

# The Development of a Numerical Simulation for Aviation Maintenance Technician Training Course Design

**P. Carlo Cacciabue  
Douglas Owen  
Carlo Mauri**

European Commission  
Joint Research Centre  
21020 Ispra (VA), ITALY  
pietro.cacciabue@jrc.it

## Abstract

The priorities of the simulator were to develop a structure that can be applied to many aviation maintenance tasks to provide useful output to assist in the design of aviation maintenance training systems. The simulator was based on three main steps: design, implementation, and run. The theoretical models applied are compatible with the "Information Processing System" paradigm and the practical application show the potential applications of the simulation. Sample output from the simulator is included.

## Introduction

One of many factors affecting the safety of modern commercial aviation is that of aviation maintenance. While the percentage of commercial aviation incidents caused by maintenance errors is relatively low, the accidents tend to be serious. Moreover, accidents related to aviation maintenance are also almost exclusively due to human error. The aim of the work described here was to develop a tool based on numerical simulation to assist in the development Aircraft Maintenance Technician (AMT) training, to reduce the rates and severity of human error.

Specifically, the simulator was designed to analyse maintenance tasks in order to identify the interactions of AMTs with the work environment, and task demands that may result in erroneous performances. Furthermore, the simulator may be used to both develop training programs which emphasise vulnerable action sequences, and to analyse the quality of the procedures that AMTs work from. The tool was built using an object-oriented language based on a relational database.

The conceptual approach focussed on: output quality, in terms of being able to produce a range of potential AMT behaviours; flexibility to simulate the performance of many maintenance tasks; and logical integrity of the internal structure of the simulator. The cognitive validity and

fidelity of the simulator were not a priority, as no theory has yet to encompass all aspects of human cognition operating in complex and demanding environments. However, some elements of the model *represent* basic cognitive processes that are necessary to model possible AMT behaviour. To some extent, from the perspective of this tool, it is quite irrelevant what particular cognitive processes result in a particular error. Rather, it is more important that the simulator is able to reproduce errors that are similar or compatible with those of a real AMT and to accurately replicate their effects.

This paper describes the contributing theoretical models, the internal workings of the simulator, the informational requirements to simulate a maintenance task, the functionality and utility of the simulator output. Finally, the usability, potential, and limitations of the simulator are discussed.

## Simulation Theory

This simulator was primarily based on the integration of two existing models: SHELL (Hawkins, 1987; Edwards, 1988) and RMC/PIPE (Cacciabue, 1998) (Figure 1). The integration of these two models provides a comprehensive structure with which to simulate the actions of an AMT operating in a complex and dynamic work environment that comprises of varying task demands and environmental and organisational conditions. A brief description of the SHELL and RMC/PIPE models follows.

### Shell model

The SHELL model describes the interactions between human, procedures, and equipment in a work environment using five elements:

Software (S): Maintenance procedures for a task.  
Hardware (H): Tools, materials (lubricants, anti-corrosion agents, etc.), objects, and equipment.

Environment (E): External organisational and environmental context.

Liveware (L): The characteristics of the AMT.

Liveware (L): Other humans in the task environment (e.g., colleagues, supervisors).

The interactions between the elements in the SHELL model are as follows with examples from aviation maintenance:

Human-Environment (HE): The effect of temperature, rain, wind, organisation issues on the AMT.

Human-Hardware (HH): The effects of the interaction of the AMT and tools and objects used for the task; e.g., electrical generators, spanners, platforms.

Human-Software (HS): The effects of the interaction of the AMT and the ADRES manual procedures that are followed to complete the task.

Human-Human (HH): The effects of the interaction between the AMT and other people like colleagues and supervisors.

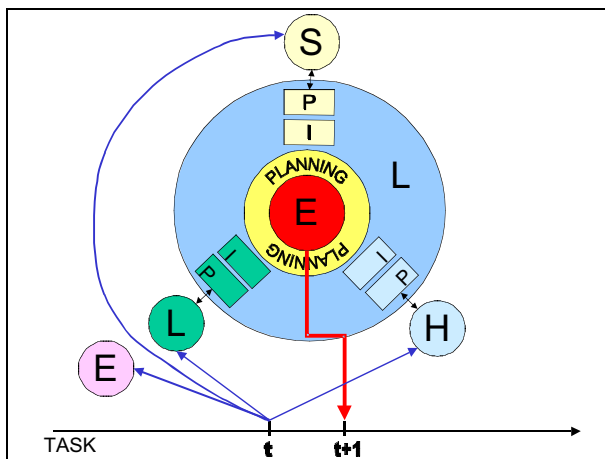


Figure 1: Integration of RMC/PIPE and SHELL models

## RMC/PIPE Model

The Reference Model of Cognition (RCM) is based on the well known “Information Processing System” paradigm for describing cognitive and behavioural performance of human beings interacting with machines (Neisser, 1967). In this case, we have considered only the four simple cognitive functions of Perception, Interpretation, Planning and Execution (PIPE). In the RMC, the PIPE functions are governed and supported the cognitive processes AoR (Allocation of Resources) and KB (Knowledge Base). The effects of AoR and KB on the behaviour of the simulated AMT are considered through the modelling of errors and the interactions with other actors and working contexts, according to SHELL model.

In the simulator, we focused on the processes internal to an individual, rather than the interaction of the individual with external events. Using the integrated SHELL and

RMC/PIPE models, the virtual AMT plans and executes an action in light of the interactions with the SHELL elements through RMC/PIPE functions.

The execution of a particular action is the output from the processes of perception, interpretation and planning. These three elements are related to the “t” instant. The execution of an action permits the passing to the next instant “t+1” (Figure 1).

The planning process comprises the perception and interpretation of the Hardware, Software, and Liveware. The execution of an action is affected positively or negatively by the task demands of the action (Hardware) and the Environment. The environment influences the AMT indirectly through, for example, temperature, time pressure and other company rules.

Every variable that makes up each of the SHELL elements is defined within this simulator. Variables can either be fixed for the duration of a simulation run through a maintenance task, or they can be dynamic changes in response to AMT actions, or a follow pre-set curve. The variables that define the AMT are also subject to change throughout the task. The variables that define the AMT characteristics determine whether the AMT is more or less susceptible to error under certain conditions.

## Action Execution Structure

The practical application of the integrated SHELL and RMC/PIPE models is best seen in the action execution flowchart (Figure 2). An action is a discrete behaviour performed by the virtual AMT that modifies the object and/or tool states. For every action that is performed, the simulator uses this flowchart to determine the outcome of each action. The path that is taken through this flowchart describes the decisions made by the virtual AMT that lead to production of a particular action.

The output of the action execution flowchart can be either correct, or erroneous with commission or omission error, or may result in a recovery action (if necessary), or to end the task. This structure was designed to enable the reproduction of all the possible erroneous actions that an AMT can perform.

This flowchart is based on several decision blocks and possible action outcomes. Each block is related to a particular function performed during an action execution by either the AMT or the simulator. Each block represents functions defined by the integrated model that are necessary to produce the range of AMT actions. Some of these blocks represent matching functions performed exclusively by the simulator. These act to compare the current tool and object states with the ideal states. In this way, the simulator is able to detect if all the requirements have been met to move on to the next action. The final step is to update system as a result of the outcome of the action executed.



## Functionality

A simulator run is based on three major stages: the set-up, the simulation run itself, and the generation of output data (Figure 3).

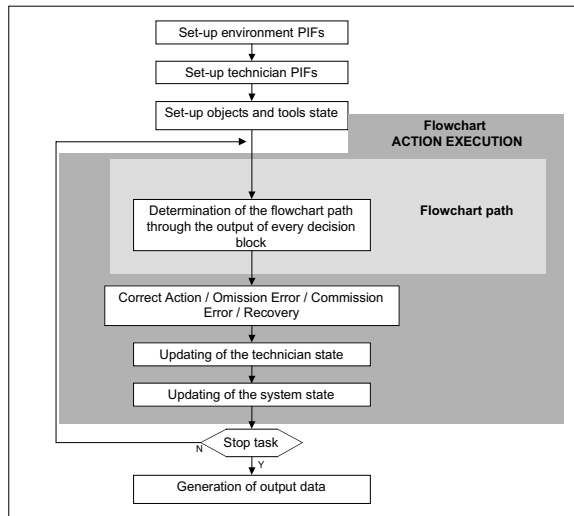


Figure 3: Simulator run structure

**Set-Up.** During the set-up all starting values of the variables are assigned for the AMT and the environment. The starting states of all tools and objects are also defined. In addition, the user can select to simulate the entire maintenance task, or part of the task. At any one moment during the simulator run, the states of the tools and objects variables define the system state.

**Tool States:** The software can automatically determine the correct state of tools and objects to start the simulator. Changes to this default state can be made manually. **Object States:** Starting states for all objects that are not tools are defined in the same way as the tool states. Changes can be made as above.

**AMT Characteristics:** The definition of these variables allows the user to set the characteristics of the virtual AMT before a simulation run. Variables include dynamic states like fatigue and stress which can be affected by task demands and environmental conditions, and stable states like expertise.

**Environmental conditions:** The environmental variables are made up of Performance Influencing Factors (PIFs) and define the work environment of the virtual AMT. The environment is defined by modifying organisational, weather, time, and lighting variables.

**Simulation run.** During the simulation run the simulator goes automatically through every action using the action execution flowchart, taking into account the effects of the AMT characteristics and PIFs. Throughout this process the simulator records information about the path taken for each action execution and events that occur during the run.

**Output generation.** The simulator is able to produce a comprehensive data set, logging the path taken through the action execution flowchart, the outcome of every action, the effects of every error, and the recovery actions taken to rectify erroneous system states. Also, the output can plot the curves of AMT and environmental conditions throughout the task.

## Applications

The utility of this software stems from the variety of ways to induce errors during a simulation run. Basic random and manual error insertion modes focus entirely on effects of error on the prescribed ideal action sequence, whereas the more complex Error Inducing Functions have a wider scope incorporating environment/AMT/task feedback loops.

The aim of Error Inducing Functions was to provide structure for refinement toward a model representing the complex interactions between the real environment, the task demands, and the AMT in the domain of aviation maintenance. This function was designed to allow the simulation of all these factors working in parallel, as they may do in the real work environment. Therefore, this mode has the potential to be a very powerful tool. However, if care is not taken in entering appropriate values that represent fully the relative weights of environment variables and AMT variables, the simulator output has the potential to be quite misleading.

### Basic Error Insertion

**Random Error Insertion.** This mode of error insertion allows the simulator to insert a random number and position of randomly selected error types into the simulation run. It serves the function of allowing a 'quick and dirty' check of the integrity of the procedure/action sequence and error recovery procedures. Further, with multiple simulator runs, this mode can be used to identify if and where critical steps exist that are vulnerable to any sub-optimal task performance.

**Manual error insertion.** This mode allows the user to manually select and insert the position and type of error to the simulator. This allows the simulator to generate answers to 'What happens if...?' questions. The answers to these questions are limited to the path subsequently necessarily followed through the action sequence and/or recovery actions.

**Pre-set AMT settings mode.** This mode can demonstrate the effects of failures that can occur at stages and sub-stages of the PIPE model. For example, a simulation run can be made demonstrating consistently poor planning performance through a high execution threshold of an 'analyse the system' block making its performance unlikely.

No AMT variables are modified in this mode from task demands or environmental variables. Therefore the output represents only the interaction between the task performance (software and hardware) and an unchanging AMT model (liveware) as described by the SHELL model.

### Advanced Error Insertion

**Error Inducing Function.** This mode utilises a feedback loop. After each action is performed, the states of the AMT are incremented as a function of the environmental conditions and the task demands either towards or away from an error threshold. For example, an intricate and demanding action will increment the AMT closer to an erroneous action, while a very simple task may produce no increment, or move the AMT’s state away from errors. When the AMT follows particular path an error can be produced of either an omission or commission error type. Due to a lack of empirical data, it is impossible to link a specific combination of environment and task demand variables to a specific error type. In reality, there may in fact be a degree of randomness to this link. In this simulator, the action execution flowchart dictates the error type. If the error produced is a commission error then a probability function selects one of the pre-determined commission errors defined for the action in question.

**Error inducing function plus manual error insertion.** This mode couples the scope of the Error Inducing Function with the ability to manual insert an error. This mode of operation allows the user to view the effects of an error on the state of the AMT and the subsequent performance over the rest of the task. When used in this way, the simulator can be used to detect errors that may escalate over the course of the task, or result in greater probability of performing further errors in interaction with the selected environmental variables.

**Error inducing function plus training effect model.** Here, the effect of training can be simulated by manually enhancing or diminishing the likelihood of the simulated AMT choosing one action path over another. For example, if a training program focussed on the importance of reading the procedures before performing an action simulator runs can be made with a low likelihood of reading the procedures versus a high likelihood of reading the procedures. This could then demonstrate the potential effects on overall task performance if this type of training was effective.

### Sample Output

To demonstrate the interaction between the environment, AMT, and task demands a sample output from a simulator run is provided using the Error Inducing Function Mode. The output comprises thirteen steps from the initial stages of the ASC removal maintenance task (Figure 4, and Table

D). The AMT characteristics set for the simulation run were to represent an AMT who was of medium-high expertise, not fatigued, and with medium motivation.

The only environmental PIF configured was humidity with a medium starting value. The humidity was set to rise over the run. All the objects and tools were in their correct state at the start of the simulation run.

For eleven of the thirteen steps the AMT makes no errors. However, the AMT makes errors in step 11 and step 13. These are the results of the combination of the increasing humidity and the task demands affecting the AMT state that subsequently detrimentally affected task performance.

The error in step 11 was to damage the wire of the electrical connector while securing it to the aircraft structure. The error in step 13 was in failing to plan an inspection of the system, and therefore the need to place a protective cap on the electrical connector was not realised and not performed.

Step	Action Code	Description
1	3.D.2.A.01.01	Grab Screwdriver
2	3.D.2.A.01.02	Remove all screws (APSS)
3	3.D.2.A.01.03	Place screws (APSS) in container
4	3.D.2.A.01.04	Grab panel
5	3.D.2.A.01.05	Place on ground
6	4.A.1.A.01.01	Grab aileron (LA)
7	4.A.1.A.01.02	Move aileron up until ASC piston rod fully extended
8	4.A.2.A.01.01	Grab flange on connector (elecon 01)
9	4.A.2.A.01.02	Rotate flange anti-clockwise 1/4 turn to unlock
10	4.A.2.A.01.03	Pull connector (elecon 01) downwards
11	4.A.2.A.01.04	Secure wire (elewire_01) to right of ASC body through triangular wing structure
12	4.A.3.A.01.01	Grab cap (cap_03)
13	4.A.3.A.01.02	Press cap (cap_03) on cable socket

Step 11: Commission Error;

Step 13: Omission Error.

*Table I: Simulation Run Steps*

The two errors committed were of different types; the first being a commission error, and the second an omission error. The factors contributing to the commission error were high values of stress, workload and physical fatigue, and reduced motivation that caused an error in the execution block of the step. The factor contributing to the omission error was principally the high value of mental fatigue that caused the AMT to ‘decide’ not to analyse the system. A decision to analyse the system could have led the AMT to recover from an earlier memory failure and to recognize the appropriate action to perform.

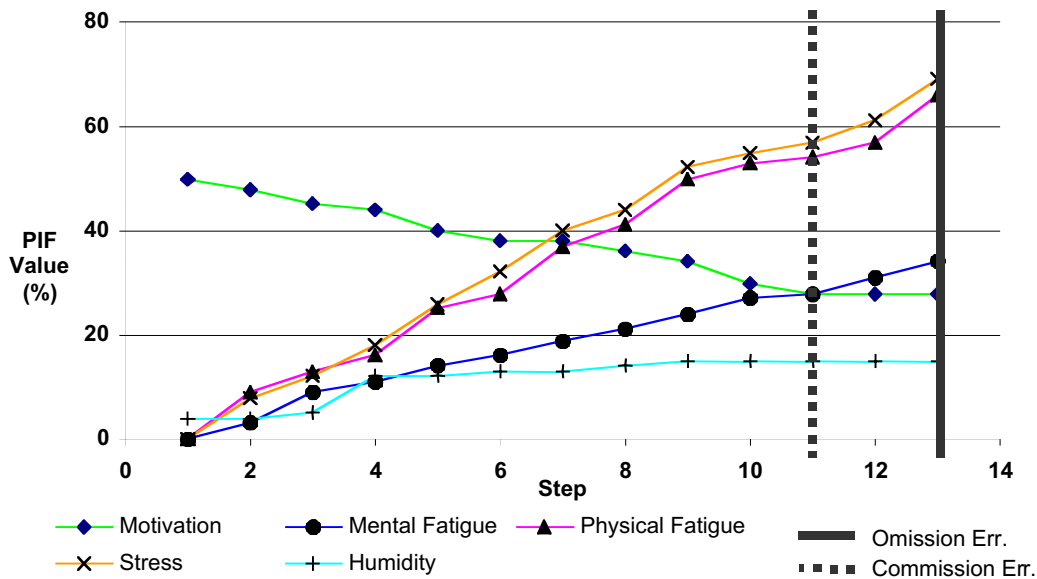


Figure 4: Graph of relevant Environment and Technician PIFs using sample data

### Usability

While usability is not a high priority at this early stage in the development of the numerical simulator, some consideration has been given to this issue. An interface has been developed to aid the entering of data into the simulator for maintenance procedures including the removal/installation of the Aileron Servo Control that this prototype simulates.

The flexibility of the Error Inducing Function also required an interface to be developed that allows the user to make adjustments to how the function affects task. The graphical user interface (GUI) that has been developed focussed on speed of data entry and displaying inter-relationships between variables. This system makes use of mouse driven sliders and has the additional functionality that several action steps can be grouped to use the same settings where appropriate. Grouping actions together can lead to very large reductions in the number inputs required to set up the simulator for use.

### Limitations & Potential

The assembling of comprehensive data is vital to the utility of numerical simulations. No matter how good or valid the structure of a numerical simulation, the simulator can be rendered useless by incomplete or inaccurate information.

The issues relating to the limitations of this simulator mainly relate to the validation of the simulator output. As yet there are almost no empirical data on the interactions of an AMT with the real work environment. Although what data was available has been used. Currently the simulator is using rather arbitrary values for many of the variables regarding task demands, the AMT, and the magnitude of PIF interaction effects on AMT performance.

In relation to this point, it is important to note the domain specific nature of the tasks, operator, and environments applicable to aircraft maintenance. The work of an AMT is comparable to other tasks on a general level. It is complex work, being carried out by experts in variety of environmental conditions, and in the context of organisational pressures. However, the specific interactions of the task, operator, and environment in aviation maintenance are unique. Therefore, the validity of the simulation output relies absolutely on information of the aviation maintenance task, the aviation maintenance technician, and the aviation maintenance environment.

The gathering of this data itself is a challenge due to the relatively informal and slippery nature of the work practices in comparison to, say, an aircraft pilot's. A methodology to conduct basic field observations with the aim to objectively measuring and quantifying relevant variables and interactions has been designed.

However, providing these validity issues can be solved and reliable data can be shown for several maintenance tasks then it may be possible to demonstrate that the simulator has predictive validity. If this is shown, then the simulator has the potential for the analysis of maintenance procedures during their development phase without the same extent of trial and error evaluation, which still occurs within the aircraft maintenance organisations. Designers of the maintenance procedures may be able to evaluate the procedures for their logical integrity, and more importantly, for their susceptibility for AMT error generation. This is of significant benefit to safety as a considerable number of different workplace usage scenarios could be simulated and those steps vulnerable to error could be identified without an error being made on a real aircraft that can be both expensive and pose a significant risk.

A more technical issue is that, in its current form, the simulator considers only one technician working in isolation on one task. It is usual for AMTs to work as a team or at least to give assistance for a few steps during a colleague's task. The object oriented design used for this simulator allows great flexibility in this direction. It would certainly be possible to add more AMTs in the future using the existing simulator structure.

If this approach is given sufficient support, it may provide the basis of a cheap, but effective and fast, tool for evaluating many practices in aviation maintenance.

## References

- Cacciabue P.C. 1998. *Modelling and Simulation of Human Behaviour in System Control*. ISBN 3-540-76233-7 Springer-Verlag, London, UK.
- Edwards, E. 1988. Introductory overview. In E.L. Wiener, and D.C. Nagel eds. *Human Factors in Aviation*, Academic Press, San Diego.
- Hawkins, F. H. 1987. *Human factors in flight*. Gower, Aldershot (UK).
- Hollnagel, E. 1993a. *Human Reliability Analysis: Context and Control*. Academic Press, London.
- Hollnagel, E. 1993b. Requirements for dynamic modeling of man-machine interaction. *Nuclear Engineering and Design*.
- Neisser, U. 1967. *Cognitive Psychology*. Appleton-Century-Crofts, New York.
- Norman, D. A. 1981. Categorization of Action Slip. *Psychological Review*, 88, 1-15.
- Pew, R. W., and S. Baron 1983. *Perspective on performance modeling*. *Automatica*, 19, No 6, 663-676.
- Rasmussen, J. 1986. *Information processes and human-machine interaction*. Elsevier- North Holland, Amsterdam. The Netherlands.
- Reason, J. 1990. *Human error*. Cambridge University Press. Cambridge UK.
- Sheridan, T. B., and W.R. Ferrel. 1974. *Man-machine system: Information, control and Decision Model of Human Performance*. MIT Press. Cambridge. MA.