

OneWorld: Evolving an architecture for complex system representation in engagement simulation

R.D. Armstrong^a, **M.J. Christie**^a, and **P.R. Mellen**^a

^a *Weapons and Combat Systems Division, Defence Science and Technology Group, Edinburgh, South Australia*

Email: OneWorldAdmin@dst.defence.gov.au

Abstract: The Weapons and Combat Systems Division (WCSD) within the Defence Science and Technology Group are responsible for providing advice to the Australian Defence Force (ADF) on the performance of Australian weapons systems. These weapons systems cover a range of system types including surface ships, aircraft, and ground based assets. This advice is used to inform the capability acquisition process and how those capabilities could be best employed by the ADF. WCSD has developed a range of modelling and simulation (M&S) tools in order to undertake analysis to underpin this advice. In order to streamline aspects of this development pipeline and simplify integration efforts WCSD developed a composite model architecture called OneWorld.

OneWorld was developed to describe and define a series of modelling interfaces and datatypes which could be used to integrate new models. It also provides an initial set of pre-integrated models that can be used to represent generic systems. It was created with three core design goals:

- Simplify the process of integrating new models into a simulation.
- Promote model re-use through ease of integration.
- Reduce maintenance requirements across the M&S toolset.

As the problem space being analysed by WCSD has continued to grow in scope and complexity, there has been greater demand on the M&S toolset to represent a larger number of systems with greater fidelity. OneWorld now requires major updates to ensure it can continue to support these activities into the future. Whilst ongoing development has allowed OneWorld to remain a key component of WCSD's M&S toolset there are a number of limitations with the architecture. These are becoming more pronounced as the scope and complexity of simulations increases, and as more external partners apply OneWorld to meet their own needs. Three key limitations include:

- **Flexibility.** Whilst having fixed interfaces and datatypes guarantees that any model that is using a particular OneWorld signal is sending the expected data; it limits how readily new signals can be developed as model requirements evolve.
- **Extensibility.** The well-defined interfaces guarantee interoperability between models however this does impose a restriction on model developers and can encourage users to integrate their models at higher levels in the hierarchy to avoid the constraints.
- **Scalability.** OneWorld is built to execute models within SimFramework. This currently requires the entire world to be run as a single process limiting the scale and complexity of simulations that can be supported.

Despite these limitations OneWorld provides key capability to WCSD and must be further developed to meet future requirements. In doing so WCSD, in collaboration with their partners, continue to evolve a robust M&S capability for engagement simulation that provides valuable analysis of ADF capability.

Keywords: *Integrated modelling, engagement modelling, computational simulation, modelling architecture*

1. INTRODUCTION

The Weapons and Combat Systems Division (WCSD) within the Defence Science and Technology Group (DSTG) are responsible for providing advice to the Australian Defence Force on the performance of Australian weapons systems. These weapons systems cover a range of system types including surface ships, aircraft, and ground based assets. This advice is used to inform the capability acquisition process and explore how those capabilities could be best employed by the ADF. WCSD has developed and uses a range of modelling and simulation (M&S) tools in order to undertake analysis to underpin this advice. In order to streamline aspects of this development pipeline and simplify integration efforts WCSD developed a composite model architecture called OneWorld. As the problem space being investigated by WCSD has continued to grow in scope and complexity, there has been greater demand on the M&S toolset to represent a higher number of systems with greater fidelity. OneWorld now requires major updates to ensure it can continue to support these activities into the future.

2. BACKGROUND

As M&S grew in importance as an analytical means for WCSD, many complimentary tools were developed. Each of these tools had their own bespoke simulation setup and unique, individual use-case. Some were directly hard coded as one-vs-one simulations, whilst others were more scalable to few-vs-few. Eventually many of these were brought together into a more cohesive suite of tools, architectures and frameworks for modelling and simulating operational scenarios.

SimFramework is a simulation engine for running models that was internally developed by WCSD. Models designed to be run in SimFramework are built using Mars, a set of C++ libraries used to define a model's parameters, signals and functions to ensure compatibility with SimFramework and interoperability with other Mars models. MarsGen is a model specification application that allows modellers to define their model design and automatically generates a Mars C++ model skeleton. This allows the modellers to focus on including their model specific mathematics or logical functions, rather than surrounding software infrastructure.

Prior to the development of OneWorld, WCSD engagement modelling was predominantly focused on relatively small-scale simulations that typically included only a few models. Each simulation was bespoke and required a "World" model to be built comprising only the necessary models for that simulation. This World model was a composite¹ model that created the model instances and managed the routing of signals between models. An example of a simple composite World model is shown in Figure 1.

An issue with this process was that each new set of analysis requirements often necessitated changes or additions to the simulation usually requiring a new World model to be created or an existing one to be modified. This process was often time intensive, expensive, error prone and delivered an inherently inflexible model (Fletcher *et al.* 2009).

Differences in the signal specifications between models also meant that often models could not communicate with each other without specialised wrapper classes. These wrapper classes translated the input and output signal specifications between models (Fletcher *et al.* 2005).

Both prior to initial development, and over the intervening years, WCSD has worked with its partners, domestic and international, to maintain an awareness of modelling tools and architectures that could provide similar functionality. Unfortunately a solution could not be found that provided the flexibility and reusability that WCSD required. Since the initial development of OneWorld others have developed toolsets to tackle similar model integration problems. An approach used by an international partner has been to develop platform level architectures to model all required subsystems in a single modelling environment such as Simulink. This approach was not suitable for the WCSD use case as WCSD required a solution that readily allowed integration

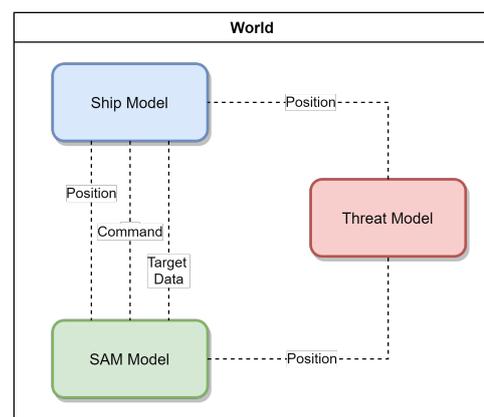


Figure 1. World composite model for simple engagement simulation

¹ A composite model is a parent model within a hierarchical structure that encapsulates child models. Child models may be additional composite or atomic models (no further decomposition, leaf nodes).

of models developed in a variety of environments. Other partners have opted to retain a more bespoke workflow where significant time is invested in the development of a world for a fixed set of models. Whilst this approach is valid and could be readily scoped for individual contracts or activities, it lacked the broad flexibility desired by WCSD.

3. DEVELOPMENT OF ONEWORLD

3.1. Design Goals

OneWorld was developed as a modelling architecture to describe and define a series of modelling interfaces and datatypes which could be used to integrate new models. It also provides an initial set of pre-integrated models that can be used to represent generic systems. Design goals for OneWorld were generated from consideration of future analytical requirements, and reflection on the previous decades of engagement modelling and simulation. The three core design goals were to:

- Simplify the process of integrating new models into a simulation
- Promote model re-use through ease of integration
- Reduce maintenance requirements across the M&S toolset

Integration of new models into existing simulations was eased by creating a world that could scale the number of entities in the simulation and automatically configure their signal wiring with other models. Model integration was also simplified by providing helper functions that in conjunction with fixed interfaces (Luckman *et al.* 2009) would allow modellers to redefine the world or a model's composition dynamically.

OneWorld promotes the re-use of models by making the integration process a one-off with the outcome being a model that can be easily incorporated into any OneWorld based simulation. This was achieved by using a model specific wrapper derived from a OneWorld interface (each wrapper is responsible for converting from model specific data definitions to OneWorld compliant definitions). As a practical example of this; if a sensor model was integrated into OneWorld using the OneWorld Sensor interface, it would then be able to be re-used anywhere a Sensor leaf node could be placed in the hierarchy.

Maintenance requirements were reduced by having one reconfigurable and consistent model environment across multiple M&S applications enabling 'integrate once, use anywhere'. This is demonstrated by the ability for OneWorld models to be used in both internal 'few-vs-few' M&S environments as well as in external partner many-vs-many M&S environments (further described in 3.4).

3.2. OneWorld Structure

OneWorld is effectively a hierarchy of models designed to decompose complex military capabilities. Through a variety of helper classes, nested composite models and well-defined interfaces, OneWorld creates a reconfigurable hierarchy of models. This removes the need for bespoke world models to be created for each scenario. OneWorld allows for the insertion and removal of models at a desired level and the ability to create whole new system representations by defining them as a collection of subsystem models.

The OneWorld hierarchy is designed to be able to describe most scenarios from broad World models, down to specific sub components of a complex platform. The current OneWorld hierarchy is shown in Figure 2. The levels of the hierarchy are:

- **World:** The World level contains the signals that would need to be communicated between opposing groups (teams) within the simulation (entity position information) and global models like atmosphere.
- **Group:** The Group level represents the opposing forces in the simulation. It contains the signals that would be passed between entities in the same group, for example shared track data.
- **WeaponSystem:** The WeaponSystem is the representative level for each entity in the simulation (i.e. a single ship). Despite the nomenclature, a WeaponSystem does not need to represent offensive capabilities. A radar station or civilian aircraft would also be represented by a WeaponSystem in a OneWorld based simulation.
- **WeaponOuter:** The Weapon is the set of models representing munitions. This could be multiple models, for example an Australian Air-Warfare Destroyer may have multiple models representing its full complement of munitions including missile and gun systems.
- **System:** The System is the set of models and signals that represent the non-munition components of the entity.

- **Sub-system models:** Sitting under the System level, these sub-system models include platform motion dynamics and any combination of sensor models, combat management system models, and launcher model. This level represents where the atomic functions of the entity sit.

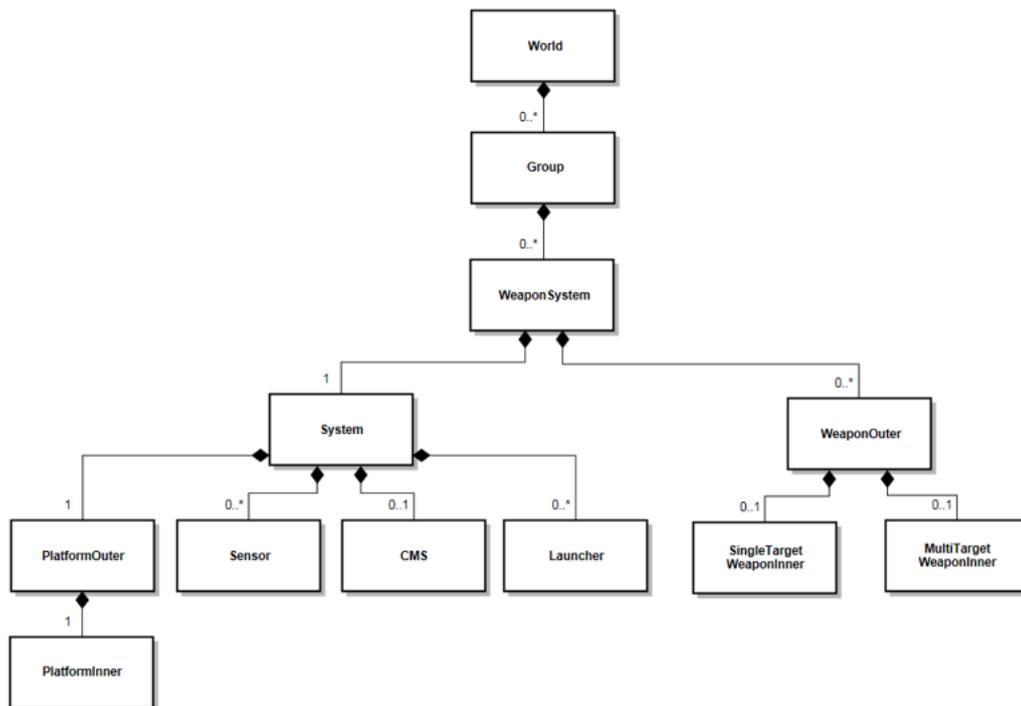


Figure 2. Diagrammatic representation of the OneWorld hierarchy as it currently exists

3.3. Model Integration

Models of systems can be integrated into the OneWorld hierarchy at the WeaponSystem, System, or SubSystem level. By allowing integration of models at various levels, OneWorld provides additional flexibility in how models are integrated into a world or simulation. The process of integration is different for models developed internally (with Mars), and third party models developed externally.

- **Internally Developed Models**
 1. Firstly the Function and Signal Specification (FASS) for the model is defined in MarsGen and a high-level program structure is generated. As part of this step the modeller selects which OneWorld interface best represents the model.
 2. The logic and mathematics of the model are developed within the predefined locations in the high-level structure and are compiled into a C++ model library.
 3. Now the model (and its dependencies) can be added to any OneWorld simulation.
- **Third Party (externally) Developed Models**
 1. A determination is made as to where in the OneWorld hierarchy the software model should exist. For example a model of an entire aircraft would be represented as a WeaponSystem whilst a model of a ship radar would be represented as a SubSystem.
 2. A ‘Shell’ for the model is generated in MarsGen with the OneWorld interfaces corresponding to the selected position in the hierarchy.
 3. Code is developed to couple the inputs, outputs, and configuration of the model being integrated to the OneWorld interface.
 4. Now the model (and its dependencies) can be added to any OneWorld simulation.

Once integrated into the OneWorld architecture a model can now be used as part of the OneWorld composite model, regardless of the development origins of the model. An example of this is shown in Figure 3. In this example in a single OneWorld World and Group there are two entities. One is an industry provided model of a bomber aircraft which has been integrated into the architecture at the WeaponsSystem level. The other is a composite model of an Air Warfare Destroyer composed of a surface-to-air missile (SAM model) provided by Partner A, a ship-motion model provided by Partner B, a phased-array radar (PAR) model provided by Partner C, and an internally developed Mars model for the combat management system (CMS). This flexibility

provided by the OneWorld architecture, to create composite models from disparate sources, aids multi-partner efforts towards co-development of models representing complex military entities.

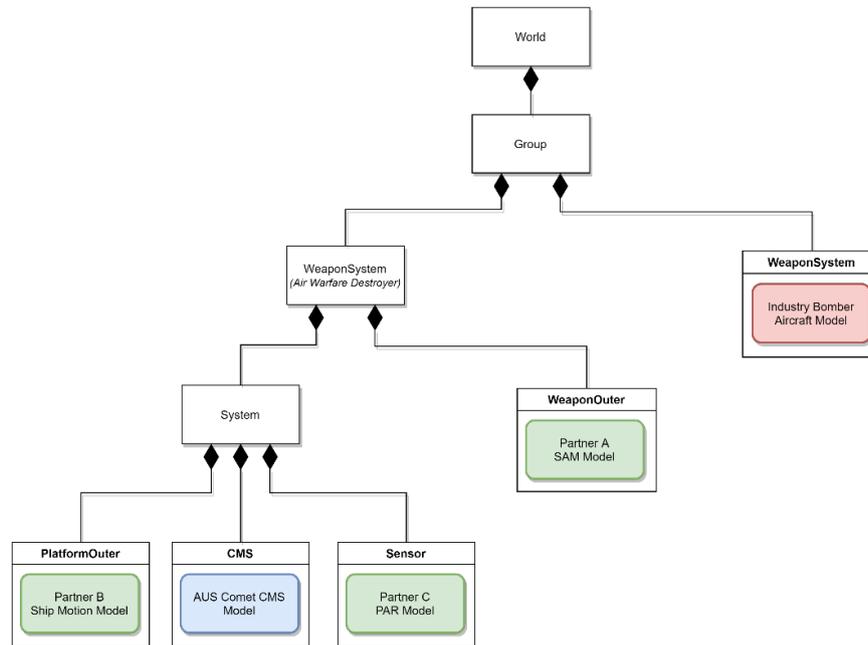


Figure 3. OneWorld hierarchy for a world containing two entities; an Air Warfare Destroyer and a Bomber Aircraft.

3.4. Third Party Simulation Environment

Due to strategic drivers, including access to partner entity models, increasing collaboration, and a desire for a distributed simulation capability into the future, there was a need to perform analysis in new simulation environments. It was decided that OneWorld would be integrated into the new simulation environments at the WeaponSystem level. This allowed retention of the plug-and-play, and model reuse, capabilities within both the existing WCSD, and new third party, simulation environments.

An example of how OneWorld models have been integrated into third party simulation environments is shown in Figure 4. Here the simulation environment uses a publisher-subscriber framework to pass inputs and outputs to the various entities in the environment.

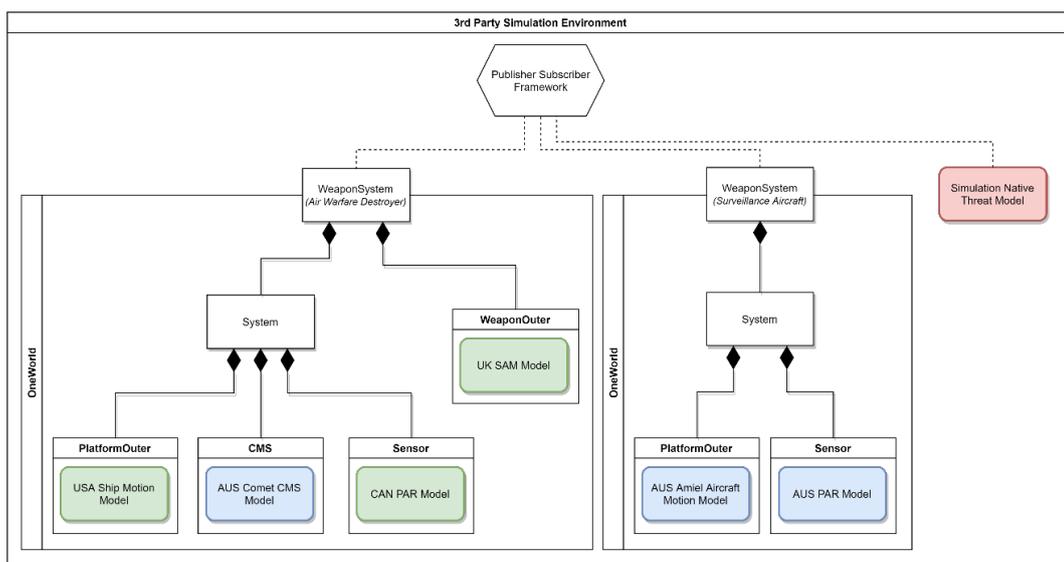


Figure 4. Multiple OneWorld models inserted into a third party simulation environment alongside models native to this environment

This framework allows each of these models to be run as distributed processes. The previously described OneWorld model of the Air-Warfare Destroyer has been integrated into the simulation at the WeaponsSystem level of the hierarchy. A OneWorld model of an airborne radar has also been integrated. Each OneWorld model runs a separate instance of SimFramework. Both of these models are able to function alongside a library of other models within the simulation environment, irrespective of whether they were developed in OneWorld, natively in the simulation environment, or developed elsewhere and wrapped into the simulation environment. If the CMS model were to be updated to include new algorithms and functionality it could easily be replaced to update the composite models being used in any of the simulation environments.

4. IMPROVEMENTS FOR THE FUTURE

Since its initial design OneWorld has continued to be used and developed by WCS D. Over the years its employment has evolved to meet the changing analytical needs of the ADF. Whilst ongoing development has allowed OneWorld to remain a key component of WCS D's M&S toolset there are a number of limitations with the architecture. These are becoming more pronounced as the scope and complexity of simulations increases, and as more external partners apply OneWorld to meet their own needs. Identification of limitation and design goals for the future of OneWorld are generated through ongoing consultation with OneWorld users, both within DST Group and industry, and through consideration of the future analytical requirements of defence clients. Three key limitations have been identified as priorities for the development of OneWorld, summarised in the following sections.

4.1. Flexibility

Using fixed interfaces and datatypes guarantees that for any model that is using a particular OneWorld signal, it is sending the expected data. The interface definition is also a way of formally ensuring that two models have the same interpretation of any data sent and received. For example, a sensor model sending track data on the Tracks signal provides all the information as required by the Track data type and any model receiving that data can be confident it is receiving all the information that is expected on a Track input signal.

This rigid form of signal and data type coupling means that what a data type is, for all our simulations, has been codified. This is particularly noticeable for complex data types like track data. By using the OneWorld Track datatype, modellers are locked into track data meaning exactly one thing for every OneWorld simulation. Despite the goal of being able to create one architecture for many different domains, it can be seen that the definition of a track could change between one-vs-one and many-vs-many scenarios. For example in a one-vs-one scenario there may be no need to provide track quality data where as in a many-vs-many scenario off-board tracking and data links may require additional fields.

A solution for strongly typed data signals would be to provide a "Track data signal" that does not use a fixed data definition and leaves it up to the modeller to ensure both models (sender and receiver) agree on a definition for what track data is comprised of. The down side to this approach is that it would provide less programmatic guarantees that two models communicating are interpreting data the same way.

4.1. Extensibility

Using fixed, well-defined interfaces ensures model interoperability within OneWorld. Unfortunately this unintentionally encouraged users to integrate their models at higher levels in the hierarchy to avoid the constraints added by the interface definitions. Any change to an interface requires it to be reflected back onto all existing models so the addition of a new signal or change to datatype would require all models implementing that interface to be updated even if those models will not use the additions. This also leads to collections of subsystem models being integrated 'together' which does not promote sub-system model re-use which was one of the design goals of OneWorld.

To make the evolved OneWorld more extensible a way of extending interfaces needs to be included without incurring the development cost of having to update each and every model that uses the same interface. Backwards compatibility of old models with new interfaces could be achieved by using versioned interfaces. This would mean that if an additional signal added to the interface is not used by an already integrated model the model would not need to be reintegrated.

4.1. Scalability

As the scenarios that need to be analysed through simulation grow larger and more complex, the scalability of the M&S solution becomes more of an issue. Due to how tightly coupled Mars, OneWorld, and SimFramework

are, the issue of scalability lies beyond only OneWorld. An example of how the ability to scale has impacted M&S capabilities is the limit on the number of missile models that can be placed on a scenario entity. Due to how signal coupling is handled in SimFramework, each additional instance of a missile model requires an increasingly longer time to initialize. If a series of successive raids (repeated incoming threats over a period of time) needs to be simulated, the amount of time needed to instantiate the simulation can become impractical when doing thousands or tens-of-thousands of runs with upwards of 40 missiles per platform. Analysts have been able to adapt to these limitations by splitting scenarios to minimize the number of missile models initialized at any one time. Reconstitution of these scenarios from their separate parts places an additional burden on data scientists.

One way in which scalability could be improved could be through updates to, or a complete re-engineering of, SimFramework. Other possibilities include investigating discrete event simulation tools other than SimFramework that can be integrated into our modelling and simulation environment. While it is early in the redevelopment process, it is clear that to solve OneWorld's scalability issues requires consideration of all aspects of the M&S environment.

5. CONCLUSION

The OneWorld architecture was developed to simplify the process of integrating models into simulations, promote re-use of models, and to reduce maintenance requirements across the M&S toolset. Whilst it achieved these goals, the use case for the architecture has evolved over time. It now requires development to meet future requirements. This development will need to improve the flexibility, extensibility and scalability of the architecture. In doing so WCSD, in collaboration with their partners, will evolve a robust M&S capability for engagement simulation that continues to grow and provide valuable analysis of ADF capability. Throughout this development, WCSD would like to broaden their collaboration with the engagement modelling community to obtain peer feedback of this architecture definition process and to maintain awareness of similar activities being conducted elsewhere in this research space.

ACKNOWLEDGMENTS

The authors would like to acknowledge the effort contributed by many people within the Defence Science and Technology Group over the years to the initial design and continued development of OneWorld.

REFERENCES

- Fletcher, D., Luckman, N., Hodson, M., 2005. Principles of simulation architecture-independent model development. Simulation Technology and Training (SimTecT) Conference.
- Fletcher, D., Hodson, M., Luckman, N., 2009. Advances in simulation architecture-independent model development. Simulation Technology and Training (SimTecT) Conference.
- Luckman, N., Hodson, M., Fletcher, D., 2009. Arbitrary composition of component models in hierarchical simulations. Simulation Technology and Training (SimTecT) Conference.