STRATEGIC AIRLIFT OPERATIONALIZING CONSTRUCTIVE SIMULATIONS

Rob Barwell, and Gabriel Wainer

Dept. of Systems and Computer Engineering

Carleton University

Ottawa, Canada

robbarwell@cmail.carleton.ca, and gwainer@sce.carleton.ca

ABSTRACT

Large organizations now use Modeling and Simulation (M&S) for complex systems. This evolution has also resulted in a lack of simple tools and processes to define models and experiments which hinders wide scale user adoption of M&S to solve many real-world problems. By exploring the Arctic logistical support base problem, we will illustrate how a simple M&S process and system can be used by operators with limited knowledge of M&S. This will be demonstrated by determining a temporary logistical support base for re-supplying military operations in the Canadian Arctic.

Keywords: Strategic Airlift, Canadian Arctic, DEVS, JSON

1   INTRODUCTION

Large organizations now use Modeling and Simulation (M&S) of complex systems using high fidelity models, which supports advanced experimentation. The operator can run an unlimited number of experiments on any given model bounded only by the computing power available to them. For instance, in 2017, the Government of Canada articulated the tasks it wants the military to conduct and how often it wants the military to conduct those tasks (Defence 2019). This resulted in many questions about whether the Canadian military has enough people and resource to accomplish their missions. Using the assumption that there are enough people and resources to conduct concurrent operations, the questions then expand into the area of strategic airlift and whether Canada could sustain the supply chains required for these missions.

Strategic airlift is defined as the ability to rapidly transport a large number of passengers and/or over-sized heavy cargo over long distances within Canada or between Canada and a theatre of operations (Works and Government Services Canada 2019). To explore the strategic airlift problem requires a generic model for transporting people and goods between locations and a flexible system to define and run experiments against the model. The main contributions of this paper are presenting a flexible operator friendly method for defining models and a generic model for strategic airlift. The result is a flexible and easy-to-use system which allows any operator with limited knowledge of M&S a way to run experiments using the strategic airlift model. We abstracted the model to only include aircraft and locations, as other components were too detailed for the problem. The cargo an aircraft carries is referenced as number of pallets and represents either goods or people. This allows for the model to be simplified as the cargo detail itself has already been studied in depth through load optimization and routing research. The model was implemented Cadmium (Vicino et al. 2019) and associated infrastructure was developed to progress the model through using microservices and REST APIs. These microservices were deployed using containers to facilitate rapid development and modifications of the model. In addition, we also introduce a flexible JSON logging format for DEVS simulators which enables flexible visualizations. This will be demonstrated through visualization of location data on a map. The strategic airlift model and simplified M&S process were combined together to look at a specific problem known as the *Arctic logistical support base*. Many experiments were conducted

to determine what is the best combination of aircraft to deliver pallets to a defined set of locations in the Canadian Arctic.

2   BACKGROUND

Our models were built using the Discrete Event Systems Specifications (DEVS) formalism (Zeigler et al. 2000). Formalisms as DEVS are not enough: models also require a process and system for operators to run experiments. One of the most challenging areas is the definition of a coherent process since it needs to be flexible enough to accommodate any system and rigid enough to allow for a standard. Extensive process flows such as the one in (Balci 2012) and others expanded and refined (Loper 2015) try and distil down the steps required to build and execute a model. Loper provides a clear consolidation of these processes by formulating a representative life cycle process that includes "establishing purpose and scope, formulating a conceptual model, acquiring and analyzing data, developing the simulation model and program, verifying and validating the model and simulation, designing experiments, executing the simulation and analyzing output, and configuration control" (Loper 2015). Whichever process is used to generate experiments usually requires background knowledge of M&S, specifically in the area of analyzing data, developing the model and program, and verifying the model and simulation. This reduces the ability for operators with limited knowledge of M&S to conduct experiments. Combining a number of previous research areas in XML and simulation as a service into the M&S process can result in increased adoption of M&S by operators.

Running experiments for an operator requires an easy to understand interface or format for defining experiments. This area has been previously explored and focuses on a way to allow the user to dynamically define a model. One example of an improved user interface is the CD++Builder (Bonaventura et al. 2010). This tool allows the user to drag and drop models in Eclipse and connect them. The benefits of this approach are an improved user experience for the operator; however, it is tied to a specific simulator and prevents re-used of models between different groups who use different simulators. UML has also been explored by eUDEVS (Risco-Martín et al. 2009) to evolve their original concept of XML DEVS definition into a graphical interface. This approach allows for improved interoperability, however, still requires the operator to know the complex intricacies of UML modeling. The earlier work by this group in defining XML atomic DEVS models (Martín et al. 2007) had potential, however, was not developed further. Producing atomic DEVS models from XML is challenging and their solution had a number of restrictions. We defined an XML format based on this work to define DEVS coupled models, which are the basis for experiments. XML was chosen due to its rigid nature compared to other open formats such as JSON.

Combing together experiments defined in XML, a simulator, and a visualization system for output requires a flexible and easy to understand infrastructure for the operators to interact with. Most of the research in this area is combined with Simulation as a Service. This has been explored recently by (Wainer and Wang 2017) in CloudRISE. At the time when CloudRISE was defined, containers were not yet a mature technology. When implementing our models, we extended this work by implementing a solution using REST APIs and containers. Containers offer a number of improvements over virtual machines especially when running single applications (Vaughan-Nichols 2015). This allows different interface to be rapidly developed and implemented for operators as their needs change vs a monolithic virtual machine system.

Military logistics is similar to supply chain in industry, however the factors driving the military are different. The supply chain field has been heavily researched across a variety of industries. (Terzi and Cavalieri 2004) provided an overview about simulation in supply chain, focusing on local versus network objectives which is common in industry, but different in the military where it is primarily self-contained in one organization. (Oliveira et al. 2019) looks at how industry assesses risk, which is quite different from the military. The industry application of supply chain typically focuses on reducing costs or mitigating risk, while the military typically focus on reducing time, prioritization of resources, or accomplishing a mission. (Xiong et al. 2017) highlight more of these challenges. This results in models which are based on typical supply chain problems; however, they need to be adapted to the unique needs of the military. We focus on using general principles from supply chain simulation in industry and apply in a military context.

A simple instance of the strategic airlift problem is the Arctic logistical support base problem. The Canadian military conducts operations in the Arctic and requires strategic airlift to bring people and resources to various locations to support missions. Typically, most cargo originates in Trenton, Ontario and then requires transport to an intermediate logistical support base in the north and then forwarded on to its final destination. The question then arises as to where this base should be located and how many aircraft are required to sustain the mission. This question has two potential solutions. One solution would focus on where to permanently establish a support base, where the other solution could be a temporary support base. The temporary support base would only be used for the duration of a mission. We focus on a temporary support base, as the permanent support base problem would require historical data beyond the scope of this work.

3 STRATEGIC AIRLIFT MODEL DESCRIPTION

The strategic airlift model was coarsely defined by design. This ensures the model can be applied to many instances of the strategic airlift problem. The design includes two DEVS atomic models, a location model, and an aircraft model. These atomic models can be combined together as shown in figure 1. Any number of locations can be defined and serviced by any number of aircraft. The primary output of the model is the amount of time required to transport a pre-defined number of pallets from point A to point B.
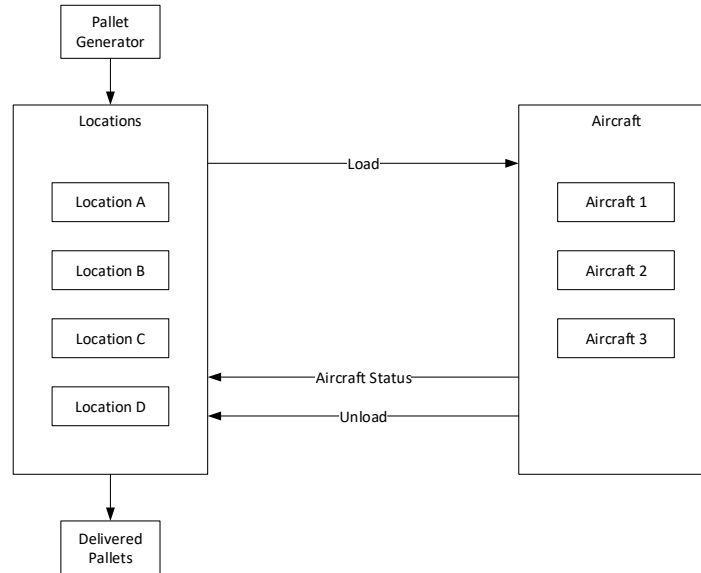

Figure 1: Strategic airlift coupled model.

The model begins by receiving a pallet at a given location. A pallet is used to represent a collection of people or goods that require transportation on an aircraft. A pallet is the smallest unit of representation in the model, and an aircraft can carry a fixed number of pallets together. Each location keeps track of which aircraft are available at that point in time. The aircraft then waits for a load to be ready for transport from the location. The location waits until it receives enough pallets to fill at least half the aircraft. Once a location has enough pallets, it groups them together and produces a load to be transported to another location and dispatches it to a waiting aircraft. The waiting aircraft then flies the load to a destination location and announces it has arrived. After the aircraft arrives, it unloads all pallets to the destination. At the destination location the pallets are either delivered if they are at the final location or added to the waiting pallet queue. Pallets wait in the queue until there is an aircraft available to transport them to the next location.

The following assumptions were made: pallets will each have a size of 1 and can be shipped with any other pallet; each location has infinite ability to store and process pallets; to reduce model complexity aircraft are serviceable and available at all times; aircraft fly directly between two locations. The airspeed for each aircraft can be reduced to account for non-straight line routing to avoid diplomatic airspace boundaries, and

account for other factors such as weather. Pallets are prioritized by pallet id and there is enough fuel to fly directly from any two points provided by the user in the scenario. The smallest granularity of time required by the model is every minute.

The model presented below has a number of beneficial features. The number of pallets each aircraft holds and how fast it travels can be configured per aircraft. Using the location and aircraft atomic model, the operator can create an experiment with unlimited numbers of each type. The routing can be changed during the simulation to account for weather and other external factors. Each aircraft starts at a home base and will transport loads of pallets to other locations as directed by each location. If there is no requirement to transport loads of pallets at a given location the aircraft will return to its home base after a defined period of time (default 1 day). Once the aircraft receives a load message from a location it will add the load to the aircraft and begin to fly. Every period (default 30 minutes) the aircraft will announce its position along with a status message until it reaches its destination. At the destination the aircraft will announce its arrival using an Aircraft Status message and unload the load to the location.

For each of the models in the application, we defined a DEVS atomic model, then we built a coupled model combining the components to achieve the objectives of the study, and then implemented each of them (and tested them thoroughly) using the DEVS Cadmium simulator. The following shows an example of one of the Atomic Models, which we used to define the behavior of the Aircraft.

Aircraft Atomic Model = <X, Y, S, ta, δext, δint, λ>

State Variables needed	Formal Specifications
Phase = INIT; Sigma = 0	X = { Load $\in$ <String, N, R, R, R, R, N, vector<Pallet>>}
Aircraft ID = Passed to constructor	
Type = Passed to constructor	Y = { Aircraft Status $\in$ <N, N, N, N, N, Time> | Load $\in$ <String, N, R, R, R, R, N, vector<Pallet>>}
Average Speed = Passed to constructor	
Home Location = Passed to constructor	S = {{Phase, Sigma, Aircraft ID, Type, Average Speed, Home Location, Max Load Size, Load, Current Latitude, Current Longitude, Distance Remaining, Total Distance}}
Max Load Size = Passed to constructor	
Load = {}	
Current Latitude = Current Longitude = 0	
Distance Remaining = Total Distance = 0	

δext (mbs, e, x <Pallet | Aircraft Status | Load | Location Information>) {
If x is Load and the load is for this aircraft
    Add x to aircraft Load
    Calculate flying route and set associated state
If phase = WAITING and Distance Remaining > 0
    Phase = FLYING; Sigma = 0;
Else If phase = FLYING
    Sigma = Update time period with e
Else If phase = WAITING Sigma = ∞ }

δint (e) {
case phase
    INIT:
        Phase = WAITING; Sigma = ∞;
    FLYING:
        If Distance Remaining < 0
            Phase = UNLOADING; Sigma = Unload Time (i.e. 30 min);
        Else
            Update current position; Sigma = Update Time (i.e. 30 min);
    UNLOADING:
        Phase = WAITING; Sigma = ∞ }

```
λ(s) {
If Phase = UNLOADING
    Send Aircraft Status message to destination location
    Send Load message to the destination location
If Phase = INIT
    Send Aircraft Status message to home location                                    }
```

There is an initialization phase before locations receive and process pallets. This phase is used to provide key information required for the model to run correctly. This includes setting up routes by processing Location Information Messages. Typically, these messages arrive at time 0, however the model is flexible, and routes can change dynamically throughout the experiment if required. The user would only need to change the time when Location Information Messages are sent. During the simulation if the model is unable to find a route between the current location and destination the pallet will be delivered with a next destination of -1 which denotes a failure to deliver the pallet.

After the initialization phase the location begins receiving pallets of people and goods to transport to a destination location. These pallets are stored in the waiting pallets queue. The pallets remain in the queue until an aircraft is available. Locations become aware of aircraft through Aircraft Status messages. When a location receives an Aircraft Status message, that aircraft is added to the waiting aircraft queue.

Load generation requires a waiting aircraft and at least enough pallets to fill half the aircraft capacity. Once these criteria are met the location retrieves the highest priority pallets which are defined by sequential pallet id, with the lowest being the most important. It then takes the first pallets by priority for the given destination location from the queue and generates a load. This load produces a Load messages which is sent to the waiting aircraft. The aircraft is then removed from the waiting aircraft queue.

When a location becomes aware an aircraft has arrived via the Aircraft Status message, it is unloaded. This is accomplished by the aircraft sending the location a Load message. The Load message contains all the pallets from the aircraft load. The location then proceeds to sort through the pallets and either delivers them if they are at the destination, or places them in the waiting pallets queue for forwarding onto the next location.

4   SUPPORTING FEATURES

Operationalizing the constructive simulation of the strategic airlift model requires the development of a number of key components. These components include: An input mechanism to define the experiment; a conversion mechanism to take the input and translate it to a language the simulator understands; a simulator to run the experiment; and a way to visualize the experiment.

The primary design criteria for these components are enabling constructive simulation of the strategic airlift model by developing a system which is easy for the operator to understand and use. Multiple designs were considered for each component and continually iterated until they met the following conditions:

- Usability - The system must be easy to understand and operate;
- Flexible - Each component needs to accommodate a wide variety of use cases which an operator would use the system for;
- Agile - The system must quickly adapt to changes, for example a new visualization is required and needs to be developed and incorporated into the system;
- Extensible - The system must allow for growth and the ability to add new feature; and
- Technology agnostic - The system must be flexible to not force the operator to use a specific technology or implementation.

The solution has 5 key components and the process starts by uploading a set of XML and data files. This process is iterative and the user can execute and visualize the simulation an unlimited number of times.

- Simulation Management - Management is required to upload, store, and manage the various models and experiments for the operator.
- Verification and Parsing - Verification of the input data and parsing is required to convert the input into a neutral format.
- Simulation Service - The neutral model format is transformed into simulator code and the experiment is executed.
- Visualization Service - The visualization service provides the results in an easy to digest format to the operator.
- Data Store - The entire process is supported by a data store for both files and information in a database.

The specific implementation in this paper is referred to as Sim Manager and has an easy-to-use web interface. Using this format allows the operator to extend or replace any component easily. For example if the operator prefers to use CD++ instead of Cadmium as their simulator, they can simply switch the simulation microservice. Additionally, the operator has the flexibility to switch the input format to something like JSON or add additional visualizations.

The technology used to implement the model and process was deliberately varied to demonstrate how anyone could add a component using any language or framework. The specific technology stack included both frameworks and different languages. The following frameworks were used in the development of the M&S solution: Spring Boot, D3, and jQuery. These were implemented through the use of Java, PHP, JavaScript, and C++. The system was specifically designed to be flexible to allow for growth using any language or technology stack. These technologies are tied together through containerized microservices using REST APIs. The benefits of this approach are listed below.

- REST API - REST APIs are a great fit for DEVS since they are stateless and complement the DEVS formalism. Everything that is required to complete the transaction is included in the web service call. Additionally, any data type can be sent and returned by REST APIs. JSON was specifically chosen as the data representation format since it is well understood by JavaScript and most modern web browsers. This makes implementation using frameworks such as D3 and jQuery seamless.

- Containers - Containers gained wide scale adoption when Docker introduced their implementation in the mid-2010's. This introduction was the catalyst to moving towards microservices and edge computing. These concepts break down historically large monolithic software programs into smaller atomic pieces and delivered as close to the user as possible. This paradigm is enabled by containers which are light weight virtualized constructs that are well suited for single purpose applications. Containers typically only contain the bare essentials for a program to run.

- Microservices - Microservices are used to break down each individual component of a software program into atomic pieces. These de-coupled pieces can operate independently allowing faster compilation and easier maintenance.

- Visualization - Visualizations are a key feature required to communicate large amounts of data easily to a human observer, to extract key pieces of knowledge.

4.1 JSON Log Format

An improved generic JSON log format was developed for Cadmium and any model can implement it. The JSON logging format consists of the following properties:

1. time - Denoting the current time for the log entry;

2. type - Denoting the type of log entry. Currently, this is state or message. Additional types can be added in the future;
3. name - Name of the model that produced the log entry; and
4. data - Any number of properties which the model wants to include in the log.

Examples of properties from the strategic airlift model included under the data property in the JSON logging format: class

- Type of model object. Useful to denote the difference between locations and aircraft, location
- This property has sub properties of lat/long. This is used to generate maps in visualizations, and
- The state of a given model.

The benefits of this JSON format is there are minimal required fields and the format remains flexible to any fields the model wants to include. Various visualizations and other tools can take advantage of this as required.

4.2 Visualizations

Two visualization components were implemented to show the agility of the JSON log format. The map component was also produced as an example of visualizing geographic data. Each component is known as a panel that can take up any dimension defined in their associated CSS class. A map panel displays any log entry with a data property of location. The map uses a class property of data to select an icon to show the operator (an operator can simply add a new icon for the associated class to the img folder of Sim Visualizer; for example, a ship could easily be added by having the model output a log entry with a class of ship, and an associated location property. If the operator includes an image of a ship in the img folder it will be displayed automatically). A map panel example is shown in figure 2.

The text panel in figure 2 is used to show how the operator can re-use the same component and use filtering to select which set of log data they want to display. When the text component is created, the operator can define the filter type to include the state, message, or both logs. They can also apply a display parameter that will select one of the data properties to display. In the text panel example, the left-hand side of the screen shows the state information from location, and the right shows the message output for each object.
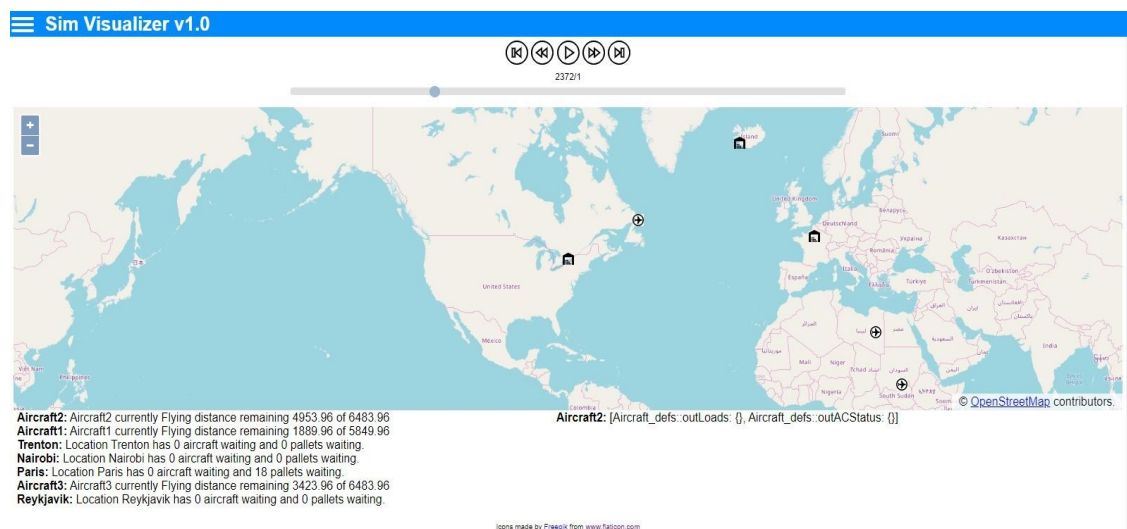

Figure 2: Sim visualizer.

5   CASE STUDY: THE ARTIC LOGISTICAL PROBLEM

The Arctic logistical problem described in section 2 is explored using the strategic airlift model in section 3. The location and aircraft atomic model were implemented using DEVS in the Cadmium simulator. All experiments were defined in a set of XMLs, executed using two microservices called SimManager and Sim Visualizer. A brief description of the problem being explored is: "What is the optimal location for a single intermediate logistics base in the Canadian Arctic?". This question is important since the Globemaster aircraft is not able to operate from all the same airfields that a Hercules aircraft can operate from.

5.1 Scenario Overview

Originally, four experiments were planned to determine the differences between no logistical support base and one logistic support base at Cold Lake, Alberta; Yellowknife, North West Territories; or Iqaluit, Nunavut. However, after review of the Cold Lake experiment results a different path was explored and resulted in six experiments being conducted. All scenarios have the follow set of parameters in common:

- All pallets will originate from Trenton, Ontario;

- Pallets will be flown by Globemaster aircraft to an intermediate logistical support base location and then to the final destination by a Hercules aircraft.

- The set of pallets will be the same for all scenarios. The pallets will be evenly distributed across each of the four final destinations, with each location having 48 pallets. The pallets are arranged in the same random order for each scenario.

- The four final destinations are: Alert, Nunavut; Hall Beach, Nunavut; Cambridge Bay, Nunavut; and Inuvik. This represents the geographic diversity of Canada's north and reduces bias for any single intermediate logistical support base location.

- Each Globemaster is configured to hold 16 pallets and fly at 490 knots. Each Hercules is configured to hold 4 pallets and fly at 300 knots.

An overview will be provided for each experiment in the next sections. After the overview is a sub section to discuss the results of each experiment.

5.2 Base Experiment

The goal of this experiment is setting a base line value for how long it would take three Hercules aircraft to resupply all four final destinations listed in the scenario overview. The scenario concludes when all aircraft return to their home base. The baseline parameters in the scenario overview are used for this experiment. The baseline scenario requires 25 days, 14 hours, and 24 minutes to transport all the pallets from Trenton to the four final locations. Table 1 contains the distances between Trenton and the four final destinations. Pallets could be delivered to locations faster using either an intermediate logistical support base or adding Globemaster aircraft instead of Hercules aircraft due to their larger carrying capacity.

Community	Distance (km)
Alert	4299
Hall Beach	2747
Cambridge Bay	3187
Inuvik	4156

Table 1: Distance from Trenton, Ontario.

5.3 Cold Lake Experiment

Using the base experiment as a comparison, this experiment explores the effects of having an intermediate logistical support base in Cold Lake, Alberta. The hypothesis is that this would reduce the time required to transport all pallets to their final destinations. One Globemaster is used to transport pallets from Trenton to Cold Lake, and three Hercules are used to transport the pallets from Cold Lake to the four final destinations specified in the scenario overview. We use the baseline parameters in the scenario overview.

Location	Trenton (km)	Cold Lake (km)
Alert	4299	3438
Hall Beach	2747	2166
Cambridge Bay	3187	1649
Inuvik	4156	1962
Trenton	0	2599

Table 2: Distance between locations.

Aircraft	Base (km)	Cold Lake v1 (km)
Globemaster	0	62376
Hercules	345336	236754
Total	345336	299130

Table 3: Number of flights.

A first scenario for Cold Lake requires 23 days, 14 hours, and 11 minutes to transport all the pallets from Trenton to the four final locations. This could lead the operator to conclude the addition of an intermediate logistical base in Cold Lake does not have a large effect on the scenario time. With the effort required to set up an intermediate logistical base, it may not be worth the return on investment. The Globemaster aircraft finished transporting pallets from Trenton at 14 days, 20 hours, and 38 minutes. This represents a savings in time, and also a savings on the distance flown, which translates to a reduction in fuel and maintenance. Table 2 shows the distances between locations. This information is used to calculate the distance flown by each aircraft during this experiment. Table 3 shows the distance flown by each type of aircraft, compared between the base experiment and this experiment. The difference is 46 206 km.

Further experimentation could be conducted to optimize the location that would result in the fewest miles flown, however other options exist to reduce the scenario completion time. The next experiment looks at the effects of adding more aircraft to the scenario. This hypothesis was developed after reviewing the logs to determine how much time aircraft spent waiting for pallets. The logs showed that almost half the time there was at least 1 aircraft waiting for pallets in Cold Lake. The waiting time issue could increase if aircraft immediately flew back after delivering their load to the final destination. Currently, all the Hercules aircraft in the experiment wait for 24 hours at the final destination prior to returning to their home base.

The previous experiment demonstrated that there is a bottleneck from Trenton to Cold Lake. Looking at the effects of adding a second Globemaster to transport pallets from Trenton to Cold Lake we can see that this new scenario requires 22 days, 14 hours, and 3 minutes to transport all the pallets from Trenton to the four final locations. Compared to the previous version, this improves the scenario time. However, a decrease in waiting time was observed which leads to a decrease in scenario time since aircraft are not waiting for pallets. We can conclude that the supply of pallets has a large effect on the overall scenario completion time. Therefore, if we want to minimize scenario completion time, more aircraft are required to transport pallets to an intermediate logistical support base. Additional observations include the total time required to transport pallets from Trenton to Cold Lake, which, in this experiment was 6 days, 20 hours, and 5 minutes. Compared with 14 days, 20 hours, and 38 minutes from version 1 of the experiment, this represents over a 50 percent reduction in time. One solution to reducing scenario time would be the aggressive dedication of resources early on to transport goods to the intermediate logistical support location.
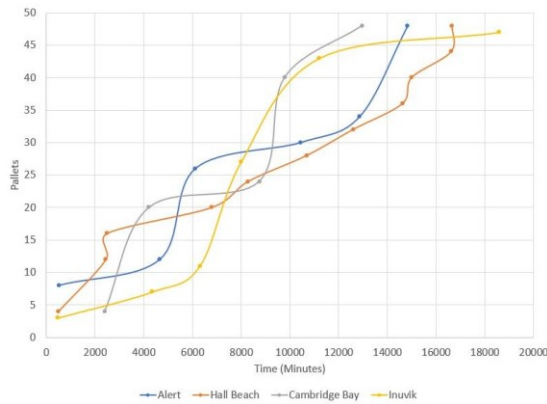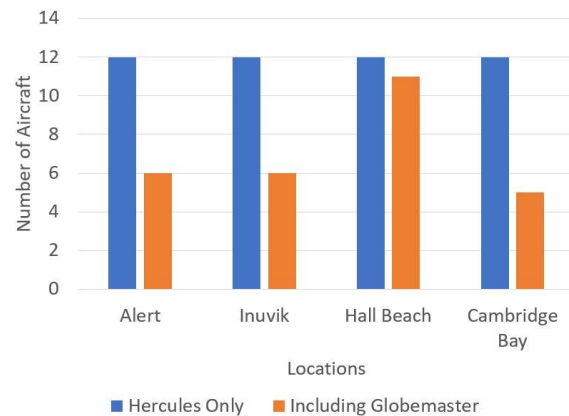
Figure 3: Delivery time vs pallets.

Figure 4: Location vs number of aircraft.

Our next experiment explores what happens if a Globemaster is positioned with two Hercules aircraft in Cold Lake instead of the three Hercules. This should hopefully result in quicker pallet distribution to the final destinations. The only change for this experiment from version 2 will be the swap of one Hercules aircraft for one Globemaster in Cold Lake. This scenario requires 14 days, 2 hours, and 16 minutes to transport all the pallets, a significant reduction in the overall scenario time compared to the base experiment.

Figure 3 shows a graph of pallets delivered by time for each of the final destinations. The use of the Globemaster was relatively equal between all the locations and allowed each location to reduce the number of flights required. Figure 4 shows the number of aircraft required to transport all goods to each location. One Globemaster significantly reduces the requirement for multiple Hercules aircraft.

5.4 Yellowknife

Version 1 of the Cold Lake experiment demonstrated reduction in flying time by using an intermediate logistical support base. This next experiment is a continuation where Yellowknife is used instead of Cold Lake. This location is further from Trenton; however, it moves closer to the center of the four final destinations. This experiment will use all the same parameters as Cold Lake version 3, except switching the location from Cold Lake to Yellowknife. The Yellowknife scenario requires 12 days, 16 hours, and 56 minutes (18 296 minutes) to transport all the pallets from Trenton to the four final locations using two Globemaster aircraft from Trenton to Yellowknife and one Globemaster and two Hercules aircraft to the four final destinations. This represents a further reduction in scenario time from the other scenarios.

5.5 Cambridge Bay

The final experiment is a follow on to the previous experiment where Cambridge Bay is used as the intermediate logistical support base and Yellowknife becomes one of the final destinations. This scenario requires 12 days, 14 hours, and 50 minutes to transport all the pallets from Trenton to the four final locations using two Globemaster aircraft from Trenton to Cambridge Bay and one Globemaster and two Hercules aircraft to the four final destinations. This represents the lowest scenario time for all the experiments.

As shown in table 4, the total distance flown and number of flights for this location is significantly less than previous experiments. However, with the distance from Trenton and addition of the Globemaster to deliver to the final destination it has increased the percent of time aircraft spend waiting. A review of the logs indicates all pallets were in Cambridge Bay at 7 days, 3 hours, 4 minutes.

Location	Number of Trips	Total Distance (km)
Alert	6	21540
Hall Beach	8	15136
Yellowknife	7	11928
Inuvik	7	16058
Trenton	15	95610
Total	43	160272

Table 4: Cambridge Bay statistics.

Based on these results, Cambridge Bay would be a good choice for an intermediate logistical support base due to its central location in the Arctic. Additional experimentation could be conducted to determine the correct mix of aircraft for re-supply to minimize the time spent waiting on the ground.

5.6 Model Conclusions

Results from experimentation indicate a number of key areas which could be optimized for when selecting an intermediate logistical support base. These include the following:

1. A location should be chosen that is geographically center between the final destinations, but not too far from the source location;

2. This model could be turned into an optimization problem where the objective would be to fly the minimum distance to deliver all the pallets. Minimum distance would translate into a reduction in fuel, maintenance, and number of aircraft and result in the lowest cost option.

3. Having a robust link to the source location is critical to move pallets from the source to intermediate location as quick as possible. This minimizes the time other aircraft spend waiting for pallets.

A good solution for this specific instance is to put an intermediate logistical supply base in Cambridge Bay, Nunavut and to use two Hercules and one Globemaster for resupply from Cambridge Bay to the final destination. At least two Globemaster aircraft would be required to maintain the flow of pallets from Trenton to Cambridge Bay. A modification to this model could include having aircraft transition between different routes. In this instance it would allow the scenario to start with two Globemasters aircraft from Trenton to quickly supply Cambridge Bay and then have one Globemaster be re-tasked to a different mission and one Globemaster continue with two Hercules to supply the final destinations.

6   CONCLUSIONS

Operationalizing constructive simulation for problems such as strategic airlift, allow the M&S community to further engage operators. This engagement is enabled by tools such as the one presented in this paper, which enables the rapid definition and implementation of experiments for operators with limited knowledge and experience in M&S. The Arctic logistical support base problem was used as an example to demonstrate the power of Sim Manager. Non-technical operators could easily use the XML format we defined to construct new experiments and simulate them. The flexible JSON logging format we defined allows for the rapid exchange of data between simulation and visualization systems. Communicating output to the user through maps, images, and text allows users to quickly convert simulation results into useful knowledge. These visualizations help bridge the gap for the operator between the technical and business world. Sim Manager and well thought through models such as strategic airlift, have few gaps for operators to become involved in M&S. Future work in this area would include taking the concepts presented in this paper and conducting further user studies with operators to gain empirical data to quantify the difference in using this method compared to traditional tools. Using the tools, frameworks, and models presented in this paper an operator can easily extend these to other problem domains such as search and rescue, sea transport, and even non-military domains such as commercial supply chain.

REFERENCES

Balci, O. 2012. "A life cycle for modeling and simulation". *SIMULATION* vol. 88 (7), pp. 870–883.

Bonaventura, M., G. A. Wainer, and R. Castro. 2010. "Advanced IDE for Modeling and Simulation of Discrete Event Systems". In *Spring Simulation Multiconference*, SpringSim'10. San Diego, CA, USA.

Defence, National 2019, Nov. "Strong, Secure, Engaged: Canada's Defence Policy".

Loper, M. L. 2015. *The Modeling and Simulation Life Cycle Process*, pp. 17–27. London, Springer London.

Martín, J. L. R., S. Mittal, M. A. López-Peña, and J. M. de la Cruz. 2007. "A W3C XML Schema for DEVS Scenarios". *2007 Spring Simulation Multiconference*, pp. 279–286. San Diego, CA, USA.

Oliveira, J., M. Jin, R. Lima, J. Kobza, and J. Montevechi. 2019. "The role of simulation and optimization methods in supply chain risk management: Performance and review standpoints". *Simulation Modelling Practice and Theory* vol. 92, pp. 17 – 44.

Risco-Martín, J. L., J. M. de la Cruz, S. Mittal, and B. P. Zeigler. 2009. "eUDEVS: Executable UML with DEVS Theory of Modeling and Simulation". *SIMULATION* vol. 85 (11-12), pp. 750–777.

Terzi, S., and S. Cavalieri. 2004. "Simulation in the supply chain context: a survey". *Computers in Industry* vol. 53 (1), pp. 3 – 16.

Vaughan-Nichols, Steven 2015, Apr 28. "Containers vs. virtual machines: How to tell which is the right choice for your enterprise". Computerworld, Inc. Apr 28, 2015; Last updated - 2015-04-29.

D. Vicino, L. Belolli, C. Ruiz-Martin, G. Wainer. "Building DEVS Models with the Cadmium Tool". Proceedings of Winter Simulation Conference. National Harbor. MD. 2019.

Wainer, G., and S. Wang. 2017, 05. "MAMS: Mashup Architecture with Modeling and Simulation as a Service". *Journal of Computational Science* vol. 21, pp. 113–131.

Public Works and Government Services Canada, Translation Bureau 2019, Nov. "strategic airlift [1 record] - TERMIUM Plus ® Search - TERMIUM Plus® ".

Xiong, B., B. Li, R. Fan, Q. Zhou, and L. Wu. 2017. "Modeling and Simulation for Effectiveness Evaluation of Dynamic Discrete Military Supply Chain Networks". *Complexity* vol. 2017, pp. 9.

Zeigler, B. P., T. G. Kim, and H. Praehofer. 2000. *Theory of Modeling and Simulation*. 2nd ed. Orlando, FL, USA, Academic Press, Inc.

AUTHOR BIOGRAPHIES

ROB BARWELL is currently working on his M.A. Sc in Electrical and Computer Engineering at Carleton University in Ottawa, ON, Canada. His email address is robbarwell@cmail.carleton.ca.

GABRIEL A. WAINER is now Full Professor at the Department of Systems and Computer Engineering at Carleton University. He is a Fellow of SCS. His e-mail and web addresses are gwainer@sce.carleton.ca and www.sce.carleton.ca/faculty/wainer.