# ASHiCS:

## Automating the Search for Hazards in Complex Systems

Kester Clegg & Robert Alexander
Dept. of Computer Science
University of York
York, United Kingdom
{kester||rda}@cs.york.ac.uk

***Foreword* - This paper describes a project that is part of SESAR Work Package E, which is addressing long-term and innovative research. The project was started early 2011 so this description is limited to an outline of the project objectives augmented by some early findings.**

***Abstract*—With increasingly complex systems to manage, safety analysts are starting to express concern that large complex systems are becoming too difficult to predict or guarantee safety when part of the system is changed or placed under stress. In order to help analysts discover hazards within complex systems, we propose a new generation of tools that make use of automated search heuristics and simulation to uncover hazards that might otherwise be missed using traditional (manual) safety analysis.**

***Keywords - ATM; modeling; simulation; safety; risk; evolutionary search.***

## I. Problem Description

As systems of systems (SoS) grow in size and complexity they become increasingly difficult to model and analyze for predictive purposes. When trying to discover factors that might lead to hazardous situations or a reduction in safety, the problem can be further compounded by the way in which safety issues can cut across functional boundaries and systems. When modeling such large systems, often composed of many smaller subsystems, the number of man hours required to exhaustively examine every consequence that changing part of the system might entail soon becomes prohibitive. An automated method that can both demonstrate sufficient coverage of the model's outcomes and provide assurance that hazards within the model have been found would be of benefit to safety engineers and system owners.

The model of a system can only be as good as the information put into it. However, creating high fidelity simulations can be extremely expensive, particularly in the case of air traffic control (ATC) where there is not only the accuracy of the data (such as the geometry of airports, flight profiles, etc.) to be concerned with, there may also be the staff costs of trying to simulate radio communication between controllers and pilots in real time and other decision making costs [1]. The costs of conducting such high fidelity simulations as part of air traffic planning have led to the development of fast time simulators that deploy their own conflict resolution algorithms while still being able to model some aspects of controller workloads. The benefits of fast time simulators are that many permutations of a proposed change can be simulated and the results gathered over a short time to be analyzed. Some aviation authorities, such as the FAA, have sufficient confidence in these results to use them as the basis of introducing changes to air traffic operations, others such as EUROCONTROL, may decide there are benefits to carrying out additional real time simulations [2].

Provided that engineers have confidence in their models, the output data of a simulation can be analyzed to discover information relevant to safety concerns. While this may be sufficient to discover hazards arising from a known or predicted state of the system, it does not provide confidence that system will be safe if it enters an unknown or unexpected set of circumstances. Hazards stemming from such circumstances are much more difficult to discover. Not only is it difficult to predict the outcome of unexpected events or situations, in large SoS it can be hard to determine the precise causal chain or set of circumstances that resulted in the hazard. This difficulty makes it hard and time consuming to discover hazardous situations through simulation and acts as a barrier to assurance that accident scenarios have been adequately tested with respect to their possible outcomes.

The ASHiCS project aims to reduce these concerns by investigating how a search tool could use existing simulation environments familiar to air traffic planners to look for hazardous situations or accidents. Given the difficulties of predicting the effects of making changes to part of a complex SoS, our project proposes a technique that could take a baseline scenario and automatically inject incidents, discover hazards or examine the knock-on effects of an emergency situation without the need for human intervention. By allowing the search to manipulate data used by the simulations, our heuristics should enable the rapid identification of hazardous situations and the causal chains that led to them.

Stating the aim of the project in this way is straightforward, but conducting an efficient search through the output of a series of complex simulations is not trivial. In a short paper such as this, there is insufficient space to give a full account of how we chose our form of search heuristic (evolutionary computation).

Instead we provide a brief background into evolutionary search and look at some related work involving simulation and search which is specific to air traffic control. Finally, we look at aspects that can influence search performance, such as the type of search landscape the simulation is likely to provide and our proposal to influence the shape of that landscape through the use of risk instruments.

## II. PREVIOUS WORK

### A. Evolutionary Computation (EC)

EC uses the process of natural selection as a search algorithm. Like biological evolution, the algorithm works over successive generations of a population, gradually moving the search closer to the objectives until either no more improvement in the population is possible or the objectives are satisfied. Each member of a generation represents a candidate solution, and each is tested against a measure of fitness. The fittest individuals are chosen and mutated slightly to form part of the next generation. The degree of mutation or how to combine individuals, the mechanisms of encoding the problem solutions, and the size and sampling of successive generations are all subjects that have received extensive study within EC.

EC has a distinguished academic and industrial record [3]. Within that time it has branched into variants [4], developed a canonical form [5] and been deployed in a wide range of industrial applications [6]. During the mid-1990s, EC began to draw media attention with claims that human-competitive patents were being discovered through the use of evolutionary search techniques [7] [8]. Evolutionary or genetic algorithms also found industrial application wherever the search for a design required taking a set of competing objectives into account. Indeed, so widespread was research in this area that an IEEE commissioned survey of the field by Coello [9] found over fifty different types of Multi-Objective Genetic Algorithm (MOGA). Examples of multi-objective optimization in an industrial context include Honda's "evolved" gas turbine fan blades [10], while more experimentally NASA has both initiated and carried out experiments to evolve antennae and other aerospace hardware [11]. Since the late 1990s, a great deal of work has also been done on using various forms of EC to discover new protein structures and encoding sequences in biology [12], where work has focused on effective coverage of very high dimensionality search spaces [13].

While much of the specialist search literature is concerned with the optimization of search performance, our aim is to demonstrate "just adequate" coverage of what is likely to be a very large and time-consuming parameter space to search. It is now well accepted that no search heuristic performs better than any other when averaged across all search landscapes (the "no free lunch theorem" of Wolpert and Macready [14] [15]). This means that without a detailed knowledge of our search landscape we cannot state definitively which type of search algorithm will be best suited to our purpose.

As previously mentioned, there are many forms of the evolutionary search paradigm, and considerable local expertise at the University of York on various forms of EC such as genetic algorithms (GA), genetic programming (GP) [16] and Cartesian Genetic Programming [17]. However, given that our initial plans for the project require the search to optimize a parameter set rather than evolve a function or program (as is generally the case with forms of genetic programming), we have taken the informed decision to experiment with different MOGAs, such as the non-dominated sorting genetic algorithm (NSGA-II) [18], differential evolution (DE) and others [19]. Our rationale for this is that initially our simulations will contain simple scenarios that vary by a few restricted parameters and until the initial experiments are carried out it is difficult to know the extent to which variability in these parameters (and others) will characterize the search landscape.

### B. Related work in the field of ATC / ATM

Within the domain of ATC / ATM research the use of automated search in conjunction with simulation has had limited application, resulting in relatively few published papers. However in 2011 a paper jointly authored by a team from the University of New South Wales and EUROCONTROL was published as part of the 9th ATM seminar series [20] which we would like to cover in greater detail as it demonstrates a technique similar to that proposed by ASHiCS and provides some evidence that method can work given sufficient resources and expertise. The paper describes the use of an EC type algorithm in conjunction with simulated scenarios to discover factors affecting delays. The authors used what was termed the Computational Red Teaming (CRT) Framework to identify patterns in arrival traffic and ground events that lead to delays in dynamic continuous descent arrivals (CDA) scenarios. CRT relies on a co-evolutionary search process that evaluates traffic distributions and ground events to identify delay bottlenecks in the system. Essentially the computational environment allows problems to compete with solvers, which the authors explain as follows: "problems are evolved to stress-test a system to identify its points of failures. The idea here is to play the devil's advocate where we evolve increasingly complex traffic patterns and constrained ground events which may lead to identifying tipping points in an advanced air traffic procedure operations [21] and to discover implicit relations in the scenarios patterns that lead to them [by] using data mining techniques".

The similarities to the ASHiCS approach are that an automated search process discovers increasingly difficult circumstances in the simulation to manage for ATC. However, we are concerned about the practicality of a co-evolutionary approach as part of an analysis tool. A co-evolutionary approach has no single static fitness function; the fitness function for each "species" evolves in response the competing evolutionary strategy of the other species. This has consequences in how the search space is explored and the types of problem that can be tackled using this approach. For example, a co-evolutionary approach requires competing objectives, and therefore the experiment design must consider how to represent each of the competitive elements. Experiment design, particularly with regard to scoping or constraining the search space, is often fraught with traps that can lead to deceptive and unexpected results (see early hardware experiments in which the search process made use of physical features of the test environment unintended by the researchers [22]). By focusing on a single search process, the risk of poor experiment design is reduced. The lack of a static fitness function also brings further considerations to experiment design, particularly if heuristics to aid the two searches are substantially different, leading to unequal rates of evolution between the competing objectives.

The CRT work, led by the Australian Defence Force Academy, was a relatively large project and has many aspects of similarity to the approach identified for ASHiCS. While the search target differed to ASHiCS (they were looking to find factors affecting operational delay; we are looking for safety hazards) and the type of search heuristic was likewise chosen to investigate a specific scenario, the work shows that by using careful scoping of the input variables exposed to the search process enabled the search to be tractable. While the work used a simulator which is unavailable to ASHiCS (ATOMS [23]), we will adopt similar techniques to ensure both that our search space size is restricted and that the generation of flights or events on those flights follows a realistic distribution in the chosen scenarios.

## III. METHOD DESCRIPTION

### A. The EC Search Harness

The scenario simulations will sit within what is termed a "search harness". This describes the EC software that "wraps" around the simulation software, allowing the search to automatically start, configure, stop and select those simulation runs that are of interest to us. In our case, a simulation that results in a hazard or risk is of interest, and will therefore be judged to have higher "fitness". The harness adds those simulations of higher fitness to a pool of good individuals using a ranking process, and uses them to create the next generation of simulations by carrying out some mutation and / or crossover of their genes. The next generation of simulations are run and assessed for fitness. The process is repeated until the levels of evolved fitness in the population either reach a plateau (where no more improvement is likely or possible) or a sufficiently good simulation is found that allows us to stop the search.

### B. ATM simulation Software

Most Air Navigation Services Providers (ANSP) use computer modeling, generally in the form of Fast Time Simulation (FTS) to estimate en-route capacity. En-route airspace capacity can be defined in "purely spatial criteria as the maximum number of aircraft through any given geometrical airspace for a given time period, based upon the spatial control constraints which govern the internationally specified separation between any two aircraft given their performance characteristics" [24]. The problem of course is that spatial criteria alone are insufficient as a means to estimate the safe throughput of aircraft: some measure of how the aircraft can be managed to ensure their safe separation is also needed, and for this we need to know whether the controllers can resolve conflicts between aircraft in a safe and timely manner. Trying to estimate the safe limits of controller performance requires us to build up what is called a controller workload model, and different aspects of such a model are generally incorporated into FTS software.

Clearly when carrying out many simulations as part of search process, it helps to have simulations run as quickly as possible in order to speed up the search performance. While "real time" simulation software exists for ATM, we believe that without extensive parallelization, such software would run too slowly to allow an effective search. However, the ability to run a simulation at fast time rather than real time is not the only consideration for our project. Crucial to the success of our approach is having the ability to automate the search process – i.e. it must be able to run without human intervention, as it is not uncommon in the field of EC for search runs to last several days or even weeks. Automation means that it should be possible to "wrap" the simulation in some way, so that inputs to the simulations can be created by an external application, and for the simulation to be started and have its output analyzed by the same process.

The ASHiCS team looked at several possibilities for FTS of ATM scenarios. Our criteria included the needs expressed above in terms of incorporating and communicating with the search harness, and also required the need for input data to be created or adjusted outside the simulation application using third party tools. ATM simulation software is not necessarily designed to permit integration with third party automation applications (such as required by ASHiCS) and this need left us with relatively few choices to consider: RAMS Plus, TAAM and ATOMS. Without going into our selection criteria in detail here, we chose to proceed with RAMS Plus on the basis of it having a suitable application programming interface (API) that could be made available to us.

### C. RAMS Plus

RAMS Plus is a FTS produced by ISA Software Ltd. who have offices in both the US and Europe. RAMS Plus has a long history of development in association with EUROCONTROL. The following description of RAMS Plus is based heavily on the User Manual [25] and on information gathered from personal communication with either developers or representatives working for ISA Software Ltd.

RAMS Plus generates 4D flights profile projections through the airspace. Profiles are calculated using cruise, climb and descent speeds via some 300-plus data-supplied aircraft models, each of which can be augmented or changed by the user. Controller workloads are dynamically calculated during the simulation. The workloads are data-defined, permitting general-to-specific airspace conditions to determine the weight attached to the workload event. For example, a flight climbing into a sector may be defined to generate more workload than a flight entering the sector in cruise, or an airline may generate less workload at a familiar airport than other airlines.

This ability to define flexible workload weightings is crucial for ASHiCS as we need to build up risk instruments and safety event models, which are likely to include specific workload tasks associated with resolving accidents or other types of incident.

ISA Software has stated to us that the API available for RAMS Plus offers a programmable way to access RAM internal variables during a simulation. By setting up network connection to RAMS during a simulation, we can create a program that will listen for a trigger event, such as a plane passing a way marker or level busting. On that event, our program can then pause the simulation and interrogate that aircraft to find out information such as the remainder of its flight path, current location, height, speed, etc. By requesting an update to this data, code written by ourselves could then return (for example) an updated flight path. This feature would allow us to simulate several types of safety incident, such as having to descend rapidly to FL100 in case of loss of cabin pressure, an unexpected 2D change to the previous flight plan of an aircraft, or even crash landing into terrain in the event of engine failure or fire. By examining the effect of safety indicators, such as separation, we can assess the knock on effects on the management of other nearby aircraft.

Given such an API to the scenario simulations, it should be possible to extend the models to support representing critical information systems such as SWIMS and ADS-B. One of our aims for the project is to examine the possible repercussions of shared data across different sub-systems that are corrupt, inaccurate or delayed.

The final part of our experimental set up is to incorporate an accepted model of safety. The following section describes the approach to safety used (with minor variations) by both EUROCONTROL and the Federal Aviation Authority (FAA).

### *D. Safety and Risk Model*

The system wide approach to safety and operational risk for ATM at EUROCONTROL is defined by what is termed the Integrated Risk Picture (IRP) that has as its scope gate-to-gate operations [26]. Its development is closely coordinated with the FAA, originally within the scope of the FAA / EUROCONTROL Action Plan 15 on Safety and more recently as part of the FAA's System Safety Management Transformation (SMST) program [27].

ASHiCS will take not only the principal parts of the IRP model, but hopes to use the historical data available to EUROCONTROL to provide realistic accident rates for our project. We intend to adopt the accident categories detailed in the IRP as our baseline for risk assessment. Although it is too early in the project to provide a definitive list of the hazards and accident categories that ASHiCS will incorporate, we include a brief summary below as this list has influenced our choices of scenarios.

#### *1) IRP accident categories*

The following accident categories are modeled in detail in the IRP [26] in order to quantify the ATM contributions to them:

1.) Mid-air collision - two aircraft come into contact with each other while both are in flight.

2.) Runway collision - two aircraft come into contact with each other on the airport runway, including cases where one aircraft is on the ground and the other is in flight close to the ground.

3.) Taxiway collision - two aircraft come into contact with each other on the airport maneuvering area. This includes collisions where one aircraft is parked, being pushed back, under tow, or taxiing up to the point of runway entry.

4.) Controlled flight into terrain (CFIT) – an aircraft collides with terrain, water or another obstacle while in flight without prior loss of control.

5.) Wake turbulence accident - an aircraft suffers major damage or serious injuries to occupants due to an encounter with wake turbulence from another aircraft.

The risks used in the IRP are averages over all commercial (passenger and cargo) flights in the European Civil Aviation Conference (ECAC) region. Historical experience has been used to supply three types of data for the IRP model:

- Accident and precursor frequencies.
- Causal breakdowns.
- Maximum effects of influences

For quantification of accident and precursor frequencies, suitable data sources for the IRP were restricted to those for which exposed populations are known. For each accident and incident, a text description of the known causal factors has been obtained and used to identify the reasons for failure of each of the barriers (see Figure 1).

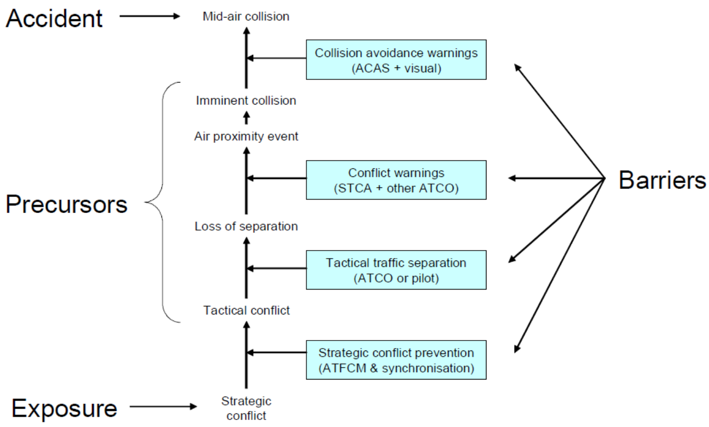


**Figure 1: Barrier diagram for mid-air collision (taken from [26]).**

ISA Software are currently implementing a means of representing the risk of incidents or events in RAMS scenarios for the FAA, as there is no existing functionality to attach the probabilities of events occurring in a RAMS simulation. The FAA's project looks at incidents such as runway overruns and how subsequent rejected landings have knock on effects on overall measures of safety, such as the reduction in overall aircraft separation (see [28]). However, it is not clear whether ASHiCS could make use of such risk events as part of the search process, as the implementation is still at an early phase and introducing stochastic variation during a search run might invalidate the results. The current approach adopted by the FAA and ISA Software is to use a combination of air traffic

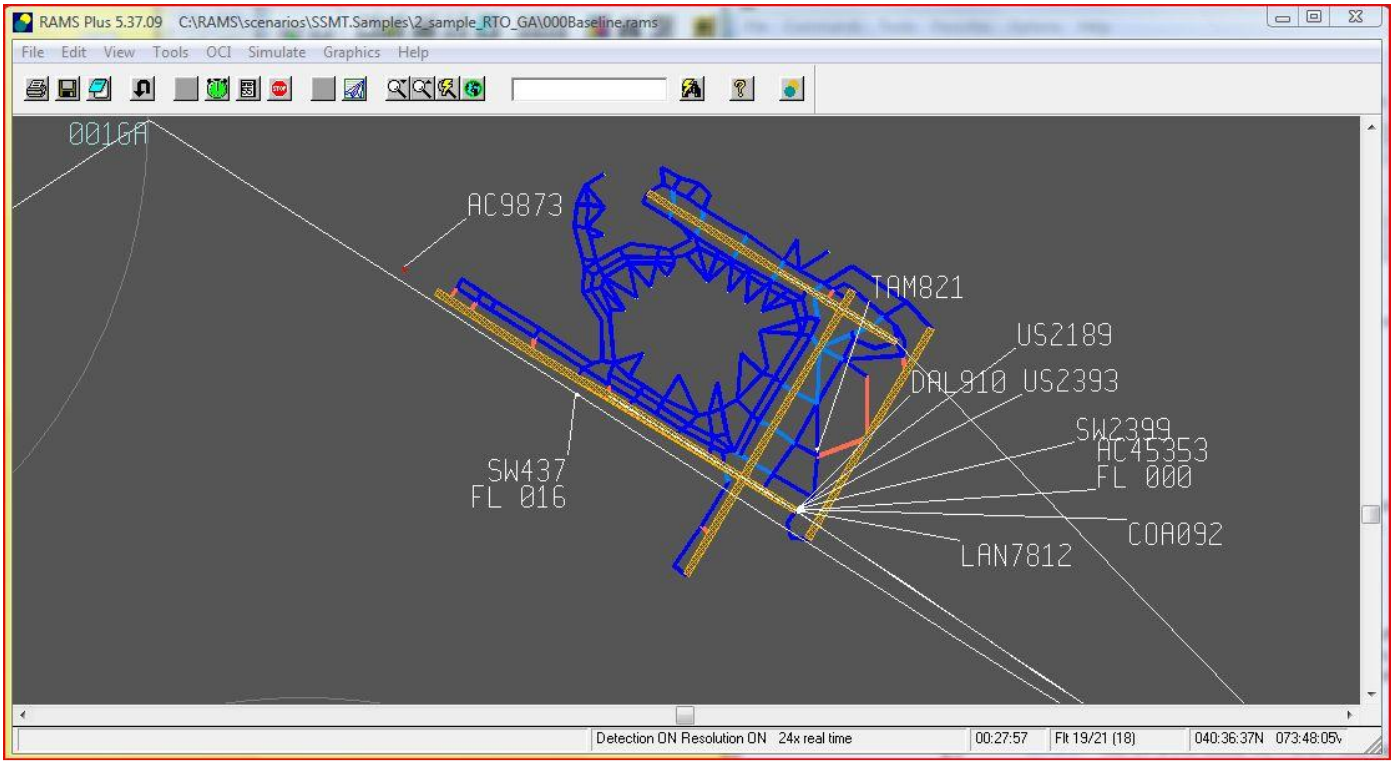


**Figure 2: Part of simulation run showing a plane (AC9873) overshooting the runway, causing the following flight to reject landing (image provided by Kenny Martin, ISA Software Ltd).**

density and event probability to obtain a stochastic distribution of generated airspace events in their simulations (this again bears similarity to the Gaussian distribution applied by [24]). ISA Software to use similar event probabilities based on event sequence diagrams (ESDs) or fault trees in our later scenario models. While ASHiCS may not approach either traffic or event generation in exactly the same way, we hope to use similar event probabilities based on event sequence diagrams (ESDs) or fault trees in our later scenario models.

*E. Initial Scenario*

We have selected sudden loss of cabin pressure (decompression) as a suitable incident to simulate for our initial scenario. By starting with a relatively simple scenario, such as a single plane losing altitude, ASHiCS aims to quickly build up the project code to test that: a) the search harness was able to communicate and issue updates to RAMS Plus during run time; b) that the search was able to demonstrate coverage of the given flight path; c) that a fitness function could parse the report outputs. Once this initial step has been achieved, it would form a platform that will be scaled up to include more planes and within which we could begin in insert more complex risk instruments to guide the search process. By scaling up as the development of the software proceeds, we hope to be able to adapt and change the search heuristics in order to try and improve search performance.

Decompression is defined as the inability of the aircraft's pressurization system to maintain its designed pressure schedule (Skybary: http://www.skybrary.aero). If the aircraft is equipped, this can trigger the Automatic Emergency Descent System (AEDS) which may initiate the descent if the pilots fail to respond to a cautionary alert - potentially indicating that the crew is incapacitated through the effects of hypoxia. The aircraft will also be put into a rapid descent at its maximum operating speed towards FL 100 – the target altitude for depressurization incidents, at which oxygen masks are no longer necessary. We intend to replicate a similar decompression incident to that which occurred to a Boeing 737 on 9th May 2010, whose full details can be found in the report AAIB Bulletin: 9/2010 EW/C2010/05/01.

However, we will first simulate an emergency descent that maintains the aircraft's original flight plan. The second phase will attempt an emergency descent followed by an en-route diversion. Our scenario will not replicate the geographical details of the above incident, nor will it be restricted to the Boeing 737 aircraft type. Later adjustments to the scenario will increase the air traffic density within the air sector to test the effectiveness of the search performance in finding the "worst case scenario" for a random decompression incident to occur to any of the aircraft in the simulation. We wish to emphasize that our choice of incident for the initial scenario does not imply our future work will investigate similar incidents or use the same measures of risk.

## *F. Evaluation of search heuristics*

After further developing the initial scenario to the extent we believe it represents an accurate characterization of the more complex scenarios we intend to study, we will conduct a short comparison of search heuristic performance. This is something missing from other studies [20] and which is often at best only carried out against other EC-type algorithms [9].

## *G. Progess to date*

A detailed plan of work is available from our first project deliverable which describes the steps taken to create the initial scenario and design the search schema. Provided we encounter no unexpected delays obtaining or working with the RAMS API, our expectations are that we will have our first scenario simulations (i.e. the random triggering of cabin pressure loss to an aircraft within a RAMS scenario) and search harness ready for early 2012. We will take on board the recommendations of a EUROCONTROL advisory panel held $14^{th}$ November 2011 and hope to extend our initial scenarios to include the recommendations during the summer of 2012.

# IV. Expected Outcome and Contribution to SESAR

Our long term aim is demonstrate that automated search can play a part in the safety assessment of operational scenarios. We expect that as air traffic density increases with the introduction of SESAR, more sophisticated techniques will be required to check that safety-related incidents do not have unintended consequences for other parts of the complex system that makes up modern ATM.


# Acknowledgment

ASHiCS would like to thank Dr. Sherry Borener of the FAA's System Safety Management Transformation Program [27], who made it possible for the ASHiCS team to meet with herself and developers from ISA Software Ltd. (Kenny Martin (USA) and Ian Crooks (France). ASHiCS would also like to thank Ian Crooks for discussions about our initial safety incident scenario to simulate within RAMS Plus.


# References


[1] A. Cook, European Air Traffic Management: principles, practice and research, Ashgate Publishing Ltd, 2007.

[2] A. Majumdar, "Understanding En-Route SEctor Capacity in Europe," in *European Air Traffic Management: principles, practice and research*, Ashgate , 2007, pp. 65-95.

[3] D. B. Fogel, Evolutionary Computation: the fossil record, Piscataway, 1998.

[4] D. B. Fogel, "Evolutionary Computing," *IEEE,* 2000.

[5] D. Dimutrescu, l. B. Lazzerine, L. Jain and A. Dumitrescu, . Evolutionary Computing, CRC Press LLC, 2000.

[6] G. Robinson and P. McIlroy, "Exploring some commercial applications of genetic programming," *Evolutionary Computing,* Vols. , Number 993. , 1995.

[7] C. Fonlupt, "Book review: Genetic programming IV: Routine human competitive machine intelligenc," *Genetic Programming and Evolvable Machines 6,* p. 231–233, 2005.

[8] J. R. Koza, M. Keane and M. Streeter, "Evolving inventions.," *Scientific American*, p. 52–59., 2003.

[9] C. A. Coello, " An updated survey of GA-based multiobjective optimization techniques," *ACM Computing Survey,* vol. 32, no. 2, p. 109–143, 2000.

[10] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing,* vol. 9, no. 1, pp. 3-12, 2005.

[11] J. Miller, "Review: First NASA DOD Workshop on Evolvable Hardware 1999," *Miller, J. (2000). Review: First NASA DOD Workshop on Genetic Programming and Evolvable Machines 1,* vol. 1, p. 171–174, 2000.

[12] R. Goodacre, "Making sense of the metabolome using evolutionary computation: seeing the wood with the trees," *Journal of Experimintal Botany,* vol. 56(410), no. , pp. 245-254 , 2005.

[13] P. S. Ngan, M. L. Wong, W. Lam, K. S. Leung and J. C. Cheng, "Medical data mining using evolutionary computation," *Artificial Intelligence in Medicine,* vol. 16, no. 1, 1999.

[14] D. Wolpert and W. Macready, "No free lunch theorems for optimization.," *IEEE Trans. on Evolutionary Computation,* vol. 1, no. 1, p. 67–82., 1997.

[15] D. Wolpert and W. Macready, "No free lunch theorems for search," Santa Fe Institute, Santa Fe, NM, 1995.

[16] J. R. Koza, Genetic Programming II: Automatic Discovery of Reusable Programs., MIT Press, 1994.

[17] J. Walker and J. F. Miller, "The Automatic Acquisition, Evolution and Re-use of Modules in Cartesian Genetic Programming," 2008.

[18] D. Kalyanmoy, Multi-Objective Optimization Using Evolutionary Algorithms, Wiley, 2001.

[19] K. Price, M. S. Rainer and J. A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series), Springer, 2005.

[20] S. Alam, W. Zhao, J. Tang, C. Lokan, H. Abbass, M. Ellejmi and S. Kirby, "Discovering Delay Patterns in Arrival Traffic with Dynamic Continuous Descent Approaches using Co-Evolutionary Red Teaming," in *9th ATM Seminar*, Berlin, 2011.

[21] A. Bender, H. Abbass and S. Alam, "MEBRA: multiobjective evolutionary-based risk assessment," *IEEE Computational Intelligence Magazine,* vol. 4, no. 3, p. 29–36, 2009.

[22] A. Harding, Hardware Evolution: Automatic Design of Electronic Circuits in Reconfigurable Hardware by Artificial Evolution, Springer, 1999.

[23] S. Alam, H. A. Abbass and M. Barlow, "ATOMS: Air Traffic Operations and Management Simulator," *IEEE Transactions on intelligent transportation systems,* vol. 9, no. 2, June 2008.

[24] EUROCONTROL, "European ATC Harmonization and Integration Programme (EATCHIP) - Report Phase 1," 1991.

[25] ISA Software, *RAMS Plus User Manual Version 5.36,* 2011.

[26] E. Perrin, B. Kirwan and R. Stroup, "A Systemic model of ATM Safety: the integrated risk," in *7th ATM Seminar*, Barcelona, 2007.

[27] FAA, "System Safety Management Transformation," May 1st, 2011.

[28] S. Borenor, "Innaxis Complex World - Key note presentation: Complexity and safety performance - Will circular accident causality result from NextGen and SESAR?," 2011. [Online]. Available: http://complexworld.innaxis.org/presentations/borener.pdf.

[29] D. Goldberg, Genetic algorithms in search, optimization, and machine learning, Addison-Wesley, 1989, pp. Goldberg, D. (1989). Genetic algorithms in search, optimization, and machine learning. Addison-Wesley.

[30] B. Hilburn, "Cognitive Complexity in Air Traffic Control: A literature revew," EEC note 04/04, 2004.

[31] K. Martin, *SSMT Modeling Approach, presentation given on behalf of ISA Software for a System Safety Management Team (SSMT) FAA team meeting in Washington DC on 27 April 2011,* 2011.