26th International Congress on Modelling and Simulation, Adelaide, SA, Australia, 30 Nov–4 Dec 2025
mssanz.org.au/modsim2025

Holistic naval ship design in CSIRO Workspace

D Rose*, P Hield*, M Bolger, L Hetherton**, D Thomas** and D Watkins****

**DSTG, Fishermans Bend*
***CSIRO, Data61, Clayton*

Abstract: Naval ships are highly complex and challenging to design, consisting of many integrated systems. The Defence Science and Technology Group (DSTG) is developing a holistic warship performance modelling program called InteShip, with the aim of supporting evidence-based decision making for naval ship concept design and through-life support. InteShip relies on CSIRO Workspace as the underlying software to build workflows for naval ship design. Workspace has many useful features that enhance InteShip functionality. DSTG and CSIRO have developed an ongoing collaboration which aims to enhance Australia's sovereign capability in naval ship design. This collaboration brings together software engineers and naval architects, providing the unique combination of skills necessary for this project. Future development will include expanding the repository of naval ship performance measures contained within InteShip, as well as using current and future Workspace features to enhance usability of the modelling and simulation tool.

Keywords: *Ship, Design, Modelling*

1. INTRODUCTION

Naval ships are highly complex and challenging to design. They consist of many integrated systems (see Figure 1) and must be capable of meeting a number of often conflicting requirements. Seemingly small changes to one system can have a significant effect on the overall design. A balance must be found which optimises performance whilst remaining within the constraints imposed by budgets and physics.

The traditional spiral design method (Andrews 1985) can struggle to converge on a balanced system. Decisions taken early in the design process can restrict the options available later and may lead to incorrect assumptions and late discoveries which can require substantial rework as the deadline approaches.

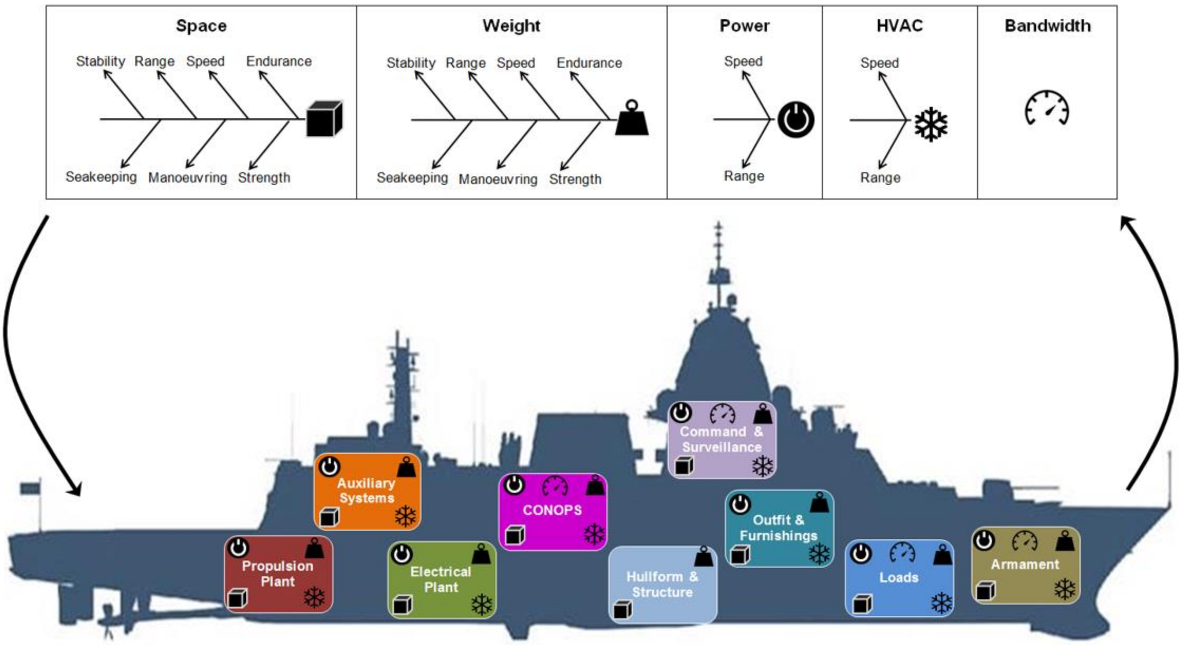

Figure 1 The extensive range of systems that a naval vessel requires to operate are all inter-linked and influence multiple performance aspects (Smith and Hield 2022).

The Defence Science and Technology Group (DSTG) is developing a holistic warship performance modelling program called InteShip, with the aim of supporting evidence-based decision making for naval ship concept design and through-life support. InteShip takes a set-based design approach (Scaled Agile Inc 2011), allowing many design options to be considered simultaneously. The pros and cons of each option can be considered, and the trade-offs between different requirements can be clearly understood before decisions are made.

InteShip consists of a collection of models for different naval ship performance metrics, e.g. range, endurance, habitability, top-speed, etc. These models are linked together to create a workflow, where the ship design parameters are the inputs and the outputs are the ship performance metrics (see Figure 2). InteShip is flexible, allowing different models to be included in the workflow depending on the question being considered or the input data available. The uncertainty in the performance predictions can be estimated, providing a level of confidence in the results. This can be used to focus future effort on the areas of the design which can have the most impact.

The individual models may be developed in-house at DSTG or may make use of commercial software packages accessed via an application programming interface (API). As such, individual modules may be developed in different programming languages and may use different external libraries. Therefore a platform is needed which would enable these different models to be easily interconnected. Ideally this would happen in a simple Graphical User Interface (GUI) environment. DSTG chose to evaluate the Workspace platform developed by the Commonwealth Scientific and Industrial Research Organisation (CSIRO) for this task.

2. WORKSPACE

Workspace is a scientific workflow and application development framework that has been under development and use in the Computational Model Group of CSIRO since 2005 and was externally released in 2014 (Watkins, et al, 2023, Cleary, et. al. 2020, Cleary, et. al. 2019, Hetherton, et. al. 2019, Oakes, et. al. 2019, Cleary, et. al. 2017, Watkins, et. al. 2017, Cleary, et. al. 2015). The Workspace design goals have four key pillars which are summarised as: Analyse, Collaborate, Commercialise and Everywhere:

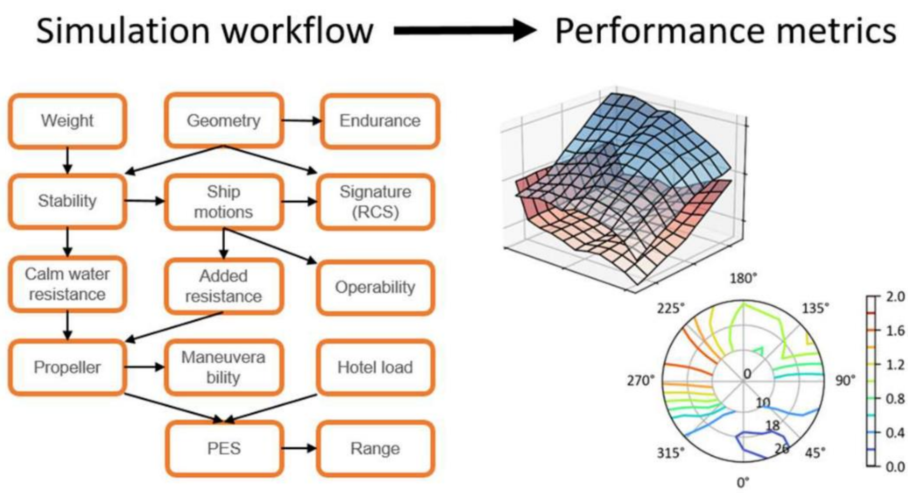

Figure 2 Example workflow showing several connected models with performance metrics calculated by running the workflow for a range of inputs (Smith et. al. 2022).

1. Analyse – increase user productivity and reproducibility. Users can visually create workflows by connecting pre-existing operations on a canvas and then executing these workflows. These workflows can then be saved (as XML files) and rerun later. Users can attach widgets to Inputs and Outputs on the workflow to adjust values and interactively visualise results.
2. Collaboration – Make collaboration easier by allowing the simple reuse of workflows and operations. As workflows are simple XML files, these can be copied between different users/environment and subsequently executed, with one caveat – all the operations in the workflow need to be instantiated on any target system. Operations can be added to Workspace via its plugin architecture. Once added, operations (or widgets or datatypes) can be used just as if they were created and supplied as part of Workspace.
3. Commercialisation - Workspace is built on software that implies no inherent restrictions on users, i.e. it does not use commercial products which could incur costs or use software with "viral" licenses which would thereby impose licensing restrictions on users.
4. Everywhere – Workspace provides runtimes which can execute workflows in multiple environments, for example Workspace has the ability to execute on Mac, Linux and Windows and provides distributed executed scenarios.

3. BENEFITS OF WORKSPACE FOR DSTG

Workspace is well suited to DSTG requirements as the modular operations of Workspace are well aligned to the InteShip architecture. When talking about InteShip we talk about "models" – these are small units of work that perform specific tasks including assessment of power and energy performance or calculating the range. In Workspace, at a higher level, a "workflow" is a collection of models to create a simulation. These modules are normally created in Python but make use of several commercial software packages. Workspace's strong support for Python allows the current InteShip developers to leverage existing skills as well as the possibilities of attracting new developers. The Workspace User Interface (UI) supports drag-and-drop of data paths between operations which provides visual feedback as to the relationship between operations. The ability for Workspace to run as a server under Windows, Linux and Mac will allow InteShip to be executed in high performance and distributed compute environments when required with no code changes.

4. WORKSPACE GUI

The Workspace GUI allows our model developers to create a library of models and join them into a workflow. The Naval Architects combine individual variables into "datapacks" which are python dictionaries. These datapacks reduce the number of "inputs to" and "outputs from" models from, in some cases, more than 100 to a manageable number. The Naval Architects using Workspace have the ability to drag inputs, outputs and models around the canvas to visualise the workflow.

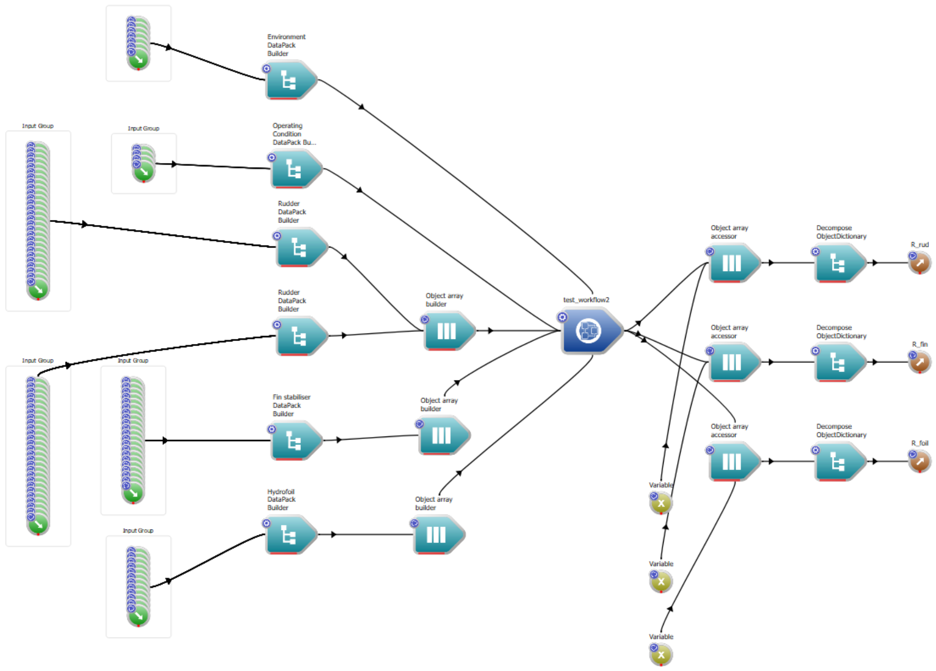

Figure 3 Example model created in Workspace, making use of datapacks

5. WORKSPACE SUPPORT FOR PYTHON

Although Workspace itself is written in C++ using Qt and supports native operations, datatypes, and widgets through C++ plugins, it also provides excellent Python support as an alternative extension mechanism. This strong Python integration has enabled a smooth transition of models from the team's previous environments to Workspace. Most of the models rely on Python packages as part of their processing, and while Workspace ships with its own release of Python that can be enhanced using the standard pip-installs-python (pip) package manager, it can easily be directed to use other compatible Python versions on the system, including Python virtual environments. Python is the programming language of choice for scientists and engineers as it's easy to learn and there are many libraries available, making this flexibility particularly valuable for accommodating diverse research environments and dependencies.

Workspace's embedded Python scripting capability provides bidirectional data exchange between the interactive visual workflows and Python code through the RunPythonScript operation. This operation supports both inline scripts and external Python files, with comprehensive type mapping that automatically converts between Workspace's native data types (including primitives, compound types, lists and dictionaries, Qt types, JSON documents, 3D geometry data, and NumPy arrays) and their Python equivalents. The system features dynamic I/O port creation, allowing users to configure arbitrary named inputs and outputs through the operation's properties interface, with each port automatically mapped to corresponding Python variables in the script's namespace. Advanced features include persistent namespaces for maintaining Python state between executions, configurable module search paths supporting both filesystem and Qt resource locations, GUI thread execution for Python packages requiring main thread access, and robust error reporting. This architecture enables researchers to leverage Python's extensive scientific computing ecosystem within Workspace's visual workflow environment, facilitating integration of custom algorithms, data analysis routines, and third-party Python libraries into complex multi-step processing pipelines while maintaining the visual workflow paradigm for overall process orchestration.

6. PYTHON CALLING WORKSPACE

There is also the ability to call a Workspace workflow from Python, using the Workspace-Python interface (Workspace Python Interface 2025). This has allowed our python developers to create a custom UI for InteShip (Figure 4). This UI allows appropriate QT Widgets to be assembled in a logical layout to create a UI that can guide users to enter valid data without the need for error messages. There are also visual cues provided by the UI to the user to confirm the data choices made.

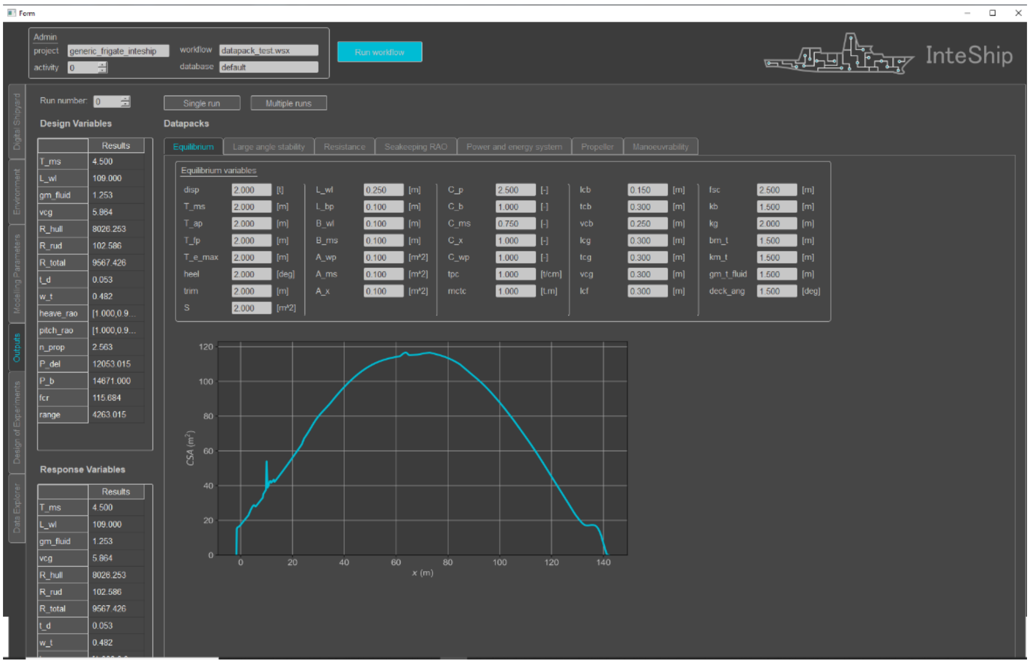

Figure 4 InteShip GUI allowing a user to access an InteShip model built in Workspace.

The Workspace-Python interface is a lightweight Python module that allows Workspace workflows to be configured and executed from within Python code. Each executed Workspace workflow runs as an independent child process; an instance of the workspace-web executable, insulating the calling process from any errors that occur, and simply enabling individual user sessions to be bound to separate running workflows. The python module binds to the Workspace C-API which it uses to interact with the running processes via Transmission Control Protocol Internet Protocol (TCP-IP). While originally intended for continuously running workflows embedded in web applications, the interface has been used to great effect here combining external Python code with Workspace workflows. In the near future, this TCP-IP interface will be replaced with a direct interface using pybind11, which will greatly improve performance for applications that are highly communicative with their underlying workflows.

The Python interface provides methods that enable two different application design techniques, depending on the application requirements:

- **Sequential programming:** workflow inputs can be set prior to execution, the workflow can be executed, and results inspected once execution is complete. This is most effective for writing tests where interactivity is not required during the execution process.
- **Concurrent programming:** workflow inputs can be set at any time – before or during execution – with watch callbacks used to obtain outputs (or intermediate data via Workspace "global names") as soon as they are up-to-date. The Workspace execution engine guarantees that the data is safe to access for the duration of the response callback. This method is useful for applications that rely on asynchronous user-driven events, where different parts of the application must be updated asynchronously to mirror different aspects of the workflow state.

7. PARAMETER SWEEP TOOL

A large amount of analysis performed by InteShip users is "what if", for example what if the cruising speed is changed, what if there are environmental changes (temperature, wave height, etc) will the ship perform within its operating limits. The Parameter Sweep Tool (Figure 5) that comes with Workspace allows the flow to be executed with a changing set of values and capture of the results for further analysis

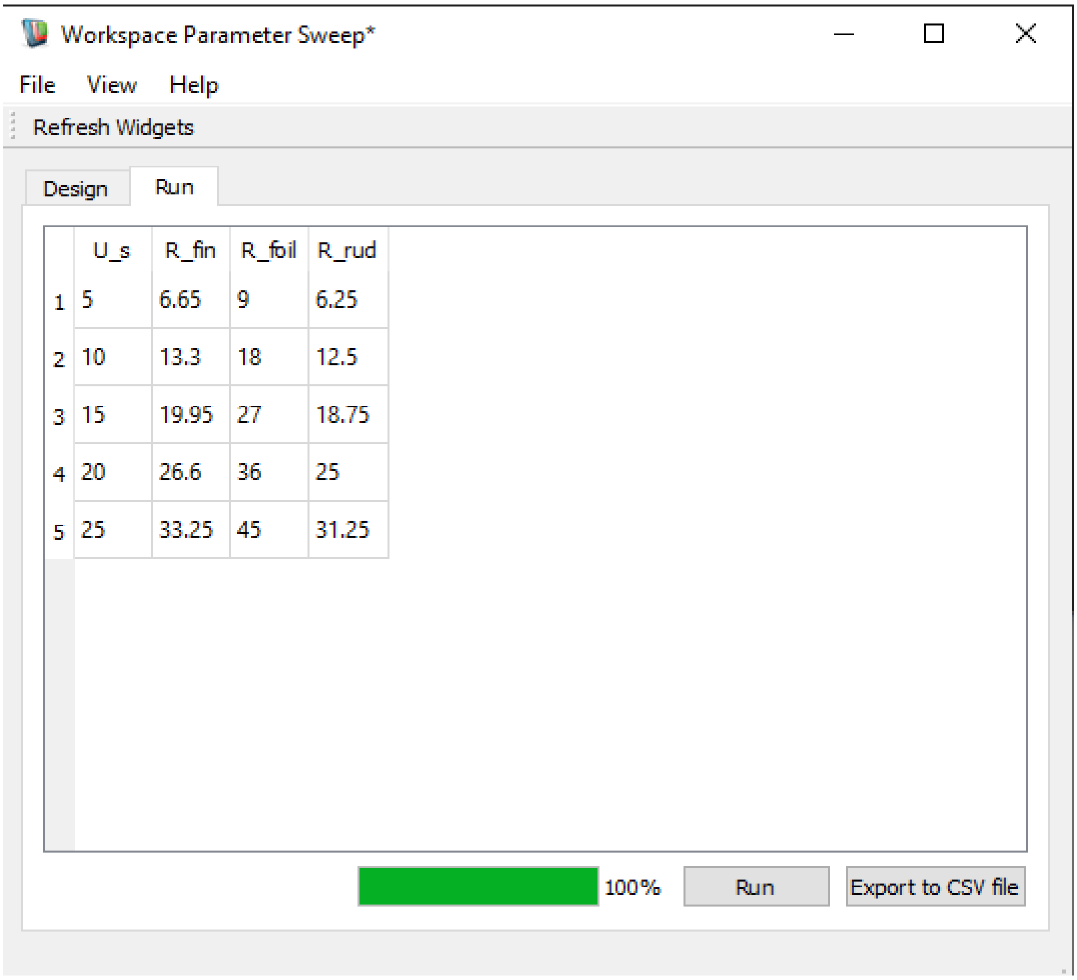

Figure 5 Workspace parameter sweep tool

8. CURRENT STATE OF PLAY

The one-for-one migration from the previous tool has been successfully completed. The focus has now shifted towards enhancing InteShip with new features, such as a user interface, and migrating Excel spreadsheets into a popular database. Additionally, there is a commitment to creating new models to address the ever-evolving simulation requirements and Defence's innovative methods for evaluating fleet performance.

9. FUTURE WORK

Our future work is focused on being able to deploy InteShip outside our DSTG science and technology team, this will mean having an installer that will bring Workspace, InteShip and associated software to external users – a need supported by Workspace's packaging capability. A more extensive pybind11 interface is currently being developed by CSIRO's Workspace team to support more powerful execution of Workspace workflows from Python environments. This includes enhanced capabilities for introspection of workflow structure and simpler access to any workflow element or data, which will be of particular interest for the InteShip team's planned dynamic Python scripts and Python-based applications. The teams are also investigating how to integrate Workspace and Python in a more compatible way to support other Python Qt modules such as QtPy, PySide, and the Spyder IDE, alongside the ongoing migration to Qt6.

There is also expectation that flows and models will become more compute intensive in the future and that execution of InteShip will need to include processing beyond Windows desktops to High Performance Computing (HPC) clusters. Workspace's existing parallel processing capabilities, which include local execution, network distribution, and support for HPC clusters like Simple Linux Utility for Resource Management (SLURM), will prove invaluable as InteShip's simulation use cases become more complex and require high-performance computing environments.

10. CONCLUSION

DSTG has transitioned InteShip to use CSIRO Workspace as the underlying software to build workflows for naval ship design. Workspace has many useful features that enhance InteShip functionality, including linking models, calling other software, GUIs and the parameter sweep tool.

DSTG and CSIRO have developed an ongoing collaboration which aims to enhance Australia's sovereign capability in naval ship design. This collaboration brings together software engineers and naval architects, providing the unique combination of skills necessary for this project. Future development will include expanding the repository of naval ship performance measures contained within InteShip, as well as using current and future Workspace features to enhance usability of the modelling and simulation tool.

REFERENCES

Andrews, H. (1985) "An Integrated Approach to Ship Synthesis", Transactions from Royal Institute of Naval Architects. Volume 128, Pages 73-102

Cleary, P. W., Thomas, D., Hetherton, L., Bolger, M., Hilton, J. E., and Watkins, D., (2020) "Workspace: a workflow platform for supporting development and deployment of modelling and simulation", revised for: Mathematics and Computers in Simulation, (2020). Volume 175, 2020, Pages 25-61, ISSN 0378-4754, https://doi.org/10.1016/j.matcom.2019.11.011.

Cleary, P.W., Hetherton, L., Thomas, D.G., Bolger, M. and Watkins, D. (2019), "Translation of stranded IP to commercialisable software applications using Workspace", 23rd International Congress on Modelling and Simulation, Canberra, ACT, Australia, 1 to 6 December 2019, https://doi.org/10.36334/modsim.2019.D2.cleary

Cleary, P. W., Watkins, D., Hetherton, L., Bolger, M., and Thomas, D. (2017)"Opportunities for workflow tools to improve translation of research into impact", 22nd International Congress on Modelling and Simulation, Hobart, Tasmania, Australia, 3 to 8 December 2017 https://mssanz.org.au/modsim2017/Keynote/cleary.pdf

Cleary, P.W., Thomas, D., Bolger, M., Hetherton, L., Rucinski, C., and Watkins, D., (2015) "Using Workspace to automate workflow processes for modelling and simulation in engineering", 21st International Congress on Modelling and Simulation, Gold Coast, Australia, 29 Nov to 4 Dec 2015 https://www.mssanz.org.au/modsim2015/C5/cleary.pdf

Hetherton, L., Bolger, M., Zhang, Z., Oakes, N., Watkins, D. and Cleary, P.W (2019), "Workspace for Industry 4.0", 23rd International Congress on Modelling and Simulation, Canberra, ACT, Australia, 1 to 6 December 2019, https://doi.org/10.36334/modsim.2019.D2.hetherton

Oakes, N., Hetherton, L., Bolger, M., Thomas, D.G., Rucinski, C., Watkins, D. and Cleary, P.W., (2019) "Workspace – a Scientific Workflow System with commercial impact", 23rd International Congress on Modelling and Simulation, Canberra, ACT, Australia, 1 to 6 December 2019 https://doi.org/10.36334/modsim.2019.D2.oakes

Scaled Agile Inc, (2011), "Set-Based Design", [Accessed 2020 11 February], Available from: https://www.scaledagileframework.com/set-based-design/.

Smith, S. and Hield, P., (2022), "Integrated ship performance modelling (InteShip): Model methodology", DSTG-TR-3974, DSTG, Melbourne, Australia

Smith, S., Patterson, N., Rayes, C. and Hield, P., (2022), "Naval platform design considerations for meeting increased hotel load demand", 2022 International Maritime Conference, Sydney, Australia, 10 to 12 May 2022

Watkins, D., Bolger, M. and Zhang, Z. (2023), "Workspace and the Mixed Reality Lab: Where scientific workflows meet Industry 4.0", 25th International Congress on Modelling and Simulation, Darwin, NT, Australia, 9 to 14 July, https://doi.org/10.36334/modsim.2023.watkins

Watkins, D., Thomas, D., Hetherton, L., Bolger, M., Cleary P.W. (2017), "Workspace - a Scientific Workflow System for enabling Research Impact" 22nd International Congress on Modelling and Simulation, Hobart, Tasmania, Australia, 3 to 8 December 2017 https://mssanz.org.au/modsim2017/C3/watkins.pdf

Workspace Python Interface, (2025), "README", [Accessed 2025 8 August], Available from: https://github.com/csiro-workspace/workspace-python?tab=readme-ov-file#readme

