

# Modelling Personnel within a Defence Logistics Maintenance Process \*

Guy Edward Gallasch  
Computer Systems  
Engineering Centre  
University of South Australia  
Mawson Lakes Campus, SA,  
5095, AUSTRALIA  
guy.gallasch@  
unisa.edu.au

Benjamin Francis  
Land Operations Division  
Defence Science and  
Technology Organisation  
P.O. Box 1500, Edinburgh, SA,  
5111, AUSTRALIA  
benjamin.francis@  
dsto.defence.gov.au

Christopher Moon †  
University of Adelaide  
North Terrace Campus, SA  
5005, AUSTRALIA  
christopher.moon@  
adelaide.edu.au

Jonathan Billington  
Computer Systems  
Engineering Centre  
University of South Australia  
Mawson Lakes Campus, SA,  
5095, AUSTRALIA  
jonathan.billington@  
unisa.edu.au

## ABSTRACT

In a military deployment, the operational availability of deployed equipment will be affected by many factors including equipment usage rates, sparing policy, and importantly the composition and distribution of maintenance personnel. In this paper, we present a Coloured Petri Net model that captures the potentially complex defence logistics maintenance process that links these factors together. We simulate this model with the CPN Tools simulator and identify a number of simulation performance concerns related to the incorporation of personnel in the model. A number of alternative models of personnel are evaluated in an attempt to attain a level of simulation performance from CPN Tools that is acceptable for business reviews and headquarter operations planning. Finally, we present some directions for future investigation to improve simulation performance further.

## Categories and Subject Descriptors

I.6 [Simulation and Modelling]: Model Development, Simulation Support Systems; H.4.2 [Information Systems Applications]: Types of Systems—*decision support, logistics*

\*This work was supported by Australia's Defence Science and Technology Organisation, Contract No. 4500498737.

†Christopher Moon undertook this work while he was with the Computer Systems Engineering Centre

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*SIMUTools* 2008, March 3-7, 2008, Marseille, France  
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

## General Terms

Performance, Experimentation

## Keywords

Coloured Petri Nets, Maintenance, Simulation Performance, Models of Personnel

## 1. INTRODUCTION

The Computer Systems Engineering Centre (CSEC) of the University of South Australia and the Logistics Mission of the Land Operations Division of Australia's Defence Science and Technology Organisation (DSTO) are working on a collaborative project to model aspects of Defence logistics [7]. A dynamic Maintenance Planning Tool, based on a Timed Coloured Petri Net (CPN) [14] model, is being developed to examine the effectiveness of a deployed maintenance capability for the Australian Army. In particular the Army is interested in a tool to both validate the feasibility of plans and to explore "what if" scenarios. Previous methods have employed commercially available discrete event simulation packages (e.g. Planimate [22]) which have allowed complex models of maintenance networks to be developed. These models lack transparency however, which makes external review difficult.

The Timed CPN model captures the impact on operational availability of a deployed system of equipment as a result of, among other things, the composition and disposition of personnel, and equipment usage. In addition, the model accounts for the dynamic and networked characteristics of the system. The current model has been derived from an initial prototype developed during 2006. Originally this model was designed to assess the capability of the maintenance process, focussing on how personnel are utilised. A more complete prototype model has been developed in 2007.

Although the model captured the maintenance process

well, simulation of ‘realistic’ scenarios with the Timed CPN simulator of CPN Tools [6, 15] revealed serious simulation performance concerns and consequently necessitated model implementation and design reviews. The performance concerns have primarily involved the representation of personnel and their associated use within the model. Whilst the present model represents a military maintenance process, the issues raised here may equally apply to any number of similar logistics models.

CPNs and other high-level Petri nets have long been applied to the verification of complex distributed systems such as computer networks and protocols [2]. In the area of defence, CPNs have been applied to the modelling of defence logistics networks [10, 11], the modelling and analysis of operations planning and the production of a prototype tool based on CPNs [17, 26], and modelling and analysis of the Australian Defence Force planning process [20].

There have been numerous previous attempts to model and analyse Defence maintenance processes, e.g. [3, 12, 13, 18, 19, 23]. More generally, there are models of the maintenance of fleets of aircraft or vehicles [8], maintenance of power systems [9], models of specific maintenance facilities [24] and models that schedule maintenance activities to fit around the main activities of equipment [1, 4], e.g. equipment in a production line, or military vehicles performing missions. There are also models that examine key equipment items through a maintenance system perspective [5, 21]. Some models [12, 23] make use of the discrete event simulation tool ARENA [16], while others use Bayesian and Markovian analysis techniques [4, 8, 13] or optimisation of mixed-integer linear programming models [1]. A genetic algorithm is used in [19] to optimise maintenance schedules. However, our CPN model is more extensive, flexible and general than any we have encountered thus far in the literature.

In this paper we begin by presenting our CPN model of an Army’s maintenance process. CPN Tools [6] has been used to create and simulate the CPN model. It models a user-specified distributed network of maintenance locations, rather than a single maintenance facility or location, and is not specific to any single piece or type of equipment. Personnel are explicitly modelled and are not assumed to have homogeneous skills. Of course, modelling in greater fidelity requires greater computational resources.

In order for the model to be useful as a simulation tool, it is necessary for it to achieve acceptable levels of run-time performance with the CPN Tools simulator. This paper documents our investigations into the simulation performance of the CPN Tools simulator in the context of this model, including our investigation and evaluation of alternative ways to model personnel, in an attempt to achieve an acceptable level of simulation performance. In this case, acceptable refers to minutes, not hours, of calculation time.

The contributions of this paper are threefold. Firstly, this paper introduces and describes our CPN model of the Army’s maintenance process at a high level. Secondly, we present a number of different ways to represent personnel within our CPN model. Thirdly, we provide a comparison and evaluation of aspects of CPN Tools’ simulation performance of these different modelling approaches.

The rest of this paper is organised as follows. Section 2 provides a description of our maintenance system as well as providing a top-level introduction to our model. Choices for modelling personnel are described in Section 3. The simula-

tion performance of these different models is presented and discussed in Section 4 along with a discussion of future directions. Finally, Section 5 provides some concluding remarks. It is assumed that the reader has basic familiarity with CPN concepts, although a deep understanding is not essential to understand the problems presented in this paper. (For a detailed introduction to CPNs, see [14, 15].)

## 2. MODEL DESCRIPTION

The CPN maintenance model is a hierarchical model, representing a decomposition of the processes related to performing maintenance in a military environment. The higher level pages of the hierarchical model serve as a flowchart of the process. We use these pages to describe the system and processes we are modelling. Size constraints prevent us from describing the model in detail.

The model consists of 15 pages arranged into three hierarchical levels, plus an additional layer required to initialise the model with a specific scenario. The model comprises 44 executable transitions, 13 substitution transitions (which create the hierarchical structure), 6 fusion places (places shared between pages regardless of hierarchical relationship) and 21 non-fusion places (counting places shared due to the hierarchical structure only once). Further complexity is contained in extensive model inscriptions and function definitions, written in the programming language Standard ML [25], and comprising thousands of lines of code.

### 2.1 The System Level Page

The System Level page, shown in Fig. 1, forms an overview of the operation of the key aspects of the maintenance system. This page consists mostly of substitution transitions (double-outlined rectangles), each of which represents a sub-process within the overall maintenance system. This net structure has been developed to explicitly represent the flow of equipment through the maintenance process, illustrated by the bold arcs in Fig. 1. Note that this net structure represents the maintenance process at an arbitrary number of separate locations, by encoding location information within tokens (each place is typed by a *colour set* that specifies the allowable data values, so-called *coloured tokens*, that can mark the place). We now describe the processes represented on the System Level page.

#### 2.1.1 Liability Generation

The starting point in the execution of our model is the assignment of a maintenance liability (specific details of a maintenance activity) to each of the items of equipment in the Equipment Awaiting Liability Assignment place by the Liability Generation page. Each item of equipment is paired with a “future” liability as it is transferred to the Operational Equipment place, where it is considered to be in use. These item-liability pairs remain static until such time as their maintenance liability becomes due. At this point, the Maintenance Occurrence transition moves the item of equipment from an ‘operational’ state (Operational Equipment place) to an ‘offline’ state (Equipment With Outstanding Maintenance Liability place), where the item awaits maintenance.

#### 2.1.2 Determine Transport Resources

The first step in the maintenance process is to determine how a given liability will be handled, and what, if any, transportation requirements are necessary. The approaches are:

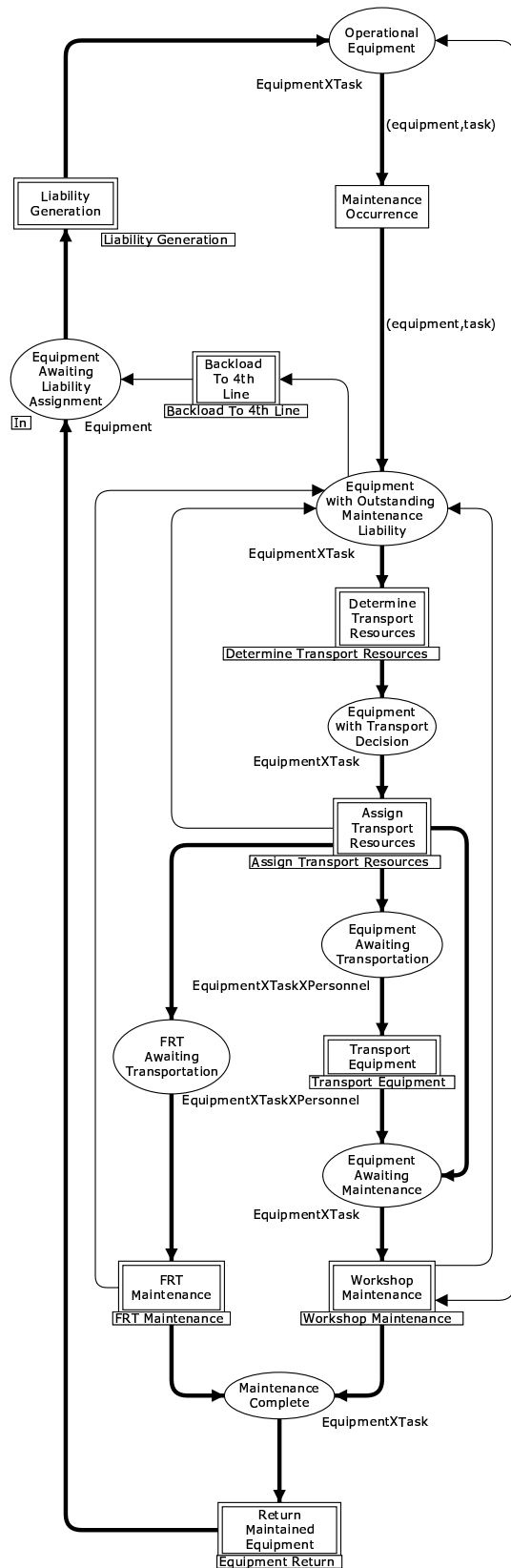


Figure 1: The System Level page, showing the high-level structure and flow of equipment (in bold) of the Maintenance CPN Model.

**In-situ:** maintain the equipment where it is;

**Self-transport:** the equipment moves itself to a suitable maintenance workshop;

**Distribution:** the equipment is moved using a more general distribution network, e.g. road or railway, that services the demands of the military operation;

**Forward Repair Team (FRT):** a team of personnel move from a maintenance workshop to the location of the equipment in order to effect the necessary maintenance; and

**Recovery:** a Recovery team moves from a maintenance workshop to the location of the equipment and brings the equipment back to the maintenance workshop.

This decision is handled by the Determine Transport Resources subpage. All methods of dealing with a liability necessitate the use of a form of transportation with the exception of in situ maintenance.

### 2.1.3 Assign Transport Resources

Once the transport requirements have been identified, the Assign Transport Resources subpage ensures that any personnel requirements are capable of being fulfilled by the system, and in turn commencing the necessary transportation action. In the event that required personnel are not available, the equipment is passed back to the Determine Transport Resources page (the arc from Assign Transport Resources to Equipment with Outstanding Maintenance Liability) and an alternative approach is selected.

### 2.1.4 Transport Equipment

Assuming all personnel requirements are satisfied, there are three possible outcomes (the three bold arcs that exit the Assign Transport Resources page). In the case of in-situ maintenance, no transportation is necessary, so the equipment item is passed directly to the Equipment Awaiting Maintenance place. Alternatively, if a Forward Repair Team is required (defined by the type of item and location) the equipment with its maintenance liability is passed to the FRT Awaiting Transportation place along with the personnel that make up the Forward Repair Team. The FRT travels to the equipment to maintain it, before returning to its assigned workshop and disbanding. Lastly, an equipment item may need to travel either rearwards or forwards within the maintenance network. The equipment may not be able to transport itself, and may therefore also require the assistance of a Recovery Team to do so. These scenarios are handled by the Transport Equipment subpage, where a recovery team is similarly created for this specific action. Once all transportation is taken care of, equipment items will enter the Workshop Maintenance process and any personnel which were required to assist with the transportation are again released to conduct other work.

### 2.1.5 FRT Maintenance and Workshop Maintenance

Both the FRT and Workshop Maintenance processes address the outstanding maintenance liability and result in a maintained item of equipment. In the case of Workshop Maintenance, a replacement item can be ordered while the item with the outstanding maintenance liability undergoes maintenance. When this happens, an item of equipment

is drawn from an equipment pool (taken from the Operational Equipment place) and transported to the location of the original item. In the case of FRT Maintenance, it is possible that FRT maintenance will fail to successfully maintain an item of equipment. This results in the item of equipment passing through the Determine Transport Resources subpage once more, where an alternative maintenance approach will be chosen and subsequently attempted.

On entering a workshop, equipment is added automatically to a prioritised queue. Tradespeople are automatically assigned to items on a first-come-first-served basis for their individual trade, with respect to the priority level of the items. Once assigned they continue to work on the repairs (identified in the liability pairing) until they either complete the maintenance task or are required to rest. The Workshop Maintenance page is examined in more detail in Section 2.2.

### 2.1.6 Backload to 4th Line

In the case of either method of maintenance, if an initial technical inspection reveals that an item of equipment is beyond the level of repair available in the field, the item of equipment is effectively removed from the area of operations, typically back to Australia or the 4th line of supply - this process is denoted here as “Backload to 4th Line”, and is modelled by the subpage of the same name.

### 2.1.7 Return Maintained Equipment

Subject to the successful completion of the Workshop Maintenance or FRT Maintenance process, the equipment item will be placed in the Maintenance Complete place, and the personnel will then be made available for either another job, or moved into an off-line or resting state. The Return Maintained Equipment process will return the equipment to its original location and deposit the equipment once more into the Equipment Awaiting Liability Assignment place. This permits the item to return to an operational state, ready for the next maintenance liability to occur and enter the maintenance process once more.

## 2.2 The Workshop Maintenance Page

The primary input to the Workshop Maintenance page (shown in Fig. 2) is the Equipment Awaiting Maintenance place, containing equipment items (with their associated liability description) that are located at a maintenance workshop. The Workshop Maintenance page has four sequential sub processes (the four substitution transitions).

### 2.2.1 Technical Inspection

This process represents the inspection of equipment and reveals the previously hidden liability to the maintenance workshop.

### 2.2.2 Inspection Decision

The subsequent Inspection Decision process determines the suitability of the workshop to address the equipment liability. This process includes any decision to replace the equipment from a replacement pool, the capabilities of the workshop to meet the grade of repair, and finally the appropriately skilled tradespersons to undertake the specific repairs.

Next, the equipment item’s maintenance liability data is compared with the capabilities of the maintenance workshop. If the workshop’s capabilities are not sufficient to perform the corresponding maintenance activities, the equip-

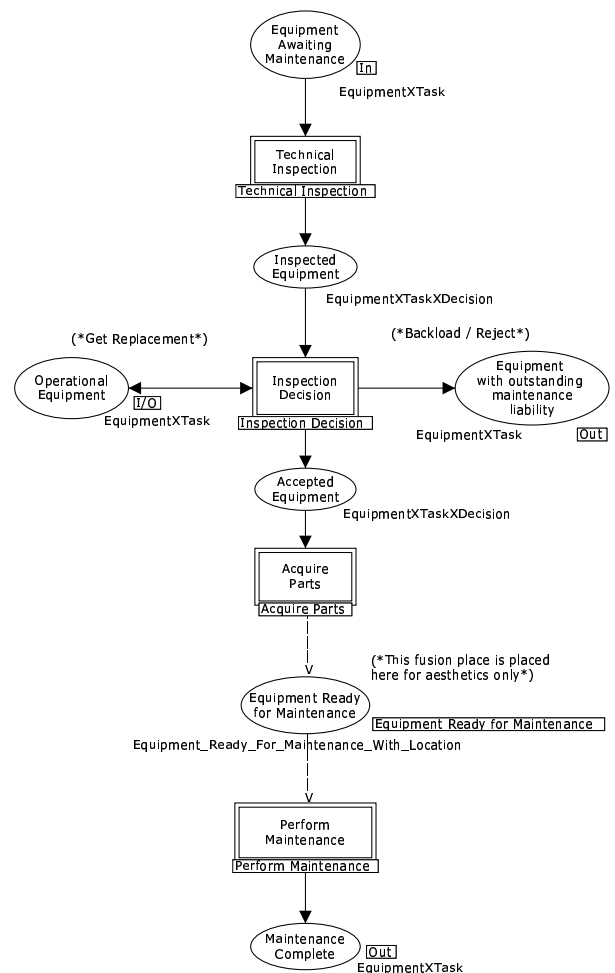


Figure 2: The Workshop Maintenance page, showing an overview of the workshop maintenance process.

ment item is returned to the Equipment with Outstanding Maintenance Liability place, so that it may undertake another iteration of the Determine Transport Resources process described in Section 2.1, including being backloaded to another workshop. This continues until either the equipment is maintained or backloaded to 4th line. If this workshop can handle the necessary maintenance activities, the equipment item progresses to the Acquire Parts process.

### 2.2.3 Acquire Parts

The Acquire Parts process represents the delays inherent when spares are required but not presently in stock. Equipment items that have had their parts sourced, and are awaiting the attention of maintenance personnel, are inserted into a prioritised queue which resides on the Equipment Ready for Maintenance place. There is one such queue for each maintenance workshop, representing the outstanding maintenance activities at each location.

### 2.2.4 Perform Maintenance

The Perform Maintenance subpage assigns personnel according to maintenance liabilities. In particular, personnel are assigned to the highest priority maintenance tasks at their location. Personnel remain assigned until they are ei-

### Listing 1: The Equipment Colour Set.

```
1 colset Operator_Type = with Operator | Pool;
2
3 colset Service_Types = with Major | Minor;
4
5 colset Equipment = record
6   equipment_type: STRING *
7   present_location: STRING *
8   home_location: STRING *
9   operator_type: Operator_Type *
10  last_service_type : Service_Types *
11  odometer: INT *
12  time_last_service:INT *
13  odometer_last_service: INT *
14  time_last_inspection:INT
15 timed;
```

ther no longer required (determined by the maintenance liability) or alternatively have to stop working e.g. to sleep. The completion of all required maintenance activities for a given item of equipment signals the end of the Perform Maintenance process, at which point the item of equipment is placed in the Maintenance Complete place, and the last tradesperson working on this activity becomes available.

## 2.3 The Model Initialisation Page

The Model Initialisation page is used to populate the model tokens representing the scenario under consideration. This includes: the topology of maintenance workshops (nodes); characteristics and capabilities of each maintenance workshop; the number and disposition of personnel distributed throughout the maintenance workshops; the equipment ready to be assigned a maintenance liability; the equipment initially waiting for parts at each workshop; and the equipment ready to be maintained at each workshop.

The population of the above places is achieved through the use of functions that read in initialisation data from text files on local storage. The input files may be populated by an external editor, e.g. from an Excel spreadsheet, thus allowing for increased modelling flexibility and expedient scenario generation and modification.

## 2.4 Important Place Types

There are three key colour sets (place types) used throughout the model. These are the **Equipment**, **Maintenance\_Task** and **Personnel** colour sets.

### 2.4.1 Equipment Colour Set

The **Equipment** colour set is shown on lines 5-15 of Listing 1. It defines a record structure that describes individual items of equipment that will require maintenance. The equipment type, present location and home location (lines 6-8) are specified as strings (rather than enumerated types) for extensibility. Equipment will either have an operator (i.e. in use) or will be in an equipment pool (line 1) and this is designated by the operator type (line 9). The last service type (line 10) records whether the last service was a major service or a minor service (line 3). Finally, the current odometer reading, time of last service, odometer reading at the last service and time of the last inspection (lines 11-14) are recorded as integers. Naturally, the odometer fields can be representative of other usage metrics as appropriate to the equipment item in question, e.g. operating hours for a generator.

### Listing 2: The Maintenance\_Task Colour Set.

```
1 colset Maintenance_Type =
2   with Service | Corrective | Inspection;
3
4 colset Grade = with L|M|H;
5
6 colset Trade = STRING;
7 colset ERT = INT;
8 colset Job = product Trade * ERT;
9 colset Job_List = list Job;
10
11 colset Priority = with Essential | Non_Essential;
12
13 colset Mobility =
14   with wheeled_mobile | wheeled_not_mobile
15     | not_wheeled;
16
17 colset Maint_Methods =
18   with FRT | SelfTransport | Distribution | InSitu
19     | Recovery;
20 colset Maint_Methods_Attempted = list Maint_Methods;
21
22 colset Maintenance_Task = record
23   maintenance_type: Maintenance_Type *
24   next_occurrence_time: INT *
25   current_request_location: STRING *
26   grade_required: Grade *
27   job_requirements: Job_List *
28   job_priority: Priority *
29   mobility: Mobility*
30   inspected: BOOL *
31   methods_attempted: Maint_Methods_Attempted *
32   assignment_timeout: INT
33 timed;
```

### 2.4.2 Maintenance Task Colour Set

The **Maintenance\_Task** colour set is shown in Listing 2. This is also a record (lines 22-33) that specifies the maintenance required and records the progression of individual items through the system. The type of maintenance required (line 23) can be either a regular service, corrective maintenance (in the event of a breakdown) or an inspection of a piece of equipment (lines 1-2). The next occurrence time (line 24) records when the next maintenance event is due to occur. The location of the entity that is currently considering the maintenance task is given by line 25. The grade of repair (line 26) required for a particular maintenance task will be either Low, Medium or Heavy (line 4) depending on the nature of the task, and hence where and by whom the item can be maintained. The job requirements (line 27) specify the tradespeople required and the length of time for which they will be required (lines 8 and 9). The trade types, e.g. vehicle mechanic, are specified using strings for extensibility (line 6). The length of time a particular trade is required for is given by the Estimated Repair Time (ERT) on line 7. The priority of each particular maintenance task (line 28) is categorised as either essential or non-essential (line 11) and is based on the type of equipment. Line 29 specifies the mobility of the equipment to be maintained. From lines 13-15, the equipment can be either wheeled and mobile (e.g. a truck that can be driven), wheeled and not mobile (e.g. a truck with a broken engine) or not wheeled (e.g. a generator). Whether the equipment has been inspected with respect to the current maintenance task is given by line 30. The *methods of maintenance* attempted (line 31) specifies which methods of maintenance have been attempted to address this maintenance task. The methods are given by lines

### Listing 3: The Personnel Colour Set.

```
1 colset Personnel_States =
2   with Working | Ready | Offline;
3
4 colset Personnel = record
5   trade: Trade *
6   home_location: STRING *
7   working_status: Personnel_States *
8   last_came_online_time: INT
9   timed;
10
11 colset Personnel_List = list Personnel timed;
```

17-20, i.e. in-situ, self-transport, distribution, FRT and Recovery. The assignment timeout (line 32) specifies how long a task will wait for personnel to be assigned to it before some other alternative method is attempted.

#### 2.4.3 Personnel Colour Set

The `Personnel` colour set, given in Listing 3, is used to record details of individual personnel. This includes details such as their trade (line 5) and home location (line 6). The working status (line 7) specifies whether a person is ready to work, currently working, or *offline* (lines 1-2). Offline means that the person is not currently available to be assigned to perform maintenance work (e.g. is sleeping or performing other duties). The time at which the person last came online is recorded by line 8. A list of personnel is given on line 11.

## 3. MODELLING PERSONNEL

Personnel as an enabling resource drive the flow of maintenance throughout the system. As such, the availability of tradespeople is a critical element of the model that influences 12 executable transitions across 7 model pages. The `Personnel Management` page is responsible only for moving personnel from a ready state to an offline state, and vice versa, at the correct times. Due to the interconnection of personnel, changes in the state of the `Personnel` place can alter the behaviour system-wide. This personnel resource dependency across many facets of the system behaviour induces significant computational effort, thus has the potential to adversely affect the performance of the simulation.

The performance issues reported in Section 4 have led us to consider two ways of modelling personnel: as individual tokens; and in lists.

### 3.1 Personnel Modelled as Tokens (Model 1)

Our first model represents personnel as individual tokens with timestamps (a time value that gives the earliest model time a token can be removed from its place by a transition). However, two difficulties were encountered: the composition of the recovery team or FRT may not be known; and the number of personnel required may be different for different maintenance tasks. This adds significant complexity to the problem of selecting personnel, as there is no easy way to specify a varying number of partially known data values on an input arc using CPNs. Because of this a workaround was implemented, in which a fixed number of personnel are selected, checked against what is required, and the excess personnel returned to the `Personnel` place. This works, provided the fixed number of personnel selected is always greater than or equal to the number of personnel that are actually required, and that the `Personnel` place always has at least this

fixed number of personnel tokens in it. This baseline model of personnel is known as Model 1.

Figure 3 shows this workaround. It depicts a fragment of the `Assign Transport Resources` page, showing only the `Assign Recovery Assets` transition (i.e. only the assignment of personnel to a recovery team). We do not attempt to describe this page in detail, however, here we see that 6 people are removed (6 tokens on the arc from `Personnel` to `Assign Recovery Assets`). The guard expression (a boolean expression that must evaluate to true for the transition to fire) ensures that the trade of the personnel removed either match what is needed, are any trade (specified by the trade ‘Whatever’) or are dummy personnel (excess personnel specified by the trade ‘Dummy’, which are not utilised). When the transition fires the code segment is executed, which splits the personnel into those required (arc from `Assign Recovery Assets` to `Equipment Awaiting Transportation`, which also changes the state of the personnel to `Working`) and those to be returned to the personnel place (arc from `Assign Recovery Assets` to `Personnel`). The ‘@’ character in some of the arc inscriptions is used to manipulate the timestamps of tokens. It is possible to express the code segment and guard more concisely using list manipulation functions, however the current approach represents a trade-off between conciseness and clarity.

### 3.2 Personnel Modelled in a List

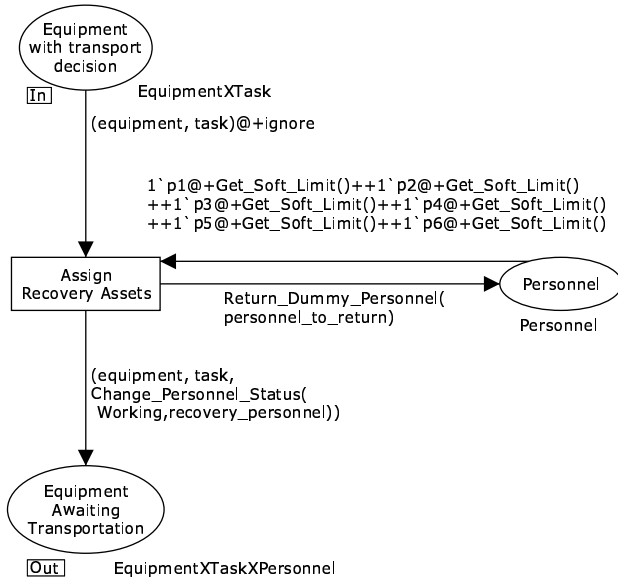
To avoid the workaround mentioned above, rather than modelling personnel as a multiset of tokens (one token per person), an alternative was to model personnel in a list. To do this, each personnel ‘token’ was augmented with a time value modelled as part of the colour set. This time value is used to store the timestamp that the personnel value would have had, if it were a token.

A multiset does not impose an ordering on its elements. Conversely, by its nature, a list does. Two possibilities were considered when ordering the list of personnel. The first (denoted by variant A) was to order the personnel values in the list firstly according to their time values and secondly according to a ‘less-than’ function defined automatically by CPN Tools for the `Personnel` colour set (to impose an order on personnel with the same time values). The second (denoted by variant B) was to order the personnel in the list according to time value only. Both these choices result in the earliest available person being at the head of the list, hence searching for available personnel in the list becomes at worst a linear operation (it is a constant operation if we only wish to find a single available person, regardless of trade, as may occur with inspections).

#### 3.2.1 Retaining Nondeterministic Choice (Model 2)

One of the features of modelling personnel as individual tokens in a CPN is that there is a nondeterministic choice between eligible personnel to cycle offline or online, or to be assigned to tasks. To retain this nondeterministic choice, we modelled a procedure that:

- splits all eligible personnel off the front of the list, based on their time values (eligible personnel will always be at the front of the list due to the ordering that we impose);
- converts these personnel values into equivalent timed personnel tokens to obtain a multiset of all eligible personnel;



```
[ (* Guard for Assign Recovery Assets *)
List.hd(#methods_attempted task) = Recovery andalso
let val jr = List.map(fn(trade,ert) => trade)
  (Format_Recovery_Personnel(Recovery_Personnel()))
in
  (List.nth(jr,0) = #trade p1
  or else (List.nth(jr,0) = Whatever andalso #trade p1 <> Dummy))
andalso (if ((#trade p1) <> Dummy)
  then (#home_location p1)=(#current_request_location task) else true)
andalso (#working_status p1) = Ready
andalso (List.nth(jr,1) = #trade p2
  or else (List.nth(jr,1) = Whatever andalso #trade p2 <> Dummy))
andalso (if ((#trade p2) <> Dummy)
  then (#home_location p2)=(#current_request_location task) else true)
andalso (#working_status p2) = Ready
andalso (List.nth(jr,2) = #trade p3
  or else (List.nth(jr,2) = Whatever andalso #trade p3 <> Dummy))
andalso (if ((#trade p3) <> Dummy)
  then (#home_location p3)=(#current_request_location task) else true)
andalso (#working_status p3) = Ready
andalso (List.nth(jr,3) = #trade p4
  or else (List.nth(jr,3) = Whatever andalso #trade p4 <> Dummy))
andalso (if ((#trade p4) <> Dummy)
  then (#home_location p4)=(#current_request_location task) else true)
andalso (#working_status p4) = Ready
andalso (List.nth(jr,4) = #trade p5
  or else (List.nth(jr,4) = Whatever andalso #trade p5 <> Dummy))
andalso (if ((#trade p5) <> Dummy)
  then (#home_location p5)=(#current_request_location task) else true)
andalso (#working_status p5) = Ready
andalso (List.nth(jr,5) = #trade p6
  or else (List.nth(jr,5) = Whatever andalso #trade p6 <> Dummy))
andalso (if ((#trade p6) <> Dummy)
  then (#home_location p6)=(#current_request_location task) else true)
andalso (#working_status p6) = Ready
end]
```

```
(* Code segment for Assign Recovery Assets *)
input (p1,p2,p3,p4,p5,p6);
output (recovery_personnel,personnel_to_return);
action
  ((if((#trade p1) <> Dummy) then [p1] else []) ^^
  (if((#trade p2) <> Dummy) then [p2] else []) ^^
  (if((#trade p3) <> Dummy) then [p3] else []) ^^
  (if((#trade p4) <> Dummy) then [p4] else []) ^^
  (if((#trade p5) <> Dummy) then [p5] else []) ^^
  (if((#trade p6) <> Dummy) then [p6] else []),
  (if((#trade p1) = Dummy) then [p1] else []) ^^
  (if((#trade p2) = Dummy) then [p2] else []) ^^
  (if((#trade p3) = Dummy) then [p3] else []) ^^
  (if((#trade p4) = Dummy) then [p4] else []) ^^
  (if((#trade p5) = Dummy) then [p5] else []) ^^
  (if((#trade p6) = Dummy) then [p6] else []));
```

Figure 3: A fragment of the Assign Transport Resources page of Model 1.

- selects nondeterministically from these personnel; and
- uses a list representation of the eligible personnel to clean up the rest of the personnel (in the absence of reset arcs) and re-inserts the remaining eligible personnel back into the personnel list.

Both the cycling of personnel and the assignment of personnel to tasks is modelled in a similar way, although the assignment of personnel to tasks was significantly more complex, as more than one person may be selected. Due to their size and complexity, the pages that model this behaviour are excluded from this paper due to size constraints.

We refer to the two models of personnel in lists in which nondeterministic choice is retained, with the two variants (A and B) as Model 2A and Model 2B respectively.

### 3.2.2 Removing Nondeterministic Choice (Models 3 and 4)

The degree of nondeterministic choice exhibited by models 2A and 2B may not be necessary for the desired outcomes of the model. For example, if 100 people are due to go offline at a specific time, we do not require that the order in which these 100 people go offline is chosen nondeterministically, as they will all move offline at the same model time. It is sufficient for our purposes that one specific order is selected and followed, e.g. the order in which the personnel appear in the list. We created four models of personnel in lists without nondeterministic choice for cycling offline and online:

- Models 3A and 3B are models in which the person to move offline or online is the head of the list (with variants A and B).
- Models 4A and 4B refer to models in which all personnel eligible to move offline or online are moved in one action (again with variants A and B).

Hence, models 3A and 4A do not differ with respect to the selection of personnel to be assigned to tasks. This remains nondeterministic, due to the fact that the appropriate person may not always be the person at the head of the list (e.g. if a specific trade is required). This is also true of models 3B and 4B.

## 4. SIMULATION PERFORMANCE

Personnel are involved in most aspects of the CPN model in one way or another. As already mentioned, two of the key involvements of personnel are the cycling of personnel online and offline and the assigning of personnel to a Recovery Team or a Forward Repair Team. In this section, the simulation performance of all seven models with respect to these two activities is evaluated and discussed. All experiments have been carried out with the Timed CPN Simulator of CPN Tools version 2.2.0 on a 2GHz Intel Core 2 Duo processor with 2Gb of memory.

### 4.1 Initial Investigation

Model 1, where personnel are modelled as tokens, was found to perform reasonably well on small scenarios (10's of people) when cycling personnel online and offline, but performed very badly when assigning personnel to teams, due to the nature of the assignment and the workaround, as discussed in Section 3.1. Two transitions were found to be responsible for this: the Assign Recovery Assets and Assign

FRT transitions on the Assign Transport Resources subpage. We discovered that the ‘less than’ function for the Personnel colour set was being executed of the order of  $10^7$  times, even for small scenarios, every time the enabling of these two transitions was checked. Because the Personnel place is connected to these two transitions (amongst others), every time the marking of the Personnel place changes, a check for the enabling of these two transitions is instigated. This resulted in a four to five second delay between successive steps of the simulation, where the delay was divided evenly between these two transitions. Hence we were able to determine that the time was being taken in the enabling check subsequent to the firing of a transition that affected the marking of the Personnel place, rather than in performing the act of firing the transition itself.

Informal testing of the list models of personnel revealed that the assignment of personnel to teams appeared to perform well, at least for small examples, but as we shall see the cycling of personnel online and offline was not always better than the model of personnel as tokens.

## 4.2 Detailed Performance Testing

Three areas of concern with respect to performance were identified for more formal testing, as discussed below.

### 4.2.1 Personnel Cycling Offline and Online

This test examined the performance of cycling personnel online and offline while changing the number of personnel introduced into the system initially. Simulations were run for 20 days of modelled time, whilst recording the real duration of the simulation. A scenario of three maintenance workshops was considered, where each workshop was assigned personnel of three different trades, except for the third workshop, which had four trades. This gives a base of 10 personnel in the system (one of each trade at each workshop). This base scenario was then linearly scaled to investigate performance. Each scenario was run a number of times, and the results averaged, with little variation observed between the separate runs.

Figure 4 shows a graph of the real time taken for each run as the number of personnel increases, for each of the seven models. Models 2A, 2B, 3A and 3B all perform worse than Model 1, with models 2A and 2B performing the worst of all. This is likely to be caused by the additional manipulation of lists and tokens required in these models. Particularly, the additional processing required to retain nondeterministic choice for cycling personnel offline and online in models 2A and 2B introduces a significant overhead. This also explains why models 3A and 3B outperform models 2A and 2B. Little difference can be seen in the performance of the two list ordering schemes (variant A versus variant B).

Models 4A and 4B both significantly outperform Model 1 (and all other models). One reason is the ease with which an enabled binding element can be found for the transitions that cycle personnel online and offline. There is only one list of personnel that are online, and one list of personnel that are offline, instead of a (potentially) large multiset of personnel from which to select. Another reason (with respect to models 2A and 2B) is that the choice of personnel to cycle is no longer nondeterministic. Models 4A and 4B outperform models 3A and 3B because the transitions to cycle personnel online and offline move all eligible personnel in the same action, hence there are less enabling checks, less

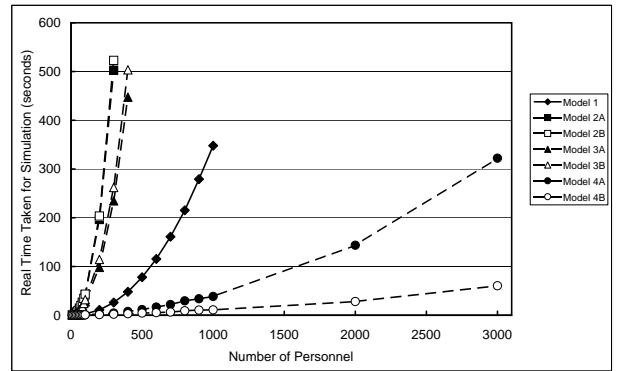


Figure 4: Performance of Personnel Cycling Offline and Online as No. of Personnel Increases.

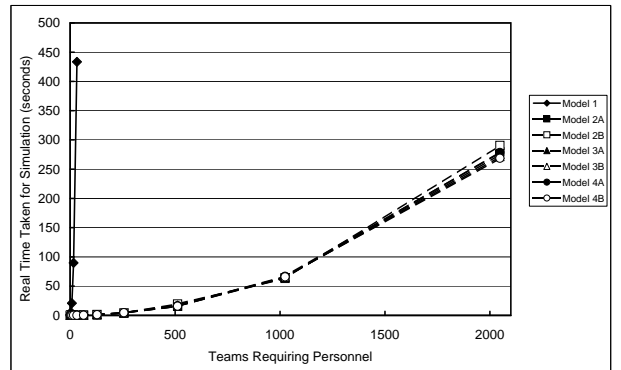


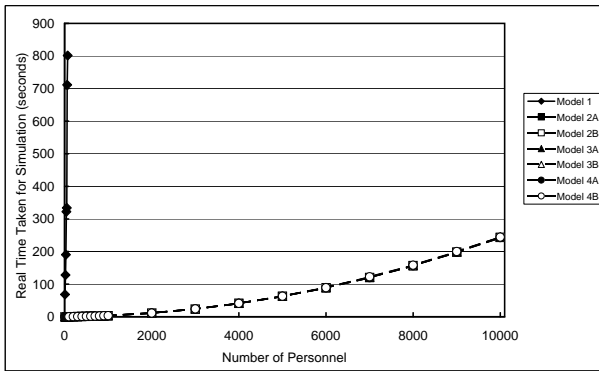
Figure 5: Performance of Assigning Personnel to Teams as No. of Teams Increases.

transition occurrences and less list manipulation. It should be noted however that performance will deteriorate if the number of personnel eligible to move offline or online at the same time decreases, e.g. as personnel times become increasingly staggered, there will be greater numbers of transition occurrences involving smaller numbers of personnel in each occurrence. The worst case performance would be equivalent to that of models 3A and 3B.

### 4.2.2 Assigning Personnel to Teams

The first of two performance tests for assigning personnel to teams keeps the number of personnel fixed but increases the number of teams that simultaneously require personnel. The results are presented for Recovery teams, however the results for assigning personnel to an FRT are nearly identical, and hence are not presented in this paper. In order for the supply of available personnel not to become exhausted before all teams are allocated, personnel assigned to a team are instantly recycled back into the Personnel place, for assignment to another team. The scenario considered is the same as in Section 4.2.1 with 10 personnel distributed over 3 workshops. The model was executed until all teams were assigned. Figure 5 shows a graph of the result.

In this situation, all models of personnel using lists outperform Model 1, and furthermore they all perform approximately the same. This is to be expected, as all 6 models of personnel using lists still employ a nondeterministic choice for the selection of personnel. Interestingly, whereas nonde-



**Figure 6: Performance of Assigning Personnel to Teams as No. of Personnel Increases.**

terministic choice (models 2A and 2B) perform much worse than model 1 for cycling personnel offline and online, in this test we see that the nondeterministic choice performs better than model 1. This is a measure of how badly the workaround employed in model 1 (for assigning a group of personnel with unknown composition and varying size) actually performs.

The second of the two tests for assigning personnel to teams keeps the number of teams requiring personnel fixed, but increases the number of personnel. Five teams were used as the set of teams requiring personnel, and personnel numbers and distribution were as specified in Section 4.2.1. A graph of this is shown in Fig. 6.

Again, we see that all of the models of personnel using lists drastically outperform the baseline model of personnel as tokens (model 1). The difference is even more acute than in Fig. 5. As before, all of the models of personnel using lists perform nearly identically, because the action of assigning personnel to teams is not affected by the mechanism by which personnel are cycled offline and online. It would appear that the ordering of the list does not have any significant impact on the performance of the simulator (variant A versus variant B).

### 4.3 Overall Performance and Discussion

The simulation performance of the prototype CPN model when simulated with CPN Tools was unacceptable to the extent that it became a topic for investigation in its own right. A significant performance impost was found to be related to the way in which personnel were modelled. In particular, changing the state of places within a CPN requires additional checking for enabled binding elements of all transitions directly connected to that place. In a large workforce model, where we have potentially complex (and to some extent arbitrary) interactions of personnel and tasks, this can result in an explosion in the number of calculations. In this paper we have explored alternative models of personnel in order to alleviate some of these additional checks. Modeling personnel as individual tokens is a highly inefficient method - with a list-based solution being far more computationally expedient without losing much of the desired behaviour of the token-based solution.

Likewise it is also likely that the nature of using a “generic” CPN engine, designed to handle any number of arbitrarily defined data types, is inherently inefficient when compared to customized engines. Development of such customized en-

gines is non-trivial and expensive, and we have therefore attempted to constrain solutions to CPN models executed from within CPN Tools. Whilst the examined models have shown a significant improvement in performance, we believe that further advances can be made, and we outline a number of options to be explored in the future.

The first step is to select the best performing model (model 4B) and investigate overall model performance for varying levels of staff. Further performance hurdles may be encountered once a complete scenario is examined, either emerging from operating the model as a whole, or in parts of the model that have not yet undergone performance profiling.

Another avenue for future investigation is to consider a higher level of abstraction, while still being able to answer questions about personnel. We propose to investigate a radical change to the way personnel are modelled. To do this, we shall implicitly model the change of state from online to offline and vice versa, rather than explicitly using transition occurrences and changes to the marking of the net. Because the cycling of personnel is a deterministic process (in the absence of maintenance tasks) it follows that by knowing the most recent maintenance activity of all personnel, all future availabilities (until the next maintenance activity starts) can be derived. Therefore, explicit modelling of the transitions of personnel online and offline may not be necessary.

Another, less radical, method for attacking the current performance issues could be to identify areas in the model where nondeterministic choice remains prevalent but which does not add much value to the model in terms of achieving its objectives. Such nondeterministic choice may be able to be removed, which may provide a speed-up in the simulation of the model. This may go hand-in-hand with small increases in the level of abstraction. Non-determinism in this case may be replaced by heuristics to approximate some aspects of the maintenance process.

## 5. CONCLUSIONS

This paper has presented, at a high level, a prototype Coloured Petri Net model of the Australian Army’s maintenance process, developed as part of a collaborative project between the Computer Systems Engineering Centre of the University of South Australia and the Logistics Mission of Australia’s Defence Science and Technology Organisation. This paper has also presented our concerns regarding simulation performance of the model when simulated with the Timed CPN simulator of CPN Tools, and our attempts to improve the performance to a level acceptable for timely evaluations of scenarios. We have made some significant performance gains over our original baseline (model 1). However, for realistically sized scenarios (hundreds of personnel and thousands of pieces of equipment over tens of locations) there is still a long way to go before the model performs fast enough to be useful.

We believe that these performance issues are not unique to our particular model, and will likely manifest themselves under any large (industrial-size) logistics model. As such, we believe that a more fundamental exploration of the nature of these performance constraints will benefit users of both CPN Tools and developers of large-scale CPN models in general. In particular we will now endeavor to identify the specific causal relationships between model, tool and scenario more clearly to identify broader rules for representing these complex systems effectively in CPN Tools.

## 6. ACKNOWLEDGMENTS

The authors would like to acknowledge their colleague, Dr. Nimrod Lilith, for support in developing some aspects of the model and for participation in many active discussions.

## 7. REFERENCES

- [1] J. Ashayeri, A. Teelen, and W. Selen. Production and maintenance planning model for the process industry. *International Journal of Production Research*, 34(12):3311–3326, 1996.
- [2] J. Billington, M. Diaz, and G. Rozenberg, editors. *Application of Petri Nets to Communication Networks*, volume 1605 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999.
- [3] L. Briskin and W. S. Demmy. An overview of the network repair level analysis model [military systems]. In *Proceedings of the IEEE 1988 National Aerospace and Electronics Conference (NAECON 1988)*, volume 4, pages 1414–1420, 1988.
- [4] C. R. Cassady, W. P. Murdock, and E. A. Pohl. Selective maintenance for support equipment involving multiple maintenance actions. *European Journal of Operational Research*, 129(2):252–258, 2001.
- [5] T. N. Cook and R. C. DiNicola. Modeling Combat Maintenance Operations. In *Proceedings of the Reliability and Maintainability Symposium*, pages 390–396, 1984.
- [6] CPN Tools. <http://wiki.daimi.au.dk/cpntools/cpntools.wiki>.
- [7] DSTO. *Coloured Petri Net Modelling of Defence Logistics*. Australian Government, Department of Defence, Contract No. 4500498737, 1 February 2006.
- [8] W. W. Fisher. Markov process modelling of a maintenance system with spares, repair, cannibalization and manpower constraints. *Mathematical and Computer Modelling*, 13(7):119–125, 1990.
- [9] O. Fouathia, J.-C. Maun, P.-E. Labeau, and D. Wiot. Stochastic approach using Petri nets for maintenance optimization in Belgian power systems. In *Proceedings of 8th International Conference on Probabilistic Methods Applied to Power Systems*, pages 168–173, 2004.
- [10] G. E. Gallasch, N. Lilith, and J. Billington. Extending a Coloured Petri Net Model of a Defence Logistics Network. Technical Report CSEC-29, Computer Systems Engineering Centre Report Series, University of South Australia, September 2007.
- [11] G. E. Gallasch, N. Lilith, J. Billington, L. Zhang, A. Bender, and B. Francis. Modelling Defence Logistics Networks. *International Journal on Software Tools for Technology Transfer, special section on CPN'06*, page 19 pages, 2007. Available via <http://dx.doi.org/10.1007/s10009-007-0052-z>.
- [12] T. Gossard, N. Brown, S. Powers, and D. Crippen. Scalable Integration Model for Objective Resource Capability Evaluations (SIM-FORCE). In *Winter Simulation Conference Proceedings*, volume 2, pages 1316–1323, 1999.
- [13] A. Jacopino, F. Groen, and A. Mosleh. Modelling imperfect inspection and maintenance in defence aviation through bayesian analysis of the KIJIMA type I general renewal process (GRP). In *Proceedings of Reliability and Maintainability Symposium (RAMS'06)*, pages 470–475, 2006.
- [14] K. Jensen. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Volumes 1 to 3, Basic Concepts, Analysis Methods, and Practical Use*. Monographs in Theoretical Computer Science. Springer-Verlag, 2nd edition, 1997.
- [15] K. Jensen, L. M. Kristensen, and L. Wells. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*, 9(3-4):213–254, June 2007.
- [16] W. D. Kelton, R. P. Sadowski, and D. T. Sturrock. *Simulation with Arena*. McGraw-Hill Higher Education, 3rd edition, 2004.
- [17] L. M. Kristensen, P. Mechlenborg, L. Zhang, B. Mitchell, and G. E. Gallasch. Model-based Development of a Course of Action Scheduling Tool. *International Journal on Software Tools for Technology Transfer, special section on CPN'06*, 2007. In Press.
- [18] O. Maimon and M. Last. Information-efficient design of an automatic aircraft maintenance supervisor. *Computers & Operations Research*, 20(4):421–434, 1993.
- [19] V. Mattila and K. Virtanen. A simulation-based optimization model to schedule periodic maintenance of a fleet of aircraft. In *Proceedings of European Simulation and Modelling Conference 2005 (ESM'2005)*, pages 479–483, 2005.
- [20] B. Mitchell, L. M. Kristensen, and L. Zhang. Formal Specification and State Space Analysis of an Operational Planning Process. In *Fifth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, pages 1–17. Department of Computer Science, University of Aarhus, 2004. Available via <http://www.daimi.au.dk/CPnets/workshop04/cpn/papers/>.
- [21] S. R. Parker. Combat Vehicle Reliability Assessment Simulation Model (CVRASM). In *Winter Simulation Conference Proceedings*, pages 491–498, 1991.
- [22] Planimate. <http://www.planimate.com>, 2007.
- [23] T. Raivio, E. Kuumola, V. A. Mattila, K. Virtanen, and R. P. Hamalainen. A simulation model for military aircraft maintenance and availability. In *Proceedings of Modelling and Simulation 2001, 15th European Simulation Multiconference (ESM'2001)*, pages 190–194, 2001.
- [24] M. Ramadass, J. Rosenberger, B. Huff, S. Gonterman, and R. N. Subramanian. Simulation Modelling for a Bus Maintenance Facility. In *Winter Simulation Conference Proceedings*, pages 1222–1225, 2004.
- [25] J. Ullman. *Elements of ML Programming*. Prentice-Hall, 2nd edition, 1998.
- [26] L. Zhang, L. M. Kristensen, B. Mitchell, G. E. Gallasch, P. Mechlenborg, and C. Janczura. COAST - An Operational Planning Tool for Course of Action Development and Analysis. In *Proceedings of the 9th International Command and Control Research and Technology Symposium (ICCRTS), Copenhagen, Denmark.*, 2004.