23rd International Congress on Modelling and Simulation, Canberra, ACT, Australia, 1 to 6 December 2019
mssanz.org.au/modsim2019

ArcWeld: A case study of the extensibility of software applications built using Workspace architecture

David G. Thomas[a], **Anthony B. Murphy**[b], **Fiona F. Chen**[c], **Junting Xiang**[c] **and Yuqing Feng**[d]

[a] CSIRO Data61, Box 312, Clayton South VIC 3169, Australia,
[b]CSIRO Manufacturing, PO Box 218, Lindfield NSW 2070, Australia,
[c]CSIRO Manufacturing, Private Bag 10, Clayton South VIC 3169, Australia
[d]CSIRO Mineral Processing, Private Bag 10, Clayton South VIC 3169, Australia
Email: david.thomas@data61.csiro.au

Abstract: The ArcWeld software is a "factory-floor" application, designed to provide simulation of welding capability to users with little or no knowledge of the underlying mathematics or physical processes involved. As such it is an excellent demonstration of how stranded IP, which is only usable by an expert, can be converted into a user-friendly software application, presenting complex inputs and results in a simple-to-use and easy-to-understand form. ArcWeld is built on the powerful Workspace workflow framework, which presents an easy pathway for researchers to turn their scientific software and manual processes into an end product usable by a wide range of customers.

This paper describes the challenges facing the developer when producing, maintaining, and enhancing software such as ArcWeld, and how the Workspace ecosystem can mitigate many of the obstacles facing a researcher whose aim is to make their solver's modelling and simulation capability more widely available. This is illustrated using a case study, in which the changes and extensions required by a new commercial customer are described. The changes included a wider range of process parameters and the requirement for in-built graphics. This required additional input and output files as well as many changes to the graphical user interface.

A detailed description of how Workspace was used to facilitate these changes is provided. The product was previously built on Workspace; this had ongoing benefits in terms of extension and customisation. It is demonstrated that Workspace provides the flexibility and ease-of-use to allow the software to be modified without the need for any major rewriting or reworking. This streamlined the process and reduced the likelihood of errors, ensuring that the modification was fast, inexpensive and painless.

Keywords: *Arc welding, graphical user interface, workflow, Workspace, computational fluid dynamics*

1. INTRODUCTION

The situation that led to the development of ArcWeld is one that may be familiar to the reader – a researcher has source code for simulation, often written in a language such as Fortran, that models a complex physical process, and that is subject to continuous development and iterative revisions to improve the model and the results it produces. The physical process that is modelled is of interest to a range of others, potentially within the commercial sector, but the complexity of the simulation process means that any user of the software is required to be an expert in setting up, running, and interpreting the results. This simulation process commonly involves the following steps, or a variation of them:

- Set up the simulation, which requires: specification of all input parameters, usually via text files, and usually manually entered or modified; determination of location of these input files; determination of the desired location of any output files.
- Run the simulation: it is very common for this to involve running a compiled Fortran binary executable on the command line, ensuring it is run in the correct directory location with the correct input files.
- Monitor the simulation: this requires knowledge of the files produced during the simulation and how to read them. It may involve the use of scripts to parse files.
- Process the results: this may only be done at the end of the process, or it may be possible to see interim results, depending on the nature of the simulation and the implementation of the model in code. The results may be in the form of text files and/or graphics files. Viewing the latter generally requires additional software. Again, scripts may be used, and some manual editing may be involved.

This procedure may also be coupled with multiple runs with different input parameters, and possibly having to cope with incremental changes to the solver code over time.

All the above are highly susceptible to user error, and are substantial blockers that prevent uptake of potentially very useful and valuable intellectual property (IP) (Cleary et al., 2017b). The need to provide detailed and precise process instructions to run the simulation, which themselves can be subject to error and frequent update, also does not lend itself to a viable commercial product.

One, perhaps slightly naïve, approach to address these problems would be to write a piece of software from scratch that would encompass all the above steps. While this is sometimes feasible, the downsides are considerable, and include:

- The requirement to learn a good deal about the full software design process;
- The need to maintain a large software base;
- The choice either to build re-usable software (a non-trivial task) or to accept that the code will only be used for the one case;
- The fact that extensions to the software (for example, new functionality, or changes needed for a new customer) can involve significant changes to a code base.

All of these incur large costs; financial, team resources, and time.

The authors have taken a different approach, and have used the CSIRO Workspace workflow platform (Cleary et al., 2017a) when building the ArcWeld software application. Workspace is a scientific workflow system that can be applied to a range of software development roles, for example, prototyping software, batch processing, process monitoring and application construction. The key component of Workspace is the *workflow*, consisting of *operations* (representing discrete, functional components of a workflow) and *connections* between these operations, representing the passing of data between operations. Workflows can be nested, reference other external workflows, and be linked to graphical user interface (GUI) elements. In use cases such as this, Workspace enables a user to compose one or more workflows, each of which represent a component (or components) of functional flow corresponding to the processes involved in running a simulation – from composing input data sets, through monitoring the simulation run, to visualising and analysing the resulting data sets. Depending on the complexity of the simulation processes, the application can use many workflows, or maybe just the one workflow for very simple applications. A detailed case study describing the writing and capabilities of this application was presented by Murphy and Thomas (2017, 2019), which deal directly with building a new commercial-grade application based on legacy solver capabilities.

Once used and successful, such software typically generates ideas and demand for extension and modification with new capabilities needing to be added or exposed. Potential customers may emerge who require different customisations - software built in Workspace is, by its very nature, easily extensible, customisable (at workflow and application levels), and so is ideally suited to supporting continuing development of the software.

This paper shows how having an application in a workflow form implemented in Workspace allowed for easy, quick and cost-effective modification. It demonstrates how the existing code base (the term is used to include the underlying workflows as well as the C++ code) was:

- Re-used – for example, many of the solver input files did not need to be changed for the new version, and so the existing functional components remained the same;
- Extended – for example, in-application graphing capabilities were added;
- Further customised – for example, additional metals were made available for selection.

These changes provide powerful new solver capabilities in an easily accessible, user-friendly interface.

In section 2 we provide an outline of ArcWeld version 1 and Workspace, and in section 3 we explain the nature of the extensions and the motivations behind these. We then explain the main methods used in the extension, detailing how Workspace made this process considerably easier and cheaper, for example by allowing re-use of existing components and facilitating the addition of new components, than it would have been had the software been built from scratch. Section 4 provides some brief conclusions.

2. ORIGINAL IMPLEMENTATION OF ARCWELD USING WORKSPACE

ArcWeld is the name of the commercial-grade software application used to simulate MIG (metal inert gas) welding. MIG welding, also known as GMAW (gas metal arc welding), uses the intense heat flux produced by an arc plasma to partially melt two pieces of metal, which are joined when the molten region (the weld pool) solidifies. A schematic of the process is shown in Figure 1. The arc plasma is struck in a shielding gas between a wire electrode and the workpiece – the metal being joined. The wire electrode melts, forming droplets that pass through the arc into the weld pool; accordingly, the wire is continuously fed (Norrish, 1992). Performing any weld requires the appropriate selection of a wide range of parameters, including the shielding gas (typically argon for welding of aluminium, argon with oxygen or carbon dioxide for steel), workpiece geometry (bead-on-plate, lap, T-joint etc.), orientation of the wire with respect to the workpiece, arc current and welding speed, and choice of alloys for wire and workpiece, etc.

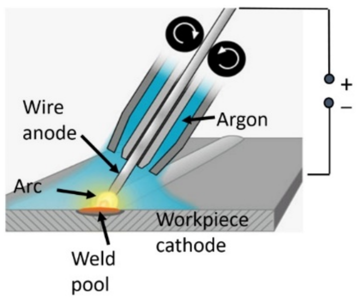


Figure 1. Schematic of the MIG welding process

The core solver is a Fortran code that solves a set of coupled partial-differential equations that describe the fluid dynamics and electromagnetics of the welding process, using the finite-volume method presented by (Patankar, 1980). Details of the equations solved, and approaches used in the model can be found in Murphy (2011, 2013a, b) and Murphy et al. (2017). When the computational model was originally developed, a binary executable (compiled Fortran code) was run from a command line on a desktop computer running Linux or Windows. Three text files were used to define the input parameters. The first defined the welding parameters (e.g., arc current, welding speed, feed rate, droplet frequency, etc.) and other settings (e.g., whether to use a pre-defined or pre-generated solution set as the initial conditions, which can speed convergence, or to start from scratch; whether to include or exclude mixing of wire and workpiece alloys in the calculations). A second file defined the geometric configuration (workpiece geometry and thickness, electrode orientation, and meshing parameters). The third file defined the type or composition of metal alloys to be used for the wire and workpieces. Editing of the text files is easily subject to user error (e.g., pointing to the wrong initial condition file, missing out a decimal point, or provision of a value in incorrect units). As simulations were running, textual output to the console, along with several output files, needed to be examined manually if the user was to monitor the state of the simulation – as such, even experienced users could easily miss pertinent output. A flow diagram that gives details of how the solver was run before integration with Workspace was given by (Murphy and Thomas, 2017, 2019).

To produce the standalone ArcWeld application, the authors used the Workspace platform, taking the approach of automating all error-prone manual tasks, as well as streamlining the other elements of the process (for example, removing the need for the user to do any file management, other than choosing a single project location). This is discussed in detail in (Murphy and Thomas, 2017, 2019). In addition, monitoring of the simulations was automated, and the results were made easily available from the application interface.

The most substantial part of the main workflow is shown in the Workspace editor in Figure 2. The left-most operations, all in dark blue, are nested workflows (i.e., they contain other workflow elements), and each one generates the contents of a single solver input file. Each is labelled to match the file generated in order to aid maintainability. These nested workflows are connected to various GUI elements (e.g., spin boxes for numerical values, checkboxes for Boolean values). The other operations collect and output the data from the workflow. These outputs are monitored by the application, and when the user is ready to run the simulation, they can be validated and then used to create the input files needed to run the simulation.

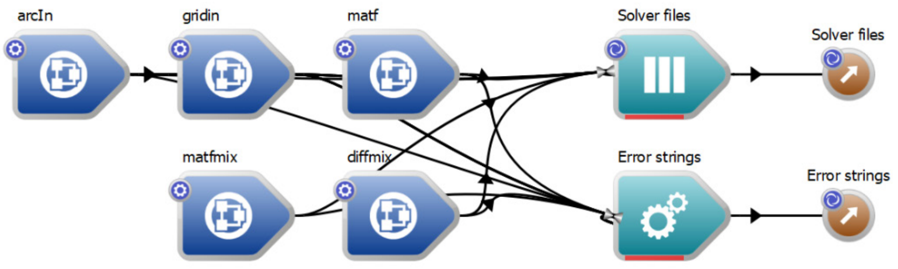


Figure 2. Main workflow for ArcWeld version 1

Version 1 of the ArcWeld was built using Workspace version 3.2 in a project jointly funded by the AutoCRC and GM Holden. It was delivered to the customer in 2014. Their primary use-case for the software was to assist welding engineers and technicians choose optimum parameters for the welding of aluminium alloys in butt and lap joint weld geometries.

The customer requested that instead of built-in graphics, the solver should write files in a form readable by the commercial TecPlot software. The GUI provided hyperlinks and scripts so that three standard figures could be opened; an *x*-*y* plot showing the weld cross-section, a contour plot showing the reinforcement height and the extent of the weld pool, and a contour plot showing the temperature distribution in a vertical slice through the wire electrode, arc and weld pool (Murphy and Thomas, 2017, 2019).

3. ARCWELD VERSION 2

In a project jointly funded by the Rail Manufacturing Cooperative Research Centre (RMCRC), CRRC Qishuyan Locomotive & Rolling Stock Technology Research Institute Co., Ltd. (CRRC QSYRI) requested four main extensions and changes to the ArcWeld software:

- Inclusion of additional metals, including Al-Mg-Zn alloys, magnesium alloys, mild steels and carbon steels;
- Inclusion of shielding gases composed of Ar-O_2 and Ar-CO_2 mixtures, which are widely used in welding of steels;
- Inclusion of two additional welding geometries, T-joint and corner-joint (or fillet) weld geometries
- Replacement of the TecPlot graphics by built-in graphics

Currently, the first and fourth extension items are being tested by the customer following an intermediate release. The other extensions are in the beta testing phase, with final delivery of version 2.0 of the software expected in the fourth quarter of 2019.

The initial implementation of ArcWeld will be in the CRRC QSYRI's Welding and NDT Training Center. It will be used in the training of welders, for example, by allowing them to visualise the weld geometry for a range of different welding parameters (arc current, welding speed, electrode angle, etc.). Use of ArcWeld simulation will further allow the optimisation of welding parameters to produce the highest quality welds for different welding situations (e.g. weld geometries, workpiece thicknesses and workpiece alloys). Finally, QSYRI foresees that the ability of ArcWeld to display detailed information about arc and weld pool properties (such as the distributions of temperature and velocity) will give detailed insights into the complex phenomena occurring during welding, deepening their understanding of the welding process. Overall, CRRC QSYRI expects significant savings in time and materials costs, training of more highly skilled welders, and improvements in weld quality. In order to deliver on the customer requirements, the authors built on the existing ArcWeld software base. There were several areas of re-use, extension, and customisation that were addressed, and we discuss these in the following subsections.

3.1. Update of Workspace framework

As ArcWeld version 1 was built with Workspace 3.2, the first part of the development for ArcWeld version 2 was to upgrade to the latest available version of Workspace, version 5.4. Updating to the latest version of Workspace is straightforward - if the update is a point release change (e.g., from version 5.3 to 5.4), then there is very little to do due to the adherence to binary compatibility between point releases of Workspace. Upgrading from one major release to another (as in this case, from version 3 to 5) can require more work depending on the platform changes and the capabilities deployed by the application. However, the mature nature of the Workspace interface and development team support meant that the process of upgrading was relatively straightforward, with only minor changes needed. For example, for some deprecated operations, such as DataCollection, conversion to the new type (ObjectDictionary) happens automatically. A further advantage over developing the application from scratch is that stability updates in Workspace are essentially free improvements for the developer.

3.2. Re-use of existing components

Most of the components of original ArcWeld Workspace workflows are unchanged between version 1 and version 2. Consider, for example, Figure 2, which shows the workflow from version 1 that generates the solver input files. The vast majority of the components that create these files did not need to be altered. The only changes required were minor updates to a few components to reflect changes to the underlying solver (for example, a flag may have changed from default TRUE to default FALSE – such a change requires only that the developer unchecks a checkbox on the operation in the Workspace editor). Figure 3 shows version 2 of the workflow. This illustrates the closeness of the two workflows.

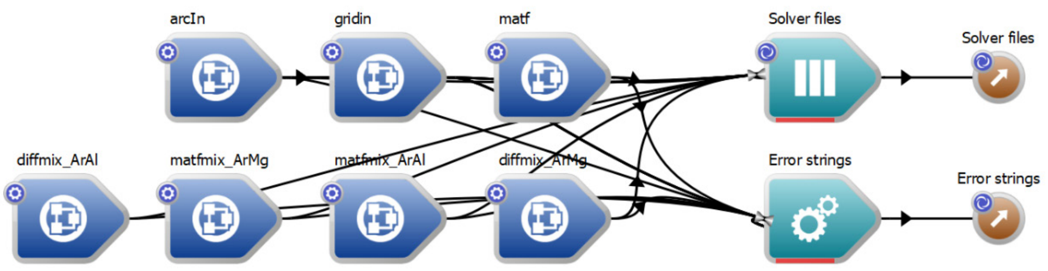
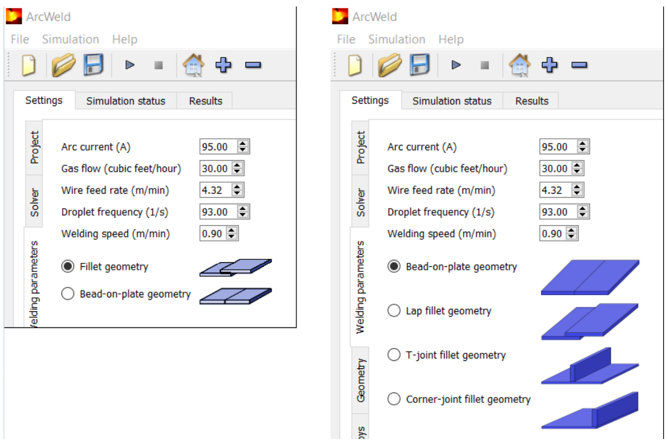


Figure 3. Main workflow for ArcWeld version 2



Arc current (A)	95.00
Gas flow (cubic feet/hour)	30.00
Wire feed rate (m/min)	4.32
Droplet frequency (1/s)	93.00
Welding speed (m/min)	0.90

[✓ Fillet geometry] Bead-on-plate geometry

Arc current (A)	95.00
Gas flow (cubic feet/hour)	30.00
Wire feed rate (m/min)	4.32
Droplet frequency (1/s)	93.00
Welding speed (m/min)	0.90

[✓ Bead-on-plate geometry] Lap fillet geometry T-joint fillet geometry Corner-joint fillet geometry

Figure 4. Version 1 (LHS) and version 2 (RHS) of the "Welding parameters" input fields

The GUI is the other major area of re-use – practically all components (check boxes, drop-downs, and other data entry fields) from version 1 exist in their original form in version 2. For users of version 1, this re-use of familiar interface components is valuable as it reduces the time needed to become accustomed to the updated software. A visual example of both re-use and extension of the GUI is presented in Figure 4. This shows the

"Welding parameters" tab in both version 1 (left-hand side, with just two configurations available) and version 2 (right-hand side, with four configuration – two old and two new).

3.3. Extensions to the software

Several extensions to the software have been made in version 2 of ArcWeld. Here we discuss some of the salient points by focusing on a subset of these.

Example – additional initial condition files

One of these extensions was driven by the requirement that, due to the extension of the list of metals available in the simulations, different versions of the matfmix file are required. The matfmix files contain tables of the arc plasma thermophysical properties (such as density, specific heat and thermal conductivity) for a wide range of temperatures and mixtures of the welding gas and the vapour evaporated from the metal surfaces (for example, argon and aluminium vapour). In the original workflow (Figure 2), it can be seen that there is a single operation labelled "matfmix" (which generates the contents of the file that can later be written to disk). In the newer version of the workflow for ArcWeld version 2, there are now two matfmix components, labelled "matfmix_ArMg" and "matfmix_ArAl", one for each different combination of gas (argon) and metal vapour (magnesium, aluminium). It is simple to add such additional workflow components using the Workspace editor without degrading the original functionality. The development steps were:

- Copying-and-pasting the original matfmix component. Users are very familiar with the concept of copy-and-paste as a means of replication, so this step is quite natural. This results in two copies of the matfmix creation operation.
- Re-labelling the two operations "matfmix_ArlAl" and "matfmix_ArMg". Labelling of the operations to match the generated solver input files is a deliberate choice, primarily aimed at facilitating ease of maintenance of the software.
- Updating a couple of text values in the operations inputs. These text inputs simply point to the new matfmix files. These files are contained in the resource pack for the application; Qt's resource system (The Qt Company Ltd, 2019) is used.
- Connecting the operations to the "Build array…" operation, which amalgamates all the files into one data structure (and array), which is then passed out of the workflow for subsequent use (when we wish to run the simulation and so create the files on disk)

Similarly, it can be seen in Figure 3 that the diffmix component, which tabulates arc plasma diffusion coefficients, has been split into two – the diffmix_ArMg and diffmix_ArAl operations. This was achieved in the same manner as for the matfmix operations.

Example – in-application graphing

The addition of in-application graphing was the single largest set of changes required to the existing ArcWeld version 1.0 code base. For an application built from scratch, this would be a very significant (and often infeasible) investment. The decision was taken to leverage the capabilities of the new Advanced Charting plugin from the Workspace ecosystem. This change was obviously not as trivial to effect as the matfmix file update discussed above, but nevertheless was still relatively simple. The main steps were:

- The solver team modified the Fortran code to output the results in the appropriate in-house format. For version 1 of ArcWeld, the solver used code to write data in a format recognised by Tecplot. For the purposes of in-application graphing, there were two main options considered, either of which the Workspace framework is well placed to support:
    - Write a Workspace operation to parse Tecplot files and produce Workspace-friendly data for plotting, or
    - Update the solver to directly output data in an easy-to-parse Workspace-friendly data format. This was the option chosen, for two reasons: it avoids dependence on a potentially changing file format (Tecplot), and is much simpler to implement, as we know everything about the data format and are in direct control of it.
- New ArcWeld Workspace operations were written to read each of the three output files that contains the data. The Workspace editor's wizards enable the creation of the necessary stub code, leaving the developer to just fill in the actual functionality. In this case, this involves parsing ASCII or binary

data as output by the solver, and populating one of Workspace's built-in data types appropriately (in this case, the multidimensional array data type ArrayNd was used, as this feed directly into the CreateChart operation).

- The data output from the new read operations (previous point) were connected to a CreateChart operation. Various configuration options are available on the CreateChart operation, relating to titles, labelling etc.
- A new widget was added in the GUI. This is achieved via Qt Designer (The Qt Company Ltd, 2018), which allows simple drag and drop of graphical components. In this case, the widget used is a Workspace-supplied ChartWidget.
- The workflow chart element was connected to the GUI chart widget. The workflow chart element is an output containing the chart data created by the CreateChart operation. The connection between these two items is simple to effect: the developer drags from the workflow (in the Workspace editor) to the chart widget (in Qt Designer). This process is described in more detail in (The Qt Company Ltd, 2018). It is worth reiterating here that one of the key benefits of using the Workspace environment is that it makes the process of connecting GUI and workflow components very simple.

4. CONCLUSION

In this paper we have outlined the process and drivers to extend a commercial-grade, standalone application using the Workspace workflow platform. The benefits of taking a simulation suite from a complex, expert-level, hard-to-use, and niche set of manual processes through to an easy-to-use, commercial-grade software application were enumerated. The main focus of the paper illustrates how an existing Workspace-based application could be easily extended, based on capabilities inherent to the Workspace ecosystem, to produce a customer-specific customisation of the original package. Benefits of using such a workflow approach were identified as; low financial cost; low resource cost (both in people and in time needed); re-use of existing components from Workspace; re-use and / or customisation of existing application-specific components.

ACKNOWLEDGMENTS

The authors greatly appreciate the financial support from the Rail Manufacturing Cooperative Research Centre (funded jointly by participating rail organisations and the Australian Federal Government's Business Cooperative Research Centres Program) through Project R3.4.1 – Arc Welding Modelling.

REFERENCES

Cleary, P., Hetherton, L., Bolger, M., Rucinski, C., Sankaranarayanan, N., Thomas, D., Watkins, D., Zhang, Z., Subramanian, R., Nguyen, D.Q. and McNally, M. (2017a) Workspace: Scientific Workflow Platform, version 13, CSIRO, https://doi.org/10.4225/08/5b03778185c17.

Cleary, P.W., Watkins, D., Hetherton, L., Bolger, M. and Thomas, D. (2017b) Opportunities for workflow tools to improve translation of research into impact. Paper presented at the 22nd International Congress on Modelling and Simulation, Hobart, Australia,

Murphy, A.B. (2011). A self-consistent three-dimensional model of the arc, electrode and weld pool in gas–metal arc welding. *Journal of Physics D: Applied Physics* 44(19), 194009.

Murphy, A.B. (2013a). Influence of droplets in gas–metal arc welding – a new modelling approach, and application to welding of aluminium. *Science and Technology of Welding and Joining* 18(1), 32-37.

Murphy, A.B. (2013b). Influence of metal vapour on arc temperatures in gas–metal arc welding: convection versus radiation. *Journal of Physics D: Applied Physics* 46(22), 224004.

Murphy, A.B., Nguyen, V., Feng, Y., Thomas, D.G. and Gunasegaram, D. (2017). A desktop computer model of the arc, weld pool and workpiece in metal inert gas welding. *Applied Mathematical Modelling* 44(1), 91-106.

Murphy, A.B. and Thomas, D.G. (2017) A computational model of arc welding – from a research tool to a software product. Paper presented at the 22nd International Congress on Modelling and Simulation, Hobart, Australia,

Murphy, A.B. and Thomas, D.G. (2019). A computational model of arc welding - from a research tool to a software product. *Mathematics and Computers in Simulation*, submitted.

Norrish, J. (1992). Advanced Welding Processes. Institute of Physics Publishing, Bristol, UK.

Patankar, S.V. (1980). Numerical Heat Transfer and Fluid Flow. Hemisphere, Washington DC.

The Qt Company Ltd (2018) Qt Designer, version 5.11, http://doc.qt.io/qt-5/qtdesigner-manual.html.

The Qt Company Ltd (2019) The Qt Resource System, version 5.13, https://doc.qt.io/qt-5/resources.html.