A RECONFIGURABLE AGENT-BASED DISCRETE EVENT SIMULATOR FOR HELICOPTER AIRCREW TRAINING

Vivian Nguyen
Mina Shokr
Ana Novak
Terry Caelli

Defence Science and Technology Group
506 Lorimer St, Fishermans Bend
VIC 3207 Australia

ABSTRACT

In this paper we consider a dynamic framework for an agile and adaptive simulator capable of robust performance under change without requiring a complete rewrite. In particular, the framework extends the current hybrid modelling paradigm which integrates agent based (AB) constraints and controls with a discrete event simulation (DES) methodology to allow for a more expressive, authentic representation of both process flows and agent policies. This research originated from the need to develop an aircrew training pipeline simulation for Australian Defence Force that can provide what-if analysis functionality, to improve the efficiency and operational effectiveness in a domain that is continually undergoing aviation training consolidation and rationalization. We discuss the challenges in selecting the appropriate framework and present a prototype implementation that handles the unavoidable evolving changes in infrastructure, human and material resources, policies and flows that frequent this domain.

1 INTRODUCTION

The Australian Defence Force helicopter aircrew training continuum[1] (HATC) is currently undergoing a major transformation where basic helicopter flight training components across multiple services are being consolidated into a single location. The rationale for this transformation is that the training continuum will be more efficient and robust. In the next few years there is a plan to phase out the current Blackhawk helicopter along with its associated resources that will likely be transitioned to a multi-role helicopter (MRH90) squadron. Changes like those mentioned above are common and occur cyclically, illustrating that the system is in a constant state of transience.

However, infrastructure changes are not the only changes that evolve in time. The policies that govern training in individual schools and squadrons are also susceptible to change, as well as rules about criteria for passing individual training components. The changes in platforms inevitably cause perturbations in the flow of students through the training program. Further, school pass and failure rates can severely impact on the predictability of the graduates to the operational squadrons, especially if the associated variance is very high.

All these characteristics demonstrate that HATC is a complex system (Magee and de Weck 2004, Sterman 1992) with many interdependent components, multiple feedback loops and unpredictable failure rates. To further exacerbate the complexity of this dynamic system, the infrastructure including squadrons, schools, courses, resources and policies that surround them, inevitably change over time. For example, schools may merge to achieve great efficiencies, courses may be relocated from one school to

[1] A training continuum encapsulates individual and collective training activities that together prepare students, undertaking the training, for deployment to operational squadrons.

another, squadrons may be closed, new platforms acquired and old platforms retired, etc. This illustrates the transient nature of the system.

This paper discusses a framework designed to address the need of Australian Defence Force to perform what-if analysis, to improve the efficiency and operational effectiveness of this complex, transient HATC system. To model such changes, while preserving a common simulation architecture, what is required is an easily reconfigurable simulation framework capable of explicitly representing the changing infrastructure, domain policies, constraints and flows. This would result in decreased effort in re-building models, provide a repository of historical changes and, finally, cost benefits.

In recent years a number of simulation software (such as AnyLogic[2]) have emerged that are capable of handling multi-modelling paradigms including Agent-Based (AB), Discrete Event Simulation (DES) and System Dynamics (SD). In this paper we focus on a newly emerging hybrid AB-DES approach to build a simulation framework that addresses the needs of helicopter aircrew training.

2 RELATED LITERATURE

In SD models, entities are viewed as "fluid like" quantities with continuous state changes and do not permit tracking of individuals through the system. In contrast, DES models a system in terms of queues and events where the entities are individuals with distinct characteristics and state changes occur at discrete points of time (Brailsford and Hilton 2001). Therefore, modelling an ordered sequence of events, such as training courses, is suited more naturally to the DES rather than the SD paradigm. Swinerd and McNaught (2012) agree that depending on the problem being modelled, one paradigm may be a more natural choice than another.

DES has been used to simulate a training continuum at the Marine Corps Communication-Electronics School (Davenport et al. 2007). Simulation modelling was also used to study the operations of 1 and 2 Canadian Forces Flying Training Schools in order to gain insights on delays and resource bottlenecks to assist with resource allocation and planning at the schools (Séguin 2011, Séguin and Hunter 2013). In these studies, resources such as aircraft, instructors or classrooms were all modelled as static resources. This approach does not easily accommodate changes in infrastructure, resources, policies and flows over-time. As opposed to the static nature of entities in DES models (Borshchev and Filippov 2004), agent-based models of schools, instructors and students in our study are defined as independent entities making decisions about their actions and interactions based on a set of rules governing their behaviour. Additionally, agent based models permit individual student tracking (Bonabeau 2002).

The modelling and simulation literature already reports on simulations that combine different modelling techniques. Heath et al. (2011) presented an interesting discussion on cross-paradigm simulation modelling by comparing pairs of different simulation paradigms (e.g., SD-DES, DES-AB, SD-AB). They argue that a DES-AB approach is suited to modelling any situation which includes resources that must perform activities as well as human interactions where individual behaviours determine how these activities progress, as is the case in our problem with the aircrew training continuum.

SD and AB have been used together to simulate healthcare systems, where the SD model provided high abstraction levels of a large scale system while individual workflows of person were modelled as agents (Djanatliev and German 2013a). However, modelling lower level details requires ordered sequences of individual events, necessitating the inclusion of a DES, where agents dynamically generated from SD and then using DES process-oriented models for individual hospital operations (Djanatliev and German 2013b).

DES and AB have been used together to simulate the dynamic allocation of human resources to affect a production process (Montevechi et al. 2015). Here, the inclusion of AB was justified to expand the "computational ability to mimic" natural human behaviour (pg.1570). It was also argued that the

[2] "AnyLogic - Multimethod Simulation Software." Accessed 22 Mar 2016. http://www.anylogic.com.

interaction of independent agents “by definition cannot be reduced to parts of the system because of the interaction among them” (pg.1562).

Anagnostou, Nouman and Taylor (Fakhimi et al. 2014, Anagnostou, Nouman, and Taylor 2013) developed an distributed AB-DES based simulation framework to support strategic and operational decision making in emergency medical services. The framework consists of individual federated models, Ambulance (AB based) and Accident & Emergency (DES based), the communication between which was handled by operations and data exchange middleware.

Another demonstration of a multi-paradigm reconfigurable AB-DES is the recent work of Liraviasl et al. (2015) for manufacturing. Their system aimed to rapidly adapt factory configurations to meet the needs of changing customer demands, new technologies, and continuing needs for efficient manufacturing under change. Their multi-paradigm model allows for on-line layout changes by reconfigurable logic controls consisting of both agent state and agent-agent coupling updates. These control different parts of the manufacturing system so that they can be updated as a function of requirements. For example, the reallocation of an idle machine to different objects, tasks or order changes requiring relocation of resources. This adds an additional on-line reconfigurable processes to the type of system developed by Anagnostou, Nouman and Taylor (ibid.).

3 CONCEPTUAL DESIGN

The HATC provides training of helicopter aircrew across Army and Navy and partially Airforce. Students, upon entering the continuum, have designated paths depending on their service as well as their types (pilot, observer or aircrewman). A Navy pilot would train at different schools, undertaking different courses to that of an Army air crewman. Factors that affect student progression through the continuum, which often significantly delay training completion, include instructor and resource availability (including aircraft and simulators), school policies, graduate targets at each school, as well as changes to operational squadron requirements. Pass rates at each school also influence student flows.

3.1 Training continuum components and their relationships

The Entity-Relationship diagram in Figure 1 shows the relationships between different components within the training continuum. A *School* contains one or more *Courses*. It also has a number of *Aircraft* and *Simulators* which are used for training during a Course. A *School* also has a number of *Instructors* that teach one or more *Courses* at the *School*. *Aircraft*, *Simulators* and *Instructors* together are *Resources* required for running a *Course*.

Students, depending on their service and type, undertake different sets of *Courses*, at completion of which they become *Trained Personnel* and are posted to respective operational *Squadrons*. After a fixed amount of time in the *Squadron* they may undertake training to become *Instructors*. If so, they convert to be *Students* again and undertake specific instructor *Courses*. Upon completion they become *Instructors* and start instructing at *Schools*.

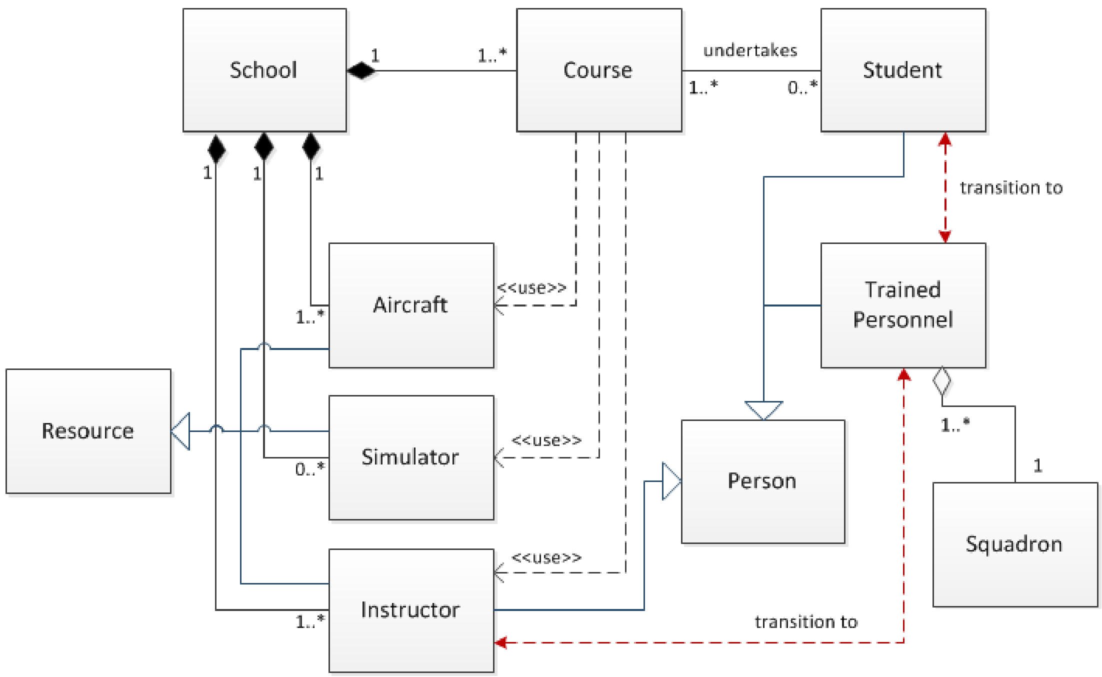

Figure 1: HATC Entity-Relationship diagram[3] (see Section 3.1 for details).
Note how some graduates take over Instructor roles.

3.2 Design considerations

Capturing the transient nature of the system is often desirable, but practically rarely achievable since it typically involves enumerating all possible system states and then developing procedures for determining what particular states, state changes, best apply to given situations and processes. This is difficult to achieve, a priori. In the case of the HATC, the possible changes that are anticipated include:

- Adding, moving or removing a squadron from the HATC network
- Adding, moving or removing a school from the HATC network
- Adding, moving or removing a course from any school in the HATC network
- Adding, moving or removing resources from schools and squadrons (i.e. aircraft, simulators, instructors etc.)
- Policy changes such as priority servicing, pass/fail rates and their variance etc.

[3] "An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system". Accessed 07 Jun 2016. http://searchcrm.techtarget.com/definition/entity-relationship-diagram.

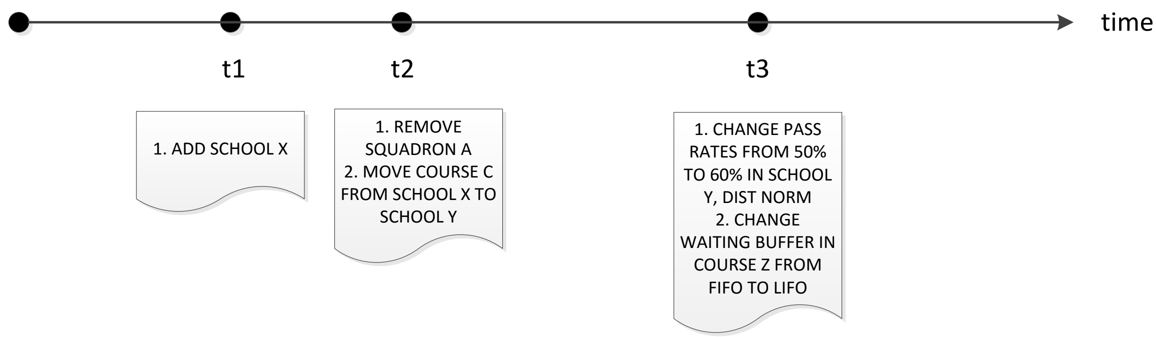

Figure 2: Possible changes to the training continuum over time.

Figure 2 illustrates the types of changes that can occur that need to be incorporated into the reconfiguration process. This also provides a history of the system's evolution over time. The anticipated evolutionary changes can be grouped as follows.

Reconfigurable structure and flows. The architecture of the Defence training infrastructure is in a constant state of evolution, always seeking to improve current efficiencies, capitalize on common components of training and minimizing delays. In order to achieve this, the simulation needs to be able to easily create new schools, merge two schools, remove old ones or move them to a different geographical location – taking into account all the consequences and influences it may have on the remainder of the training continuum and internal resourcing. Similarly, with squadrons, which can be shut down if a platform is phased out or modified to support a replacement.

On a lower level, courses being nested within the schools are also subjected to changes including creating new courses and removing or replacing old ones. The critical consideration in modelling these changes is to decouple all the nested components. This allows for greater flexibility. For example, simulating the merger of two schools requires only a move of an appropriate course into a single school.

All changes involving schools and courses will have impact on both student and instructor flows and therefore the model needs to be able to both dynamically reconstruct the paths as well as allow the flexibility to manually override if required.

Local school/squadron policies. Each school is an entity on its own with its unique set of policies, that represent the views of the services it belongs to as well as the current Commanding Officer (CO). For example, the number of students waiting for a course may exceed the school's capacity to train them due to the lack of resources. Which students get to undertake the course would totally depend on the school's policy, such as the longest waiting students get a seat on the course (First In First Out), or Airforce students have higher priority than students from other services, or students closest to graduation are offered a position on the course first.

Instructor feedback loop. Unlike typical production or manufacturing processes where a product, once created exits the system, graduates in our case do not always exit. Some of them eventually become instructors and therefore continue to play a role in the training continuum. A school's capacity to train students depends on the available number of instructors. The number of instructors supplied to schools depends on the number of graduates in the previous years. This illustrates the feedback loop that is difficult to manage due to latency of the pipeline and unpredictability of graduates numbers.

3.3 Hybrid AB and DES Conceptual Model

The conceptual model of the HATC utilizes both agent-based and discrete event simulation approaches and takes advantage of the flexible coupling options offered by the AnyLogic simulation software.

The model's architecture is illustrated in Figure 3. It is clear that students, instructors and other trained workforce have their own specific behaviours and therefore should be modelled as autonomous agents (*Person agents*). Schools and squadrons have local policies that govern their decision making, and therefore should also be modelled as agents (*School* and *Squadron agents*). On the other hand, processes involved in running a course require discrete event simulation in order to faithfully model the sequence of activities and depict bottlenecks and queuing at the resource level. As such, courses are modelled as *Course DES process flows*. Courses are run under the umbrella of their respective school policies, whilst utilizing schools' resources. Therefore, the *Course DES process flows* are embedded inside *School agents*. Students (a subset of *Person agent* population) that undertake a training course go through a specific *Course DES process flow*, during which they are simply treated as homogeneous entities. Various interactions take place between agents of different types. Squadron agents set the constraints for school intakes, a downstream school also influences the upstream school's intake, students get posted to squadrons or schools upon graduation and more.

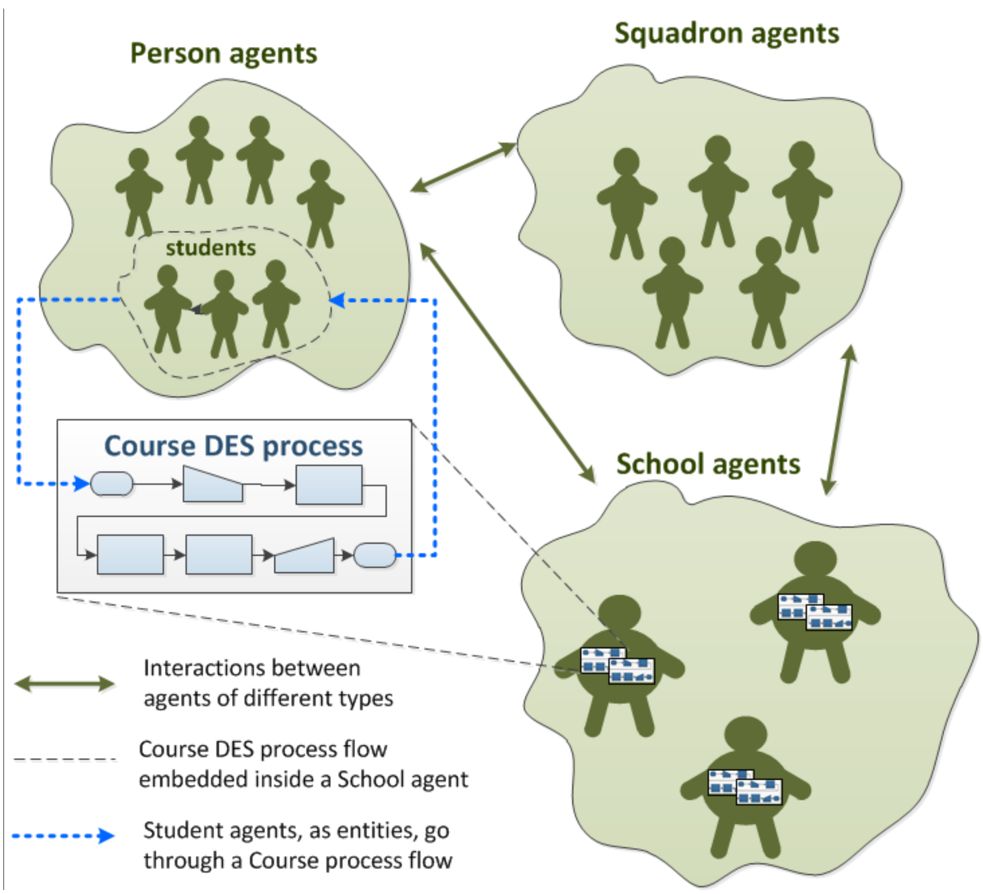

Figure 3: The HATC AB-DES model architecture

Person Agent. All human entities within the continuum are modelled as a Person agent. Modelling students, instructors and trained personnel as a single Person agent adds to the realism of the model. A Person agent has three main states, each representing a stage in their career – 'Student', 'TrainedPersonnel' and 'Instructor' (Figure 4 – Person agent box). Nested state charts within these states are used to exhibit individual dynamics of a person's day-to-day business. For example, a student is currently on a course or waiting for the next course; an instructor currently busy teaching a course or waiting to instruct, etc. The Person agent only exits the system upon retirement or separation from military service.

School Agent with DES elements. A school is modelled as an agent. This approach allows for using a state chart to mimic a school's CO's decision making process regarding the day-to-day running of the school as well as the schools policies. A school has a number of courses (Figure 4 – line A) and maintains a collection of students (Person agent in 'Student' state, Figure 4 – line B) that are enrolled in its courses.

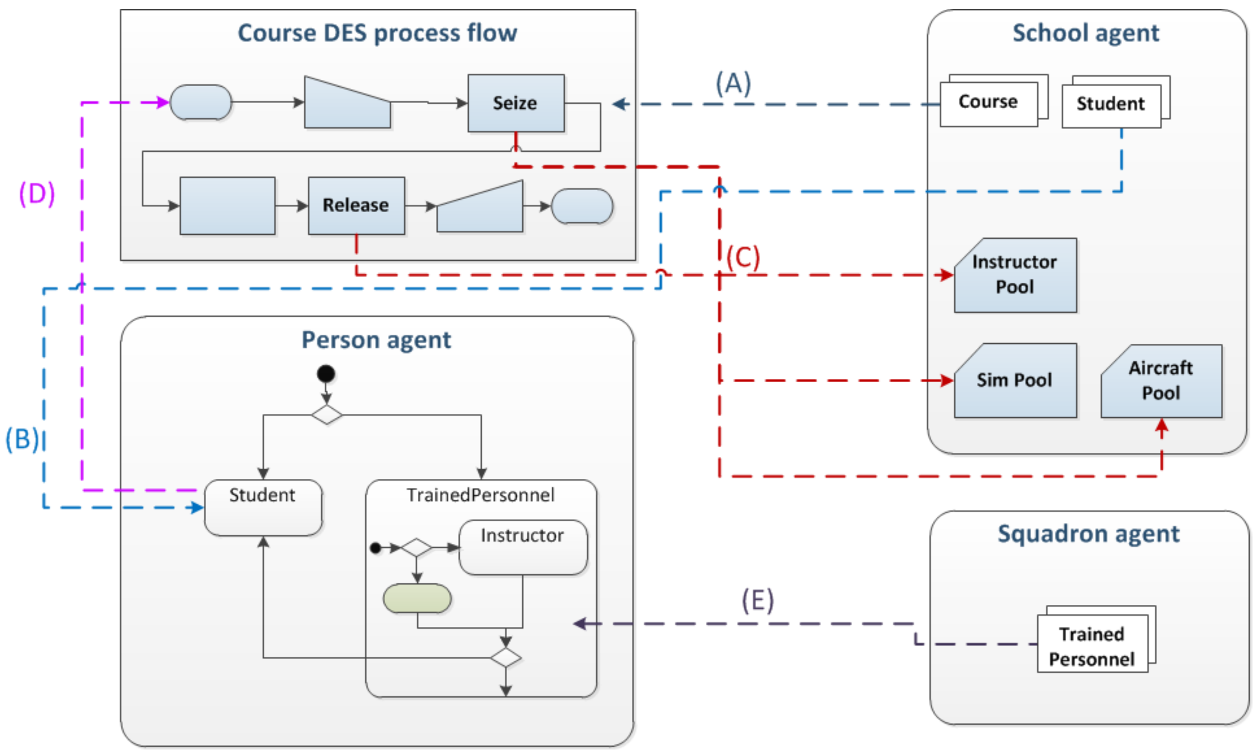

Figure 4: HATC components modelled as either agents or a DES process, their relationships and inter-dependencies.

Course process model. A Course is entirely modelled as a DES process flow and sits inside a School agent. A course is considered as a series of DES activities, which seize a predetermined set of resources from the hosting school's resource pools (Figure 4 – line C) and have a specified delay time. During the running of a course, students temporarily act as homogeneous entities in the process (line D). After completion of the course students continue to exist as agents and commence another course or become trained personnel.

Aircraft, Simulator and Instructor Resources. The school's resources are modelled as DES Resource Pools. Being DES Resources, the aircraft and simulators will have their work and maintenance schedules as well as random failures. Instructors will have work schedules as well as random sick and holiday leave.

Squadron Agent. A squadron is modelled as an agent to represent its CO, who determines the squadron's required workforce level (Person agents in 'TrainedPersonnel' state, Figure 4 – line E), which dictates the student intake into the continuum.

The proposed model architecture reflects the natural structure of the HATC. This object-oriented, multi-method simulation conceptual design allows for the HATC model to be reconfigurable without interrupting execution of the simulation (i.e. runtime reconfigurable) A new school, squadron or course object can be dynamically created and student training paths or personnel career paths can be redefined accordingly.

This design also enables easy implementation of the instructor feedback loop. A student, once graduated, joins a squadron's workforce. The Squadron agent periodically decides how many of its workforce to send back into training continuum to train and become instructors. The selected Person agents will transition from 'TrainedPersonnel' state to 'Student' state and then to 'Instructor' state. An instructor's movements through the continuum are prescribed by their career path depending on their service and type.

4 MODEL IMPLEMENTATION

A prototype model was developed in AnyLogic. The implementation of the proposed conceptual design was seamless and all concepts were successfully translated into the actual model, with one exception. Once a Person agent transitions into the 'Instructor' state, by design it should become a resource unit and is managed by a DES Resource Pool. However, it was not possible to implement this in AnyLogic. The software supports addition of a specified resource to a pool. Unfortunately, it does not (at the time of writing) support targeted removal of resource units. This is an important aspect of HATC – instructors do not stay in the same school, they move to different schools at each posting. Therefore, handling of instructors was carried out by code, which was easy enough given AnyLogic's underlying Java framework.

4.1 Model input

Model parameters are read from an input file at launch. As shown in Table 1, these include parameters required to create and initialize all courses, schools and squadrons, as well as their initial resource and workforce levels. Student training paths and trained personnel career paths are also read in at model initialization. ,

Table 1: Model external input parameters.

Object	Parameters
School	name, courses, the number of resources of each type
Squadron	name, workforce level
Course	ID, min and max number of students, required number of resources of each type, pass rate, student to instructor ratio
Student paths	A sequences of courses in a student's curriculum for the student service and type
Career paths	A sequences of posting locations (school and squadrons) for trained personnel
Course schedules	Start and end time for courses.

4.2 Runtime Reconfiguration

The aircrew training pipeline simulation is runtime reconfigurable by means of parameterised variables representing the state of components and flows in the model. Table 1 introduced parameters used in the simulation. These parameters can be changed via the user interface at runtime, that is, without interrupting the execution of the simulation or having to recompile the code. The parameterised variables are regularly polled for changes in value and conditional branches of execution followed depending on the value.

For example, reconfiguring the continuum during runtime to add a new course to a student pathway, would be achieved via the user defining the properties of the new course. This new course would be instantiated on the next polling event and would then form part of a student's pathway.

4.3 Communication and Logic Flows between agents and process models

School – Course. A school controls the schedules of all its courses and triggers the start of a course by sending a trigger message to students enrolled in the course, invoking a state change for the Person agent (to the nested 'OnCourse' state within 'Student' state).

Person – Course. When a Person's state is changed to 'OnCourse', it injects its token entity into the process model. On exit from the course process, the entity is returned to the agent, triggering another state change (to 'WaitingForACourse' or 'TrainedPersonnel').

Squadron – School and School – School. Squadrons and downstream schools control the intake of students at upstream schools, specifically, the maximum number of students to enrol into a course at the school. This is done by directly manipulating this variable.

Person – School/Squadron. A person on entering 'Instructor' or 'TrainedPersonnel' state checks for its next posting destination and injects itself into the appropriate School/Squadron collection of instructors or workforce accordingly.

5 CASE STUDY

This section presents a fictitious scenario designed to demonstrate the model's ability to reconfigure itself at runtime to address the required changes to the training continuum.

In this scenario, platform A starts to phase out two years into simulation (2018) as does all the associated training. The phasing out is scheduled to be completed within two years. More specifically, the following events occur during the phasing out period:

1. Gradually move current instructors at School A to School B.
2. Divert incoming instructors to other schools and squadrons.
3. Gradually divert current and incoming students from School A to School B.
4. Gradually reduce the number of occurrences of training courses at School A.
5. Gradually increase the number of occurrences of training courses at School B.
6. At the conclusion of the phase-out period, increase the number of aircraft and simulators at School B.

Here, the expected behaviour for School A would be to observe the gradual reduction in the number of instructors and students, until they reach zero by 2020. This is confirmed by the model output results as shown in Figure 6.

Initially, only the number of students and instructors were increased in School B, until approximately mid-2020. During this time a steady increase in the number of students was observed, indicating formation of the queue at this transition period (Figure 7). However, as soon as the accompanying aircraft and simulators were introduced at School B in early 2020, the number of students exhibited a downward trend. This showed that increasing instructors and course offerings alone was not sufficient to meet increasing student intake, but that an increase in other material resources was also required. An intake of students may occur in close proximity to a graduation, hence the peaks and troughs exhibited in Figures 6 & 7.

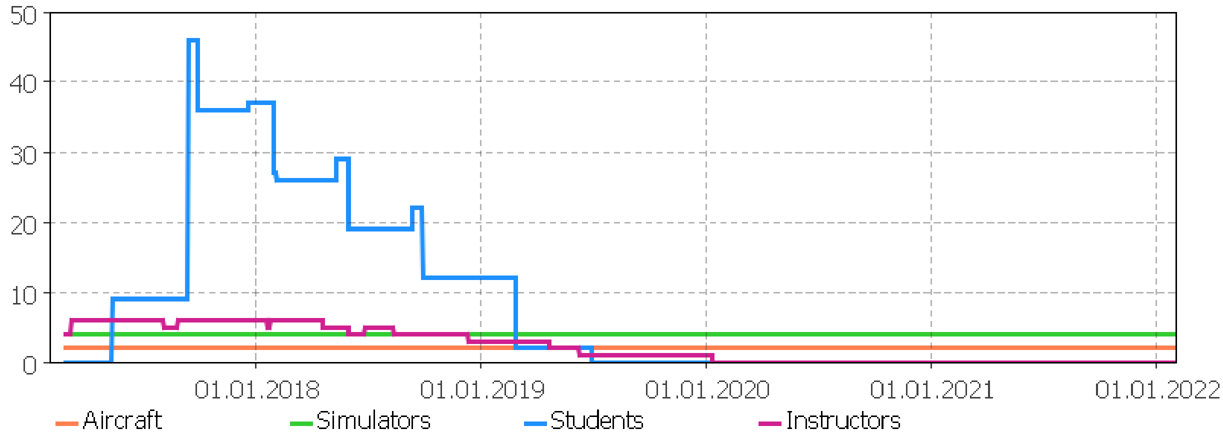

Figure 6: Aircraft, simulator, student and instructor count at School A over the transition period and beyond.

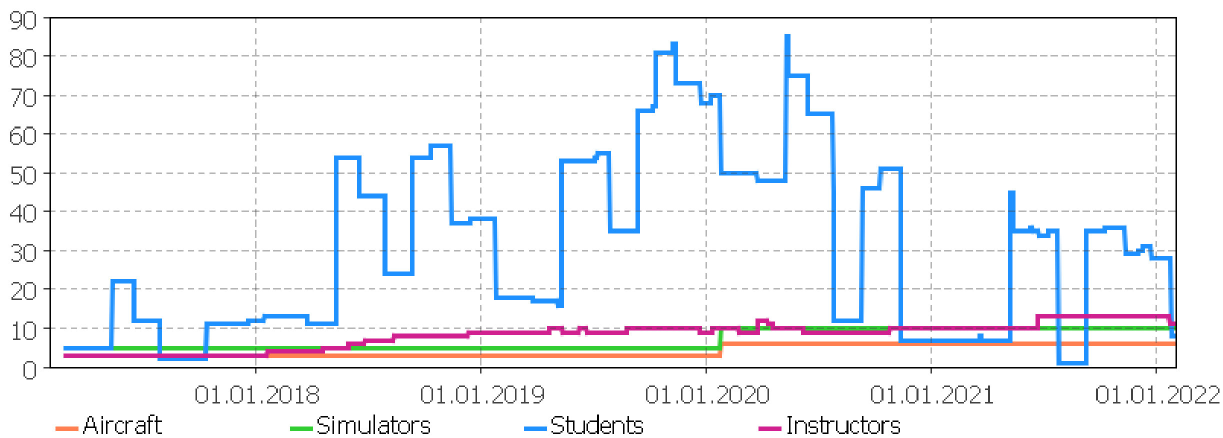

Figure 7: Aircraft, simulator, student and instructor count at School B over the transition period and beyond.

When the scenario was created, the model automatically prompted users as to how to handle the resources (e.g. relocate to a different school and at what rate) and courses attached to the school. Each of the six events, specified above, was then registered as a simulation event to be fired off at a specific point in simulation time. For example, when the simulation time hit 2018, the model fired off the first event, which triggered the creation of a new transition for instructors from School A to School B. As the transition was specified to be gradual, the model automatically performs basic calculation to smooth out the rate of the instructor transition over the two-year phase out period. Student transitions and adjustments of the number of course occurrences were also handled in the same manner. A major structural change took place in 2020 when School A was phased out and completely removed from simulation.

This scenario demonstrated the model's ability to dynamically reconfigure itself to reflect both the structural and parametric changes to the training pipelines. Moreover, the case study showed that the model could also handle gradual transition over time from the baseline state to the end state, which reflects what would happen during an actual phasing out of a platform.

6 CONCLUSIONS

This paper presented a hybrid, agile, agent-based and discrete event simulation framework for modelling HATC training pipelines. This work was developed from the need to build a simulation model for the Australian Defence Force aircrew training continuum and to provide what-if analysis capability to assist with strategic planning and decision making. The transient, constantly evolving nature of the training continuum was the major design consideration for the proposed framework. This object-oriented hybrid design enables the model to be dynamically reconfigurable to allow for the changing number of schools and courses throughout simulation, as well as the ability to adjust the affected transitions between courses accordingly.

The use of agents and agent state charts to model students and instructors and their individual behaviours has proven to be an efficient way of introducing a level of autonomy in the model. Moreover, the state charts can be expanded to include more complex behaviours and decision making heuristics, if the need arises. This is particularly important in modelling various governing policies at school and squadron levels, as demonstrated by our prototype model. The use of discrete event process flowcharts to model sequences of activities involved in running a course as well as the use of DES resource management functionalities proved to be beneficial. This approach allows for the identification of potential bottlenecks and to analyse resource utilization. The process flow charts can be expanded to model more complex activity flows without affecting the overall model.

By combining AB and DES approaches, the hybrid model can accurately represent both the process-centric view of the system as well as model individual-centric aspects of the training continuum. The proposed framework is applicable for any static and non-static training pipelines that involve autonomous entities going through a series of training activities which could require both human and material resources.

Future work includes optimization and further expanding the prototype model to include other aircrew streams, such as maintainers, and thus further validating the conceptual design.

REFERENCES

Anagnostou, A., A. Nouman, and S. J. E. Taylor. 2013. "Distributed Hybrid Agent-Based Discrete Event Emergency Medical Services Simulation." In Proceedings of the 2013 Winter Simulation Conference.

Bonabeau, Eric. 2002. "Agent-based modeling: Methods and techniques for simulating human systems." *Proceedings of the National Academy of Sciences of the United States of America* 99 (3):7280–7287.

Borshchev, A. , and A. Filippov. 2004. "From System Dynamics and Discrete Event to Practical Agent Based Modelling: Reasons, Techniques, Tools." The 22nd International Conference of the System Dynamics Society, Oxford, England, July 25 - 29, 2004.

Brailsford, S., and N. Hilton. 2001. "A comparison of discrete event simulation and system dynamics for modelling health care systems." Planning for the Future: Health Service Quality and Emergency

Accessibility. Operational Research Applied to Health Services (ORAHS) Glasgow Caledonian University.

Davenport, Jon, Charles Neu, William Smith, and Susan Heath. 2007. "Using discrete event simulation to examine marine training at the Marine Corps Communication-Electronics school." Proceedings of the 2007 Winter Simulation Conference.

Djanatliev, Anatoli, and Reinhard German. 2013a. "Large Scale Healthcare Modeling by Hybrid Simulation Techniques using AnyLogic." Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques, Cannes, France, 5-7 Mar 2013.

Djanatliev, Anatoli, and Reinhard German. 2013b. "Prospective Healthcare Decision-Making by Combined System Dynamics, Discrete Event and Agent-Based Simulation." Proceedings of the 2013 Winter Simulation Conference.

Fakhimi, Masoud, Anastasia Anagnostou, Lampros Stergioulas, and Simon J. E. Taylor. 2014. "A hybrid agent-based and discrete event simulation approach for sustainable strategic planning and simulation analytics." Proceedings of the 2014 Winter Simulation Conference.

Heath, S. K., S. C. Brailsford, A. Buss, and C. M. Macal. 2011. "Cross-paradigm simulation modeling: Challenges and successes." Proceedings of the 2011 Winter Simulation Conference.

Liraviasl, K. Khedri, H. ElMaraghy, M. Hanafy, and S.N. Samy. 2015. "A Framework for Modelling Reconfigurable Manufacturing Systems Using Hybridized Discrete-Event and Agent-based Simulation." 15th IFAC Symposium on Information Control Problems in Manufacturing — INCOM 2015.

Magee, C.L., and O.L. de Weck. 2004. "Complex System Classification." Proceedings of the 14th Annual International Symposium of the International Council on Systems Engineering (INCOSE), Toulouse, France, 20 – 24 June 2004.

Montevechi, J., D. Sena, E. Silva, A. Costa, and A. Scheidegger. 2015. "Hybrid simulation of production process of Pupunha Palm." Proceedings of the 2015 Winter Simulation Conference.

Séguin, René. 2011. "1 Canadian Forces Flying Traing School (1 CFFTS) Resource Allocation Simulation Tool." Proceedings of the 2011 Winter Simulation Conference.

Séguin, René , and Charles Hunter. 2013. "2 Canadian Forces Flying Traing School (2 CFFTS) Resource Allocation Simulation Tool." Proceedings of the 2013 Winter Simulation Conference.

Sterman, J. 1992. "System dynamics modeling for project management." Sloan School of Management, MIT (Online publication).

Swinerd, C., and K. McNaught. 2012. "Design classes for hybrid simulations involving agent-based and system dynamics models " *Simulation Modelling Practice And Theory* 25:118-133.

AUTHOR BIOGRAPHIES

VIVIAN NGUYEN is a Systems Scientist with the Australian Defence Science and Technology Group (DSTG). She holds a Bachelor of Information Technology (Software Engineering) (Hons) from University of South Australia (UniSA). She worked as a software developer in the areas of computer-supported collaborative work and smart-room environments. In recent years, she has been heavily involved in the area of modelling and simulation to support workforce analysis and organizational design. Her email address is vivian.nguyen@dsto.defence.gov.au.

MINA SHOKR is an Operations Analyst for Australian Defence Science and Technology Group (DSTG). He holds a B.Eng. (Honours) in Mechanical Engineering and B.Sci. in Computer Science from The University of Melbourne, Australia in 2009. He is currently completing a Masters in Operations Research and Management Science at The University of Melbourne. He has lead and contributed to

numerous Operations Research studies into military air tactics development and capability assessment using modelling and simulation. His email address is mina.shokr@dsto.defence.gov.au.

ANA NOVAK is an Operations Research Scientist for Australian Defence Science and Technology Group (DSTG). She received her B.Eng. (Honours) in Information Technology and Telecommunications from University of Adelaide in 2003. She obtained a Ph.D. degree in Mathematics from The University of Melbourne, Australia in 2006. Her research interests include queuing theory, probabilistic graphical models, human and computer vision, Bayesian data analysis, simulation, optimization, and decision support. Her email address is ana.novak@dsto.defence.gov.au.

TERRY CAELLI is a Professor at the University of Melbourne, The Royal Melbourne Institute of Technology and a contractor with the Defence Science and Technology Group, Department of Defence. His interests lie in Signal Processing, Human and Machine Vision, Pattern Recognition, Machine Learning and their applications in Health, Environment and Defence. He is a Fellow of the International Association for Pattern Recognition (FIAPR) and a Fellow of the Institute for Electronic and Electrical Engineers (FIEEE). His email address is terry.caelli@gmail.com.