21st International Congress on Modelling and Simulation, Gold Coast, Australia, 29 Nov to 4 Dec 2015
www.mssanz.org.au/modsim2015

Using Workspace to automate workflow processes for modelling and simulation in engineering

P.W. Cleary, D. Thomas, M. Bolger, L. Hetherton, C. Rucinski and D. Watkins

CSIRO Digital Productivity, Locked Bag 10, Clayton South, Victoria.
Email: Paul.Cleary@csiro.au

Abstract: Workspace provides a framework that allows researchers to focus on their science, to develop robust and sustainable software and to accelerate software development timeframes with a faster path to market for their commercialisation. The requirements of scientific application development such as visualisation, distributed computing, testing, integration and provenance reporting are all provided. Researchers and developers can easily develop new capabilities or expose existing libraries through C, C++, Python or JavaScript via inbuilt facilities, or "callout" to other software packages such as R. The Workspace framework makes it straightforward to mix and match existing and new capabilities within an easy to use graphical drag and drop environment, without the burden of having to design and implement the glue to make the components work together. Workflows are built and modified using an intuitive flow chart like graphical interface. They can then be executed directly within the workflow editor environment (typically as part of research or development), batch executed using the command line, or can have customised user interfaces attached (which is typical when creating fully compiled software products for distribution to other users). Workspace is built upon Qt and makes use of Qt Designer for development of application graphical user interfaces which can be connected to the Workspace workflows. Workspace is able to provide significant interoperability between otherwise independently developed and incompatible software components or operations. This is particularly useful in collaborative development between different disciplines or teams. For modelling research, the ability to connect pre- and post-processing components with pre-existing research simulation software and to package this as distributable software products represents an opportunity to commercialise pre-existing research without the need to re-develop the modelling software.

Several Workspace use cases for enhancing, supporting and commercialising engineering modelling and simulation are presented. These include:

1. Simple workflows for the full automation of operations typically performed by a user interacting with one or more pieces of software, such as pre- or post-processing operations.
2. Customised visualisation of large complex data stored, for example in NetCDF files.
3. Integration of third party libraries with inbuilt file reading and visualisation, for example in geometry construction from point clouds from laser scanned data.
4. Simple distributable software products, which can be provided to users, collaborators or customers. These would usually have a customised graphical user interface (GUI) and components for input, computation and then visualisation. An example is GrainScan.
5. Fully featured simulation packages. In this use case, the challenge is to take an existing engineering simulation package (often written in-house for research purposes using text input files for option selection and with analysis usually by collections of third party software) and to package it with a fully featured graphical user interface and embedded visualisation capability to produce a standalone software product that is easy to use. An example for simulation of particle motion in a grinding mill will be presented.
6. Multi-disciplinary research and development, which involves multiple different teams with different skill bases, software development capabilities and background modelling and analysis technologies. An example of the development of a process for automated markerless human motion capture is given which involves image analysis, optimisation and CFD modelling disciplines.

For each usage case, the nature and value of the opportunity is described and examples are discussed that demonstrate both the way that they are built and how they can be used. The benefits and challenges of using Workspace to support more efficient development of modelling and simulation platforms will be discussed.

Keywords: *Workspace, workflow, software platform, Qt*

1. INTRODUCTION

Most of human endeavour can be interpreted in terms of workflows. These are networks of actions and tasks linked together with specific ordering and with information flow through the network. Modelling and simulation are well characterised as workflows with typically a sequence of pre-processing operations to prepare data and instructions, operations for the performance and monitoring of the simulations followed by post-processing of results, analysis and then finally preparation of outputs for inclusion in some form of document, report or paper. A number of these steps usually involve the running of some sort of software package (in-house, public, free or commercial), which is most commonly semi-manual but is sometimes configured to run as a batch process. Pre- and post-processing is often integrated with the solver in commercial packages but for in-house solvers and for advanced use where the supplied capabilities are not adequate then third party solutions can be used. Simulations can be run locally, but also remotely on clusters, sometimes as web services and increasingly there are opportunities to run these on Cloud platforms (Dormer et al. 2011). This commonly requires some form of data transfer which is often manual or semi-manual. So modelling and simulation can typically be characterised as manual or semi-manual processing requiring multiple distributed tasks, possibly by more than one person, in an ad hoc network that is usually not clearly documented and not explicitly thought through. As model complexity increases and simulation volumes increase there is a growing need to automate such workflow processes.

Workspace (Cleary et al. 2014, Workspace, 2014) is a platform independent scientific framework in which workflows: are recorded in XML, allow for dynamic resource discovery, have a single inbuilt scheduler which follows a dependency graph during execution; and provide provenance support. Its interactive workflow editor, built-in visualisation and application packaging capabilities support the full software lifecycle from rapid internal prototyping and development to delivery of polished tools to collaborators and end users. IP management was a major consideration in the design of Workspace and the libraries used by Workspace place no encumbrances on users. Non-commercial products can be freely distributed while maintaining closed source code (if this is needed) while commercial products need only a commercial Workspace license. An important aspect is the capacity of Workspace to integrate low-level functions from multiple sources (including in-house and open source packages). This allows the rapid expansion of functionality with a common ‘look and feel’ for both users and from a software API perspective.

Support for modelling and simulation in engineering, geophysical and biophysical processes is a key use case in the development of Workspace. Commercial engineering applications such as ArcWeld (a welding simulation application (Murphy and Thomas, 2014), AlteTreat (for managing heat treatment of castings) and SheaRunner (which facilitates design of runner systems for thixotropic metal casting) are all built on the Workspace platform (see Cleary et al. 2014). It has also been used as a platform for building applications in bush fire prediction (SPARK see Miller et al. 2015 and Amicus see Sullivan et al. 2013) and flood prediction (Hilton et al. 2015). In this paper, the capabilities of the Workspace platform and its suitability for supporting simulation and modelling will be demonstrated using a simple workflow example, a simple GUI driven application and a full simulation software application for modelling of rock flow in grinding mills. Finally, the inclusion of mobile device and cloud support within the framework will be discussed.

2. WORKSPACE

Workspace is a powerful workflow and application development framework, developed by CSIRO, with two user categories in mind:

- Users who want to create and share scientific workflows in one coherent, easy to use environment where much of the “heavy lifting” has already been developed and proven over a number of years.
- Developers who want to make their software available as standalone products, plugins or components that can be freely or commercially distributed or mixed with capabilities from collaborators.

Workspace provides a framework that allows researchers to focus on their science, develop robust and sustainable software and still meet their software implementation timeframe. The requirements of scientific application development such as visualisation, distribution, testing, integration and provenance reporting are all provided. Researchers can easily develop new capabilities or expose existing libraries through C, C++, Python or JavaScript via inbuilt facilities - or “callout” to other software packages such as R. Thus, the Workspace framework makes it very easy to mix and match existing and new capabilities within an easy to use graphical drag and drop environment, but without the burden of having to design and implement the glue to make all the components work together. The plugin-based framework addresses the needs of both user groups in four focus areas.

2.1. Analyse

Workspace provides a graphical tool through which users can connect up various operations into a coherent workflow, execute it and interact with it while it is running. These workflows can carry out whatever set of tasks the user wants and can be of arbitrary complexity. A broad range of pre-built operations are provided, particularly for visualisation and handling of 2D and 3D data, as well as fundamental building blocks such as flow control (equivalent to if-then-else, loops, etc.), string handling, web services, database support and so on. Researchers can easily develop new capabilities (via built-in code wizards) or expose existing libraries through C, C++, Python or JavaScript via inbuilt facilities - or "callout" to other software packages such as R. Thus, the Workspace framework makes it very easy to mix and match existing and new capabilities within an easy to use drag and drop environment, but without the burden of having to design and implement the glue to make all these things work together. This plugin support allows Workspace to bring together workflows that utilise many existing frameworks, open source libraries and languages such as OpenCV, NetCDF, GDAL, PCL, C++, Fortran, Python, R and JavaScript. This provides powerful abilities for users to create custom workflows for the generation of simulation data and to then analyse and interpret it.

2.2. Collaborate

The modular nature of the Workspace framework makes it an ideal vehicle for facilitating collaboration. Users can share and reuse workflows, they can make their own software capabilities available to others via plug-ins and they can mix and match capabilities from any number of collaborators seamlessly within a single environment. Custom user interfaces can also be created to make shared workflows and plug-ins even easier to use. Special attention has been given to assisting developers wanting to write Workspace plug-ins. The technical requirements on plug-in developers are low, with the target minimum knowledge being that of a self-taught C or C++ programmer. In addition, Workspace plug-ins are inherently cross platform yet still allow developers to utilise their preferred platform and specific development environment (such as Visual Studio under Windows or XCode under OSX).

2.3. Commercialise

A key part of the Workspace design is the set of features it provides for assisting the development of commercialisable software. It supports deployment in a variety of forms, including standalone applications, as part of 3rd party plug-ins, as libraries or toolkits and even as command line tools. It provides easy creation of native release packages on all major desktop platforms as well as unit testing and continuous integration support to ensure software quality and robustness. The framework has flexible licensing that supports both open and closed source plug-ins, allowing freely available and commercially sensitive code to be used together without violating licensing terms.

2.4. Everywhere

Workspace aims to be usable in many different scientific domains. It is currently used in CSIRO projects and products dealing with manufacturing, image processing, disaster management, human motion and agriculture. Workspace aims to allow users to transparently exploit any compute resources at their disposal. This includes parallel and remote execution of workflows as well as support for all major desktop platforms (Windows, Linux, Mac). The emergence of mobile devices supporting increasingly powerful applications and cloud deployment are re-shaping many traditional areas of computer usage. Modelling and simulation usage is yet to benefit from these developments.

3. SIMPLE WORKFLOWS FOR AUTOMATION

A relatively simple example of a simulation workflow is one that reads data from one or more files, performs some processing of that data to convert it into a form suitable for a simulation solver, optionally writing this to file and then calling the solver to perform the required simulation. Once automated as a workflow, using remote compute resources is a simple matter of minor changes to the controls of the workflow which enable data transfer, connection, authentication, resource allocation and management and the return data transfer to a desktop for analysis. This process of constructing the set of simulations and distributing them to a cluster and returning and integrating the results can be performed with a simple Workspace workflow such as the one shown in Figure 1. The loop controls the selection of the required simulation attribute(s) and their setting (which could involve solver input files being generated by the workflow for each case). The counted loop operation controls the creation of the complete parameter set of solver input variables for each variation of a specific input variable. The five operations on the bottom row comprise the body of the loop and are:

- "Map counter value to parameter value" - takes the counter value and translates this to a corresponding solver input parameter value. In this workflow this is simply hard-coded in this script operation, but could easily be replaced with some operation that reads this map from a file, from a GUI element, etc.
- "Compose sub-group parameters" and "Compose Solver parameters" - take the parameter and combine it with other default parameters to create a full set of input parameters required to specify the solver operation.
- "Create solver input files" - takes the full set of solver parameters as input and writes them as input file(s) as required by the solver. Note that the "Solver files directory" is an input into this operation.
- "Run solver" - launches the solver executable. In this example, it is simply launched via a call to the operating system, but a more complex set of operations or a bespoke Workspace operation may be used.

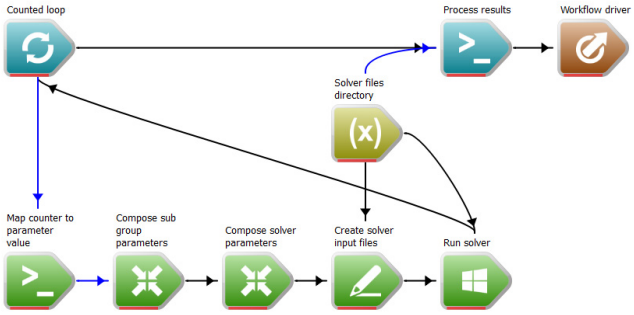

Figure 1. Simple workflow for remote execution of a series of simulations on a cluster.

This set of five operations is run once for each iteration of the "Counted loop" operation. This loop can be run locally, or via a simple setup, can be run in parallel on remote systems. Once these iterations have been completed, the "Process results" operation is executed. In this example, a script operation parses the results files in the location specified by the "Solver files directory" variable. The final operation is labelled "Workflow driver". Workspace uses such end points to determine which operations need updating and in what order to perform this. For example, it ensures that the "Map counter to parameter value" is executed before "Compose sub group parameters" since the latter depends upon the former.

4. CUSTOMISED VISUALISATION OF LARGE DATA – SIMPLE DISTRIBUTABLE APP

The workflow shown is on the left of Figure 2 (the yellow panel) and uses operations from the Workspace Geospatial plug-in to first open a large geospatial dataset (in a GDAL-compatible format), then "slice" it to select a user-defined sub-region. In this example, the dataset is approximately 1 GB of species diversity data covering the entirety of Australia. The first operation, which reads the GDAL dataset, is expanded in the figure so that its inputs and outputs are visible. After the sub-region data is obtained by the second operation, the workflow splits into two branches: the first (top) branch creates a map by extracting data from the sub-dataset and mapping each cell to a colour using a user-defined colour spectrum, while the second (bottom) branch writes the sub-region to disk in a user-specified format and location. The "UpdateBarrier" operation at the end of this branch causes it to be executed only in response to a trigger, such as an up-stream dependency, or in this case, a user button click. The user interface (UI) is shown on the right of Figure 2. Each graphical widget on the UI is directly attached to a specific input, output or operation of the workflow. For example, the "File name" widget (indicated by the upper orange shaded region in the UI) is directly connected to the "File name" input of the "Read GDAL Dataset" operation (as indicated by the orange arrow). After executing the GDAL read some of the outputs of the operation are then displayed in the upper violet shaded output widget (with the data passing from the workflow operation to the UI as shown by the violet arrow). In a similar way, the spatial extent of the sub-region to be shown is specified by the lower orange shaded widget in the UI. This information is passed from the UI widget and used as an input to the second workflow operation that controls the set-up of the sub-region. Some output information from this operation is returned to the UI and shown in the lower violet shaded widget allowing the user to see necessary diagnostic and interpretation information. The large map widget on the right of the custom UI shows the coloured map resulting from the workflow processing of the data. This widget is attached to the "MapImage" output of the "Display Map" operation in the workflow (with the data flow shown by the mid-blue arrow). The user connects the UI layer to the workflow layer using the simple drag-and-drop functionality of QtDesigner. The use of a UI layer allows an end user to access the functionality of the workflow in a user-friendly way without needing to understand all of the detail that the workflow provides.

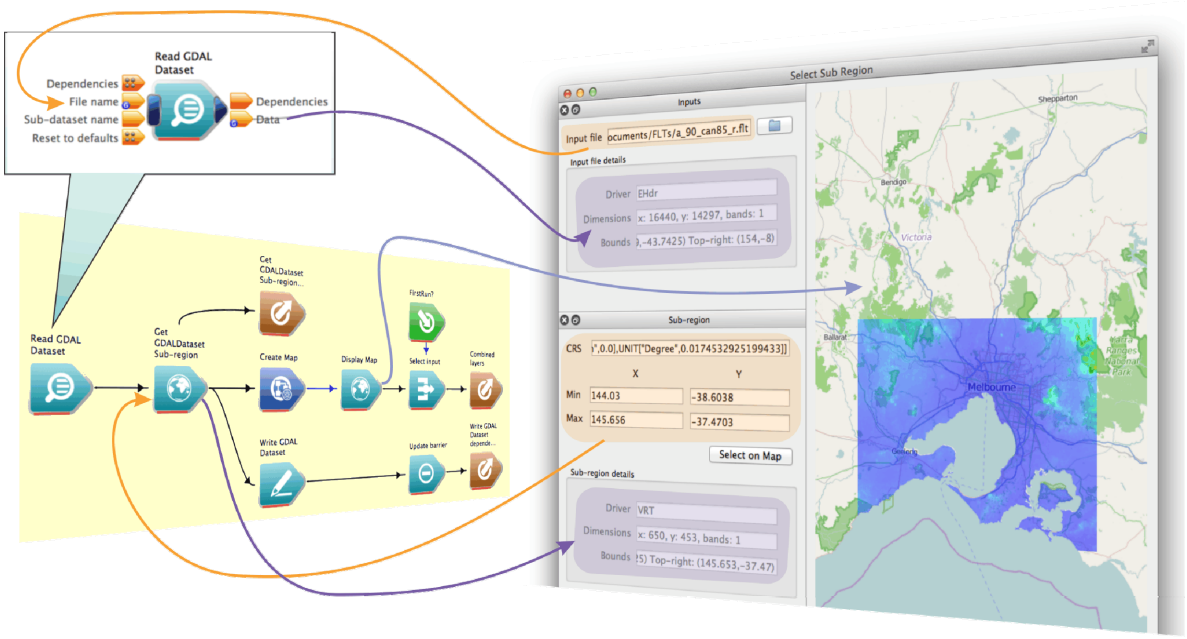

Figure 2. Simple product construction with (left) an underlying Workspace workflow, and (right) a simply connected UI that provides user access to specific input controls of selected workflow inputs.

5. FULL FEATURED SIMULATION PACKAGE

Workspace is also able to create fully-fledged applications based upon available solver IP (often from in-house research). One such commercial application is GF-Mill which is used to simulate the operation of grinding mills (see Cleary, 2004 for details of the underlying Discrete Element Method (DEM) solver and its application to mills). This software enables significant financial and time savings to be made in the design and optimisation of mill liners. On its own, the DEM solver code is typical of much scientific software in that it is non-trivial in its use due to the inherently complex nature of the problem being tackled. This necessitates many input parameters which are specified in input files – in this case they cover aspects such as material specifications, mill models, physical parameters etc. To use the software in its native state requires a user to edit input files to set the parameters and to be expert in understanding of many options. The pre- and post-processing operations and visualisation are also performed by separate software applications with manual processing in between. So the workflow using the base solver is semi-manual, time consuming and requires a high level of expertise. Using Workspace allows the streamlining and automation of this workflow making it much easier to use with an intuitive GUI with a commensurate reduction in usage errors. It also allows non-experts easy access to the capabilities of underlying software without having to master all of its complexity.

The GF-Mill application is constructed from the following components:

- Workspace workflows, often built as a nested hierarchy
- Qt "widgets" – GUI elements
- C++ framework code
- FORTRAN based DEM solver

The Workspace workflows were easily built using the Workspace GUI environment, the Qt widgets being constructed using Qt Designer, and the C++ framework code partially created by in-built Workspace GUI wizards. Due to the inherent nature of the workflows and widgets, these components easily lend themselves to reuse in similar application packages. It is also a simple process to customise these components on a per-application and per-customer basis. In addition to facilitating a simplification of the simulation set-up, Workspace also provides various components for visualisation of results. For example, in GF-Mill, Workspace's 3D scene components are used to visualise the mill model and particles inside the mill. Development leverage is created due to the ability to easily customise and embed such applications within other more complex applications. Post-processing utilities such as for solver performance, energy input and utilisation are examples of Workspace applications that can be used as stand-alone applications or embedded into other applications (enabling high levels of software re-use and lower overall development cost). A brief summary of the GF-Mill application (as shown in Figure 3) is as follows:

- The GUI presents a progression of tabs that follow the linear simulation process, input parameter set-up, simulation control and monitoring, results visualisation, results analysis,
- The user either accepts the defaults for various parameters, or changes them as they see fit,
- Parameters are grouped into easily identifiable areas of the application windows,
- The 3D scene components allow a user to visualise the initial model set-up,
- The user launches the solver via a button click. The simulation runs in the background allowing the user to quit the main GUI application leaving the simulation to continue, or they may monitor the simulation status via the GUI,
- Results can be monitored in real-time as they are produced by the solver code,
- 3D visualisation of the particles and their motion is provided,
- Post-processing utilities are used to analyse results,

GF-Mill is built using an easily configurable array of pluggable components (each built on Workspace) that can be used either stand-alone or as part of composite applications. The ability to add a GUI layer achieves more than just providing a UI with which the user can interact - it also collects the supporting operations such as geometry preparation, parameter setting, visualisation, and post-processing which would otherwise be performed with different pieces of software from a range of sources organised in ad-hoc workflows.

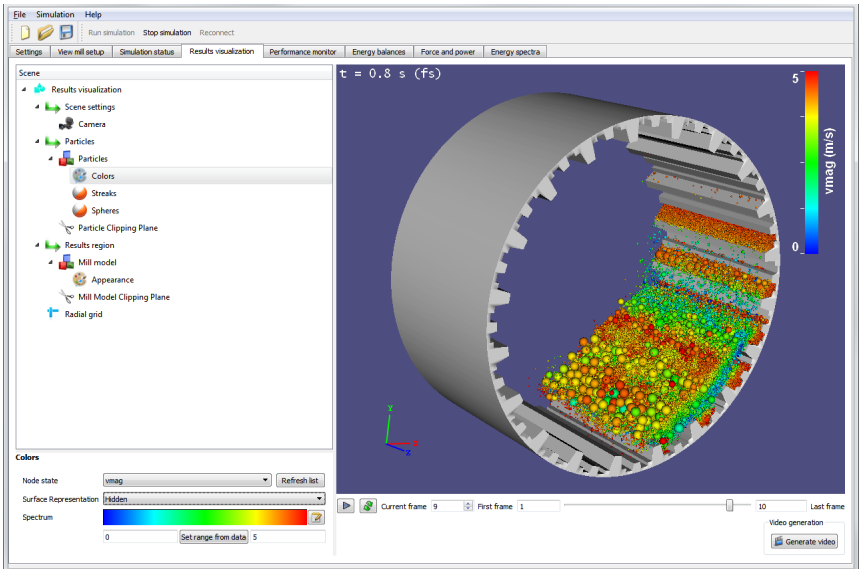

Figure 3. Full simulation application with visualisation panel shown.

6. MOBILE DEVICES, CLOUD AND WEB SERVICES

Simulation and modelling are often performed on desktop and cluster computers. Workspace automatically provides the ability to deploy to both types of platforms. It also features a distributed execution capability allowing it to distribute work to local process threads, local machine remote processes and remote machines via TCP and cluster job systems (PBS & SLURM). This provides easy distribution of simulation jobs to server farms and clusters while its pluggable architecture making it easily extended to support other job management systems and compute resources. Cloud is now a ubiquitous platform for many commercial applications and for providing web services, but there has been limited deployment of simulation capability onto Cloud servers. Deployment options for Workspace on Cloud are currently being developed. Workspace is also being extended to support mobile and web based deployment. All mobile platforms supported by Qt will be supported (subject to licensing requirements) including Android, iOS and Windows Phone. These QML applications can also be deployed to desktop platforms allowing for a single application and user interface codebase to be built and deployed to all platforms. Deployment of modelling capability as web services has also not been broadly adopted. The Workspace mill application is an early modelling application developed as the web service *web*GF-mill (Morton and Dunstall, 2004). A flexible web browser based deployment solution for Workspace is now also being developed to demonstrate this.

7. SUPPORTING COMMERCIALISATION

The strong configurability inherent in the Workspace design of applications means that interfaces can be specific to each application and therefore have a low threshold for user knowledge whilst maintaining

relatively low development cost enabled by high levels of workflow, widget and operation reuse. This allows the UI to be inexpensively tailored to specific user needs, and not forced into a broad application generality that is usually required by commercial products which must target a sufficiently large market to make the development cost viable. These commercial pressures mean that traditional modelling software is a compromise with usability sacrificed in order to support larger application problem ranges. In adopting a Workspace approach many of the underlying software elements are common to a family of related custom applications with differences primarily occurring in the visible UI layer. The development and maintenance cost of the application family is therefore not strongly sensitive to the number of customised variants. This changes the balance between usability and generality requirements favouring high levels of customisation and facilitating UI simplification. This is particularly important for research simulation and modelling software which inherently address smaller or niche markets that are rarely economic for commercialisation using traditional software development models. A critical consequence of UI simplification is the resulting reduction in the knowledge and skill level required to use such modelling packages. This is a major paradigm change for modelling software design which has significant commercial implications.

8. CONCLUSIONS

The impact of research can be greatly enhanced when it is easy to share and reproduce. This is applicable in both the research and commercial domains. In the research domain, many publishers are now insisting that the software and data supporting scientific discovery be made available for other researchers so they can verify results. In the commercial domain, partners want a simple, robust means to exploit discoveries and ship them to market. Using an advanced software platform technology, like Workspace, allows scientists to easily exploit opportunities in both domains. Together, the four key focus areas highlight the benefits of Workspace in a research environment. It allows researchers to focus on their research whilst also giving business units a path for commercialising the IP generated. By facilitating innovation and streamlining difficult or time consuming tasks, Workspace can get you to where you and your organisation want to go, faster and more effectively. This has been demonstrated for a simple modelling workflow, a simple GUI application based on a workflow and a full commercial modelling application built on Workspace.

ACKNOWLEDGEMENTS

The authors would like to thank Bronwyn Harch, John Taylor (Director of the CSIRO Computational and Simulation Sciences Future Science Platform) and Angus Vickery (CSIRO eResearch) for their generous support in the development of Workspace.

REFERENCES

Cleary, P. W., (2004), Large scale industrial DEM modelling, Engineering Computations, 21, 169-204.

Cleary, P., Bolger, B., Hetherton, L., Rucinski, C., Thomas, D., Watkins, D. (2014). Workspace: A Platform for Delivering Scientific Applications", Proc. eResearch 2014, Melbourne, Australia, 27-31 October.

Dormer, A., Cleary, P. W., and Dunstall, S. (2011). Evolution in Engineering Services and Logistics - Decision Support from the Cloud, Journal of System and Management Sciences, 1, 1-10.

Hilton, J., Miller, C., Bolger, M., Hetherton, L., and Prakash, M., (2015). An Integrated Workflow Architecture for Natural Hazards, Analytics and Decision Support, in: *Environmental Software Systems. Infrastructures, Services and Applications, IFIP Advances in Information and Comm. Tech.*, 448, 333-342.

Miller, C., Hilton J., Sullivan A. and Prakash M., (2015). SPARK–A Bushfire Spread Prediction Tool, R. Denzer et al. (Eds.): ISESS 2015, IFIP AICT 448, pp. 262–271.

Morton, D., and Dunstall, S., (2004), Using the Web to increase the availability of DEM-based mill modelling, Minerals Engineering, 17, 1199–1207.

Workspace, (2014), DOI: http://dx.doi.org/10.4225/08/54D03170101B7

Murphy, T., Thomas, D., (2014), A user-friendly predictive model of arc welding of aluminium, Proc. 4th IIW Welding Research & Collaboration Colloquium, Wollongong, Australia, 5-6 November 2014, pp. 47.

Sullivan, A., Gould, J., Cruz, M., Rucinski, C., and Prakash, M., (2013). Amicus: A national fire behaviour knowledge base for enhanced information management and better decision making, 20th International Congress on Modelling and Simulation, Adelaide, Australia, 1–6 December 2013.