of re-use of existing Workspace infrastructure.

The high-level Workspace workflow underpinning the Benchblaster application is shown in Figure 10. This workflow is composed of three important subordinate workflows. The ‘Create Blast Zones’ workflow reads in the CSV blast hole data file, builds a Workspace particle mesh model to represent the blast zones and writes out the blast holes to a DEM solver input file. The ‘Create Blast Surface Mesh and Height Map’ reads in the CSV file surface data file, creates a height map grid object and builds a Workspace surface mesh object from the grid object. The ‘GF Particles Bench Build’ workflow reads in the template DEM particle input file and takes the height map grid object to build a lattice of evenly spaced DEM particles underneath the blast surface. It then writes out the DEM particle data to a DEM solver input file.

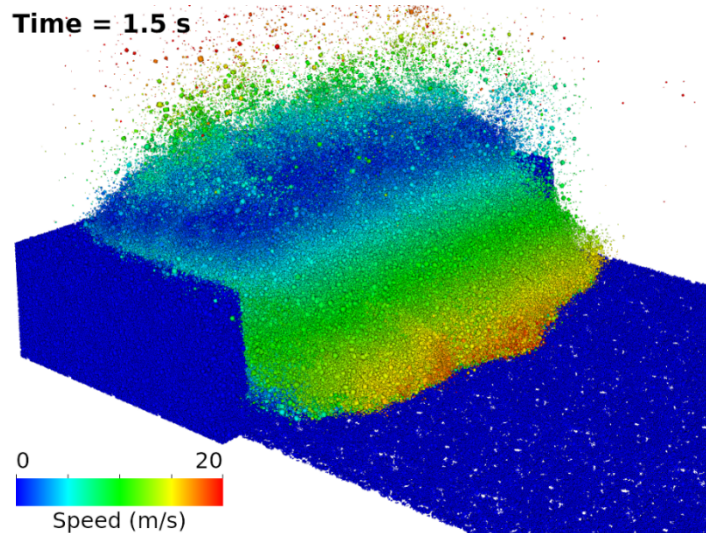


Figure 8. Visualization of simulation mid-blast

It is important to note that a significant proportion of the workflow is composed of pre-existing Workspace functionality (demonstrating that high levels of component re-use are possible) with the new capability confined to the creation of the height map grid object and building the DEM particle lattice (both of which are now part of the Workspace IP pool and therefore able to be re-used in other applications).

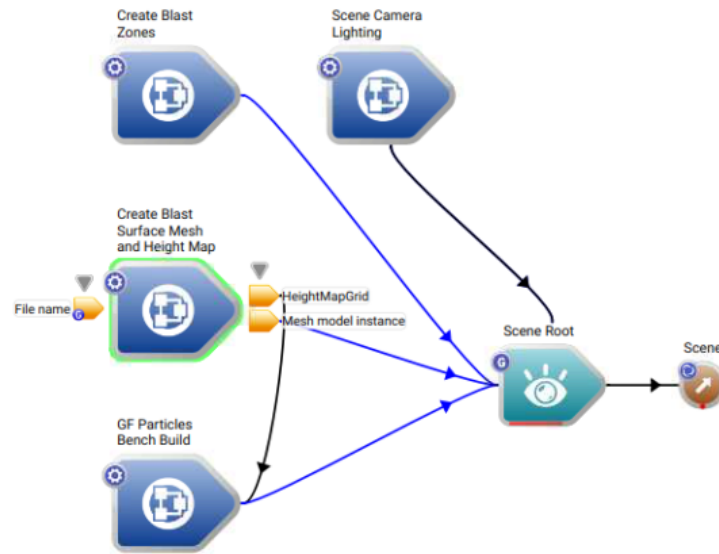


Figure 9. Top level Workspace workflow for Benchblaster

6. CONCLUSIONS

An overview of the development process of Benchblaster has been presented, highlighting how re-use of existing Workspace infrastructure has enabled a nimble creation of an advanced, stand-alone software product.

REFERENCES

- Bolger, M., Cleary, P. W., Cohen, R., Harrison, S.M., Hetheron, L., Rucinski, C., Sankaranarayanan, N., Thomas, D., Watkins, D., and Zhang, Z., 2016, Workspace: a fast and low cost methodology for delivering commercial applications based on Research IP, eResearch Australasia Conference, Melbourne, VIC, Australia, October 2016, Link: <http://hdl.handle.net/102.100.100/89557?index=1>
- Cleary, P.W., 2004. Large scale industrial DEM modelling, *Engineering Computations*, 21, 169-204.
- Cleary, P.W., 2009. Industrial particle flow modelling using DEM, *Engineering Computations*, 26, 698-743.
- Cleary, P.W., Sinnott, M.D., Cummins, S.J., Delaney, G.W., and Morrison, R.D., 2020a. Advanced comminution modelling: part 1 - crushers, *App. Math. Model.*, 88, 238-265.
- Cleary, P. W., Thomas, D., Hetheron, L., Bolger, M., Hilton, J. E., and Watkins, D., 2020b. Workspace: a workflow platform for supporting development and deployment of modelling and simulation", *Mathematics and Computers in Simulation*, 175, 25-61.
- Cohen, R. C. Z., Harrison, S.M., Cleary, P.W. and Bolger, M. Dive Mechanic: Bringing 3D virtual experimentation to elite level diving using the Workspace workflow engine, MODSIM 2017, Hobart, Australia.
- Kear, J., Kasperczyk D., Hetheron L., Quan Nguyen D., Rucinski C., Movassagh A., Salimzadeh S. and Zhang X. Fractura – Common modelling platform for multiple hydraulic fracture simulators, MODSIM 2019, Canberra, Australia
- Sellers, E., Furtney, J., Onederra, I. and Chitombo, G., 2012. Improved understanding of explosive-rock interactions using the hybrid stress blasting model. *Journal of the Southern African Institute of Mining and Metallurgy*, 112, 721-728.
- Thomas, D., Murphy, A.B., Chen, F. F., Xiang J. and Feng, Y. ArcWeld: A case study of the extensibility of software applications built using Workspace architecture, MODSIM 2019, Canberra, Australia
- Thomas, D., Cummins, S. J., and Cleary, P.W., 2023. How to create and support families of deliverable software products from a common IP pool with high levels of software reuse in a workflow, MODSIM 23, Paper, 9-13th July, Darwin, Australia.
- Thornton, A.D., 2015. Monitoring Blast Movement to Improve Ore Control and Blasting Designs. 2nd International Blasting Geology Workshop Sydney, 23rd August 2015.
- Workspace (2014), <https://research.csiro.au/workspace/>