FORMAL SPECIFICATION AND ANALYSIS OF AIRBORNE MISSION SYSTEMS

Lars M. Kristensen and Jonathan Billington, School of Electrical and Information Engineering, University of South Australia, Adelaide, Australia.

Laure Petrucci, Laboratoire Spécification et Vérification, CNRS UMR 8643, Cachan, France.

Zahid H. Qureshi, Systems Sciences Laboratory, Defence Science and Technology Organisation, Adelaide, Australia.

Raymond Kiefer, RLM Systems Pty Ltd, Adelaide, Australia.

Abstract

The Airborne Mission System in military aircraft is a complex real time system consisting of a mission control computer, sensors, displays, controls, and data buses that provide interfaces between sub-systems. The complexity and real time requirements of Airborne Mission Systems represent major challenges to the Australian Defence Force during acquisition, upgrades, and maintenance. This has motivated research into the application of formal techniques to model the mission system architecture, and to investigate and predict the effects of avionics upgrades on the behaviour and performance of the integrated mission system. We are currently using Coloured Petri Nets to model and analyse Airborne Mission Systems. This paper presents two aspects of our ongoing research using Coloured Petri Nets. Firstly we present some analysis for a generic Airborne Mission system based on previously published model. Then we describe an initial application of our modelling framework to the AP-3C Orion Maritime Patrol Aircraft's Airborne Mission System.

1. Introduction

The Australian Defence Force regularly acquires and upgrades Airborne Mission Systems (AMS) but unfortunately many of these acquisitions are delayed [1]. These delays not only increase costs they can contribute to the loss of cutting edge capability due to the rapid advances in technology. In the following sub-sections we introduce the AMS, overview real time scheduling, describe briefly Coloured Petri Nets, identify key issues and provide an overview of this paper.

1.1 Airborne Mission Systems

An AMS architecture includes several sub-systems, for example, sensors, navigation and communication, displays and controls. They are typically connected together by a serial data bus. Although the behaviour of each sub-system in isolation may be well understood, their behaviour when combined is hard to predict due to the complex interaction and mission dependencies between the various sub-systems.

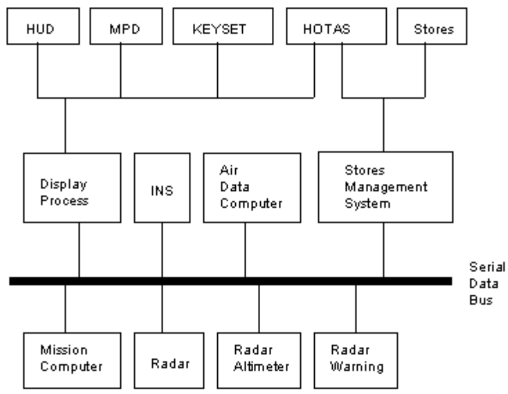

Figure 1. Generic Mission System Architecture

A generic architecture of an Avionics Mission System for a combat aircraft [2], depicted in Figure 1, consists of a mission control computer, a serial data bus, and various avionics sub-systems such as navigation, communication, radar and sensors. The control of devices, displays, and pilot controls, is handled by a collection of software tasks executing on a mission control computer. A major concern when upgrading and maintaining AMS is the scheduling of their real time tasks on the mission control computer, plus guaranteeing the data transfer across the mission data buses connecting the various sub-systems.

1.2 Real Time Scheduling

Scheduling of tasks in real time systems has traditionally been conducted using a purely algorithmic approach [3]. Recently, there has been an increasing interest in applying timed automata [4] and model checking techniques to scheduling problems [5, 6, 7]. The basic idea is to turn the scheduling problem into a reachability [8,16] problem that can be solved by analysis tools using a state space search. The advantages of formal modelling and state space methods in this setting is that the same model of the system can be used to analyse scheduling as well as other properties, such as functional correctness. Hence, it represents an integrated approach to the analysis of the system. Moreover, the simplifying assumptions required by the algorithmic approaches are not needed when applying state space methods. Petri Nets [8] have previously been used for modelling and analysis of real time systems, in particular for Ada programs [9, 10] and railway systems [11].

1.3 Coloured Petri Nets

Earlier work has mainly focused on deadlock detection and violations of safety properties. This paper presents our application of Coloured Petri Nets (CP-nets or CPNs) [12, 13] and the supporting Design/CPN tool [14] for the scheduling and input/output processing analysis in Avionics Mission Systems. This work is based on our previous modelling and analysis of AMS using CP-nets [15, 16].

Coloured Petri Nets [12, 13] are a graphically oriented modelling language for the design, specification, and verification of concurrent and distributed systems. CPNs are based on Petri Nets and the functional programming language Standard ML (SML) [17]. Petri Nets provide the primitives for modelling concurrency and synchronization, whereas SML provides the primitives for modelling data manipulation in systems and for creating compact and parameterisable CPN models. A CPN model of a system describes the states that the system may be in and the transitions between these states. CPN models are executable, which means that it is possible to investigate the behaviour of the system by simulations. CPN models can also be used for formal verification of systems based on state space analysis and model-checking [12]. CPN models can be hierarchically structured into a number of modules, with well-defined interfaces between them. The time concept makes it possible to capture the time for different activities in the system and thereby conduct performance analysis and investigate functional properties of real time systems. CPNs and the Design/CPN tool have been successfully applied in a wide range of application areas and many projects have been carried out in industry [12].

1.4 Key Issues

The key issues discussed in this paper are scheduling of tasks on the mission control computer, and the data transfer across the data buses connecting the mission control computer and the various avionics sub-systems. A typical application software task scheduling mechanism, such as for the F/A-18 and F-111 aircraft, is based on a cyclic executive [18]. The cyclic executive executes an application that is divided into a sequence of non pre-emptive tasks, invoking each task in a pre-determined order throughout the execution history of the application [19]. One can distinguish two types of tasks, namely, rategroup and background tasks. The rategroup tasks are periodic and have higher priority than the background tasks, which may be considered as aperiodic. The cyclic executive repeats its task list at a rate that is known as a major cycle. The major cycle is further divided into periods known as minor cycles. The major cycles have a set of tasks scheduled that must meet the required deadline in order to maintain the integrity of the mission system.

1.5 Paper Overview

This paper presents our application of Coloured Petri Nets and Design/CPN for the scheduling and input/output processing analysis in airborne mission systems. This work has evolved from our previous modelling and analysis of airborne mission systems using CPNs [15, 16]. In Modelling Military Airborne Mission System [15], we described our modelling framework and its application to a generic AMS [2]. This model was enhanced and its schedulability analysis presented recently [16].

In Section 2, we discuss the development of a CPN model for a generic AMS and present our analysis approach and results. In Section 3, an overview is provided of the AP-3C AMS and mission computer task scheduling and information transfer across the mission equipment data bus. In Section 4, the initial investigation into the application of the CPN modelling framework to the AP-3C Orion Maritime Patrol Aircraft is described. Finally, Section 5 provides conclusions on the modelling method and plans for further research into analysis of certain system properties.

2. Generic Mission System Analysis

An overview of the CPN model of a generic AMS is shown in Figure 2 [16]. This depicts the hierarchy page of the CPN model. The nodes in Figure 1 correspond to modules (in CPN terminology, *pages*) of the CPN AMS model. An arrow from one node to another node means that the latter is a sub-page (sub-module) of the former. The page AMS is the top-level and most abstract page in the CPN model. The CPN AMS model consists of five main parts pages:

- AMS Sensors, (Sensors#8) and its sub-page,
- Mission Control Computer (MCC#3) and its sub-pages
- Controls/displays (ControlsDisplays#) and its sub-pages,
- Stores modelled by page Stores#10, and
- Serial data bus (SerialDataBus#11) connecting the component

The CPN model captures the set of tasks executing on the mission control computer and the transmission of messages between the devices of the system via the data bus. Being a timed CPN model, the time taken by different events in the AMS is captured using timestamps. In this paper we will not go into details of the individual pages in the CPN AMS model. For this, the reader is referred to [15] and [16].

The analysis of the CPN AMS model is based on the state space method of CP-nets as supported by Design/CPN [14]. The primary focus of the analysis has been to determine whether the tasks to be scheduled on the mission control computer are completed in time, and if this is the case, provide a schedule for the set of tasks. In addition to this, the size of the input/output queues of the I/O processor has been considered.

The basic idea behind state space methods is to construct a directed graph (called the state space) with a node for each reachable state of the CPN model and an arc for each transition between states. Since the state space contains all reachable states it represents all possible executions of the CPN model.

A problem to overcome in the analysis of the CPN AMS model is that the state space is infinite, which is due to time incrementing in the system.

To obtain a finite state, we assume that tasks are repeated in each major cycle. This implies that each major cycle in the generic AMS is identical and hence, if a schedule has been found for one

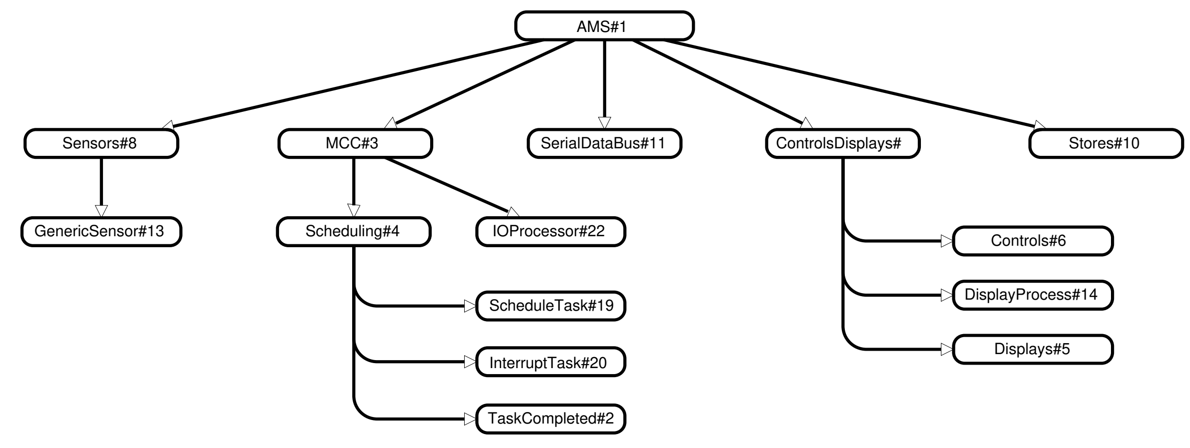

Figure 2. Hierarchy Page of CPN AMS Model.

major cycle, this schedule can be repeated in each major cycle. Therefore, we only consider the tasks in one major cycle, which leads to a finite state space.

The problem of finding a schedule can be formulated as finding a path in the state space leading from the initial state to a state where the major cycle has ended and all tasks were completed in time.

To make state space analysis feasible, we started out by selecting a small set of tasks and gradually introduced additional tasks. Also, we experimented with different priority policies for tasks accessing the CPU and for input/output queuing. Table 1 [16] lists different sets of tasks taken from [2] used for the analysis. The RG column gives the number of rategroup tasks in a given set. The BG column specifies the number of background tasks in the set. We have omitted the names of the tasks in each set, as they are not required for the discussion of our analysis results.

Table 1. Set of tasks used for analysis.

Task set	Tasks set name	RG	BG
S1	Displays and controls	6	4
S2	S1 + built-in test	7	6
S3	S2 + radar control	9	7
S4	S3 + targeting	10	10
S5	S2 + threat response	8	7
S6	S5 + RWR Control	9	8
S7	S6 + weapon control	10	12
S8	S7 + targeting	11	15

Table 2 [16] gives the size of the state space for the different sets of tasks listed in Table 1. The Nodes column gives the number of nodes in the state space, and the Arcs column gives the number of arcs in the state space. The IOSS column gives the maximum number of requests in the I/O requests queue at the I/O processor observed in the state space. The IOSP column gives the maximum number of requests in the I/O queue along a path in the state space corresponding to a schedule for the tasks.

The considered pre-emption and queuing policy allowed requests from rategroup tasks to overtake requests from background tasks in the IO queue, and both rategroup and background tasks had assigned priorities.

Table 2. Standard state space generation

Set	Nodes	Arcs	IOSS	IOSP
S1	77,982	127,316	6	4
S2	78,734	128,715	7	6
S3	485,054	811,734	9	7
S5	144,780	235,769	10	10
S6	142,022	234,257	8	7
S7	409,888	702,831	9	8

The generation of the state space was done in a breadth-first order, and the generation of the state space was truncated at nodes corresponding to a completed major cycle. In this way, a state space corresponding to one major cycle was obtained. The analysis results were obtained on a 1GHz Pentium III PC with 512 Mb of memory. Due to the size of the state space, it was not possible to obtain results for task sets S4 and S8.

In order to analyse and obtain schedules for sets S4 and S8, an extension was made to the Design/CPN tool, which allowed the state space to be generated in a depth-first order. The depth-first generation allows the state space construction to be terminated as soon as a path is found corresponding to a schedule.

Table 3 [16] lists the analysis results obtained using depth-first state space generation. It can be seen that the use of depth-first generation allowed the S4 and S8 task sets to be analysed. The statistics show that the early termination in the depth-first state space generation implied that significantly fewer states have to be considered. The benefit of early termination is obtained when a schedule exists. If no schedule exists the depth-first state space generation will consider the full state space.

Table 3. Depth-first state space generation.

Set	Nodes	Arcs	IOSS	IOSP
S1	4,120	4,119	7	6
S2	4,263	4,262	17	14
S3	5,394	5,393	17	14
S4	6,483	6,482	17	15
S5	4,497	4,496	17	14
S6	4,643	4,642	17	15
S7	5,109	5,108	17	15
S8	6,204	6,203	17	15

3. AP-3C Orion Mission System

The Maritime Patrol Group of the Royal Australian Air Force operates the AP-3C aircraft [20,21]. The AP-3C aircraft mission system upgrade provides enhanced mission capabilities and extends the P-3C life-of-type to 2015 [20,22]. It has a comprehensive sensor suite, which includes Radar, Acoustics Sensors, Infrared Detection System (IRDS), Magnetic Anomaly Detection (MAD) and Electronic Support Measures (ESM) [22,23]. The aircraft can be employed in a wide variety of single or combined roles and tasks. AP-3C roles include anti-subsurface and anti-surface warfare, surveillance, search and rescue, and maritime strike [20].

The Airborne Mission System is characterised by a set of sub-systems interconnected by a serial communications data bus. A high-level block diagram of the AP-3C Airborne Mission System (AMS) architecture is shown in Figure 3 [24,25]. It consists of a Mission Equipment Bus (MEB) that provides inter-communication channels between the following sub-systems: Embedded GPS/INS Navigation, Acoustic Processors, MAD and Radar. The Data Management System (DMS) provides specialised interfaces to the following sub-systems:Armament/Ordinance (ARM/ORD), ESM and IRDS. The operators are provided with a high-resolution display and entry panel directly from the DMS.

The Communication (Comm) sub-system interfaces with these sub-systems via the Avionics Equipment Bus (AEB), which is connected to the MEB via the Navigation sub-system [24].

3.1 Data Management System

The DMS is a centralised mission control and management sub-system. It is a complex multiprocessor system consisting of several Enhanced General Purpose Controllers (EGPC), custom computing devices and supporting software [25]. Two input/output (I/O) processor cards provide the interface between the EGPCs and the MEB. The DMS provides the overall AMS management and normally acts as the MEB bus controller (BC) [24].

The DMS software consists of a set of software components, which process sensor data and input from controls, performs necessary mission-oriented computations and provides outputs to the displays and other avionics equipment [26]. Typical software components include the executive, navigation, stores management, display control, and data management. The runtime executive schedules and dispatches the execution of control tasks and services interrupts during various operations [26]. The runtime executive component is typically responsible for the following functions: application software (task) scheduling, interrupt management, input/output scheduling and error management.

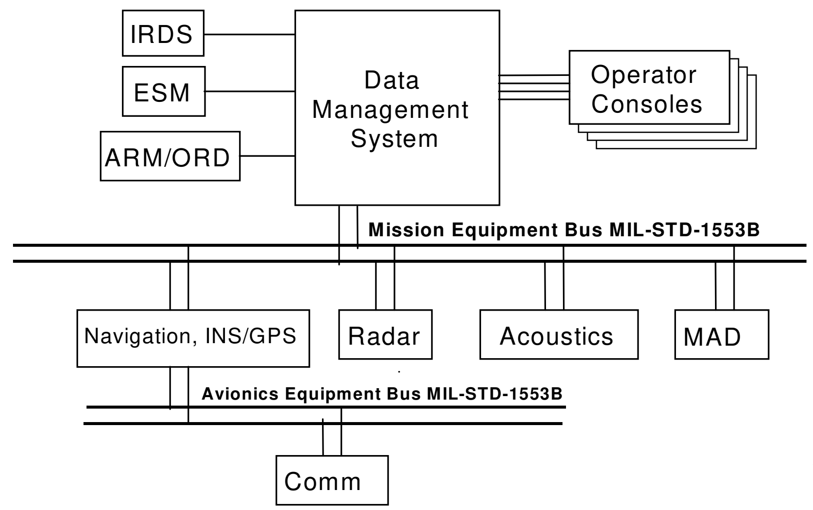

Figure 3. AP-3C Airborne Mission System Block Diagram

The DMS executive software is based on a commercial Ada run-time kernel [26]. The scheduling policy is pre-emptive and is executed by priority in a round-robin fashion [27].

An EGPC sends a number of messages to the I/O processor for transmission to the addressed remote terminal (RT) [24], e.g. Navigation sub-system or Radar subsystem, via the MEB. The MEB minor frame rate (described in the next section) sets a real-time clock interrupt to the I/O software. At the beginning of each new minor frame, an interrupt occurs, and the I/O processor starts issuing the messages for that frame.

3.2 Mission Equipment Bus

The Dual Redundant MEB is a MIL-STD-1553B digital time division multiplexed command/response serial data bus [24,28], operating at 1 Mbit per second. The serial bus architecture minimises the bulk of interconnecting wiring needed on platforms, where physical weight and volume are constrained and the command/response protocol provides confirmation of delivery for data bus messages. The MIL-STD-1553B standard supports up to quad redundant transmission media, with dual redundancy commonly used. All messages are initiated by the bus controller and require the addressed RT to receive or transmit a specified number of data words associated with a specified terminal sub-address. The MIL-STD-1553B message protocol also requires the transfer of a status word from remote terminals participating in a message transfer to confirm to the BC that the message was successfully transferred. Non-transmission of a status word indicates message failure to the BC, which could then retry the message or adopt a predefined error management policy. [28].

The AP-3C MEB uses a pre-defined framing as shown in Figure 4 [24]. It contains messages that are managed by the BC. A major frame consists of 40 minor frames of 25 ms each that are allocated messages in a pre-determined schedule. There are three types of message scheduling: Periodic, Aperiodic and Manual. Periodic (P) messages are transmitted every major frame. Manual (M) periodic messages are switched on or off depending on the mode of operation; when they are on they are transmitted every major frame. The Aperiodic (A) messages are transmitted as required. Framing synchronisation is achieved by periodic messages sent each minor and major frame. The frequency of message transmission is a function of the number of minor frames in which a message is scheduled each major frame. For example a 40Hz message is scheduled every minor frame, a 4Hz message is scheduled in 4 minor frames, 10 minor frames apart.

This scheduling mechanism provides reliable bus communication for critical scheduled periodic messages, but non-periodic (aperiodic messages) communications that require significant data transfer are limited by the remaining bandwidth as defined by the bus controller's schedule. This scheduled messaging approach allows the designer to adopt a bandwidth utilisation goal.

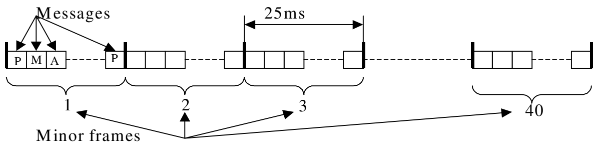

Figure 4. AP-3C MEB Major Frame

4. AP-3C Mission System Modelling

We now present the AP-3C AMS model (CPN model). The AP-3C CPN model has been constructed based on the framework represented by the CPN AMS model briefly described in Section 2. Some parts of the CPN AMS model needed only minor changes while other parts required the CPN AMS model to be further refined. The main differences between the CPN AMS and the AP-3C models concern are: the devices attached to the serial data bus, the functioning of the bus itself, and finally the presence of an additional avionics bus. The scheduling of messages on the serial data bus has been refined compared with the CPN AMS model. The reason for this is that the scheduling of DMS sub-system related tasks is loosely coupled to the scheduling of the serial data bus messages, and hence cannot be considered in isolation [24].

In the following sections we provide first an overview of the model, and then detailed descriptions of the components of the AP-3C CPN model.

4.1 Model Overview

The hierarchy page of the AP-3C CPN model is depicted in Figure 5. The main components of the system are the Data Management System (DMS#3 and its sub-pages), the MEB and AEB MIL-STD-1553B data buses (MILSTD1553#11) and the devices (Device#13) attached to the MEB. The Data Management System model is composed of an EGPC#4 page modelling the scheduling of DMS tasks as well as their execution, and an IOProcessorCard#22 page. The EGPC#4 page passes the input/output requests of tasks to the IOProcessorCard#22. The IOProcessorCard#22 then sends messages over the data bus. The data bus transmits the message to the appropriate device and waits for an acknowledgment. The reply message is then passed to the IOProcessorCard#22 that signals to the tasks, which messages have been transmitted. The processing of previously received data then resumes execution on the EGPC#4.

A main consideration in the construction of the AP-3C CPN model is to choose an appropriate level of abstraction to begin with, but which could easily be further refined later on. For example, although the AP-3C uses several EGPCs on the Data Management System, we modelled one in such a manner that several instances of the same EGPC#4 model can be considered in the future.

The AP-3C Mission Equipment Bus and the Avionics Equipment Bus both satisfy the MIL-STD-1553B requirements. Therefore, they are modelled exactly in the same manner.

As we will see later, the policies to send messages on these buses are different from the generic AMS, and are taken care of by the IOProcessorCard#22. In the modelling of the DMS, we have abstracted the DMS mission software as a set of tasks. The bus controller software and I/O processor card within the DMS has been abstracted into a single page (IOProcessor). Sub-system communication on the MEB has been modelled functionally with time because of its relation to real time AMS events.

4.2 The Airborne Mission System Model

The top-level AMS#1 page is shown in Figure 6 and corresponds to the most abstract level in the CPN model. An important component of the Airborne Mission System is the Data Management System, which is responsible for executing a set of tasks. The Data Management System communicates with the Radar, Magnetic Anomaly Detection (MAD), Acoustics Sub-system and Navigation through the Mission Equipment Bus (MEB). However, the Navigation System is also used to pass messages on the Avionics Equipment Bus to the Communication Subsystem. Note that a number of sub-systems such as the operators consoles, IRDS, ESM, ARM/ORD and Displays in Figure 3 have not been modelled since we are interested in data communications across the MEB.

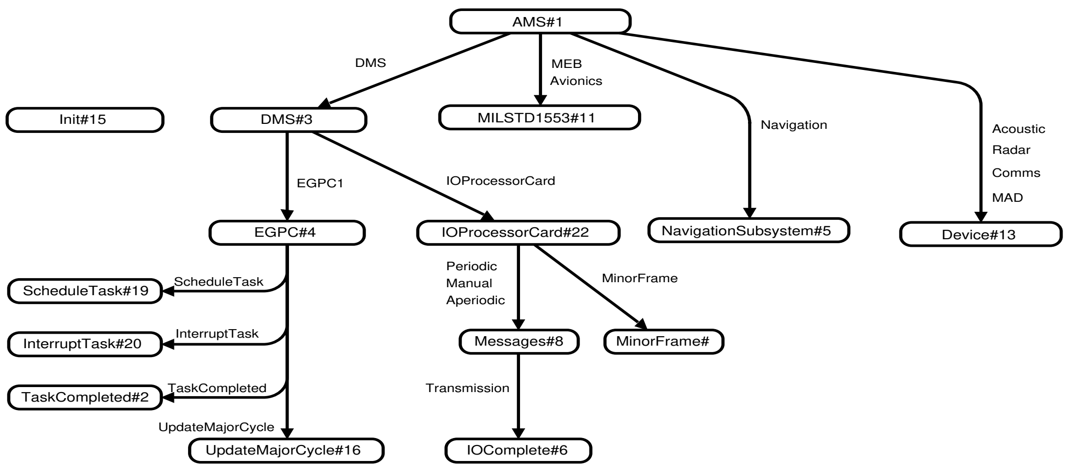

Figure 5. Hierarchy page of the AP-3C CPN model

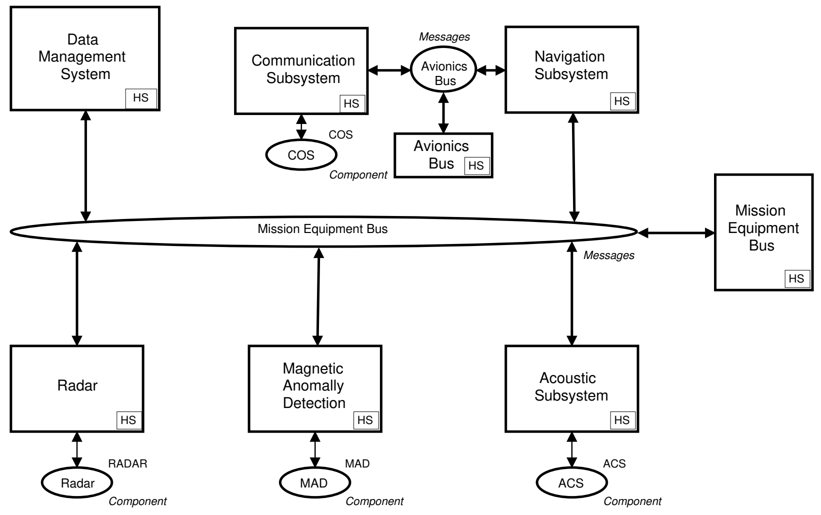

Figure 6. AP-3C Airborne Mission System – Page AMS

Each of the rectangles in Figure 6 is called a substitution transition. A substitution transition in a CPN model represents compound behaviour. The details of the compound behaviour is represented by a substitution transition and modelled by the associated sub-page (which in turn may contain substitution transitions). The sub-page associated with the Data Management System substitution transition is the DMS#3 page (see Figure 5) which models the DMS in detail (see Figure 7). The sub-pages associated with the Avionics Equipment Bus and Mission Equipment Bus substitution transition are both the MILSTD1553#11 page that models the MIL-STD-1553B data bus in detail. The sub-page associated with all other substitution transitions on page AMS#1, except the Navigation Subsystem, is the page Device#13 (see Figure 9). The page Device#13 models an abstract device attached to a serial data bus. An important abstraction made in the AP-3C CPN model is to consider all devices except the Navigation Sub-system to have the same behaviour. The details of the Navigation Sub-system are modelled by page NavigationSubsystem#5. The Navigation sub-system is not considered a simple device as it can relay messages between two data buses, in this case the MEB and the AEB.

The ellipses in Figure 6 are called places and are used to model the states of the system. The state of a CPN model is represented by a distribution of data values (known as tokens) on the places of the CPN model. The execution of a CPN model then consists of occurrences of enabled transitions that change the distribution of tokens and their values on the places of the CPN model [12,13].

The place Mission Equipment Bus is used to model the state of the MEB. Similarly, the place Avionics Bus is used to model the state of the AEB. The places COM, Radar, MAD, and ACS are used to model the state of the corresponding device.

Arcs connect transitions and places on a CPN model. These arcs have arc inscriptions describing the tokens removed from and added to places upon the occurrence of enabled transitions. Arc inscriptions are written in the functional programming language Standard ML (SML) [17].

4.3 The Data Management System Model

The DMS most abstract level is modelled by page DMS#3 depicted in Figure 7. It is the sub-page associated with the Data Management System substitution transition in Figure 6. The DMS#3 page contains two substitution transitions: EGPC1 and the 1553 I/O Processor Card. The sub-pages of the EGPC1 substitution transition model the execution of tasks on the EGPC. The two places Tasks and AllTasks are used to model the set of tasks executing on the EGPC.

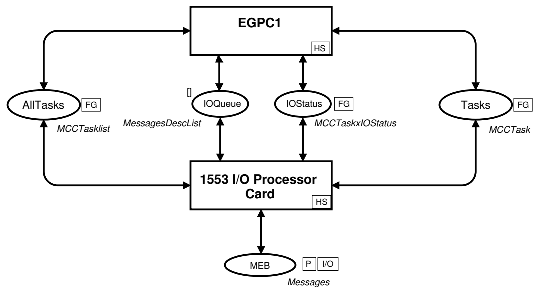

Figure 7. Data Management System –Page DMS.

Each task executing on the EGPC is represented as a token on place Tasks. This token comprises the attributes and state of the task. All tasks on the EGPC are also represented by a list on place AllTasks. This is done so that the Design/CPN tool can determine the tasks with the highest priority at a given moment in time [12,13,14].

The places IOQueue and IOStatus model the connection between the EGPC and the I/O Processor Card. In this AP-3C model, we have not included many of the internal components and connections, but have chosen to model these by a delay. The connections between the EGPC and the processor card are modelled as a queue between the EGPC1 and I/O Processor Card substitution transitions, where tasks can place requests for input and output. The place MEB is used to model the interface between the I/O 1553 Processor Card and the MEB.

Since we do not need the details of the implementation of tasks, we model the tasks as requesting their input (e.g. from sensors and radars) at the beginning of their execution, and sending their output at the end of their execution. The input requests to be performed, before task execution can start, are passed by the I/O processor to the MEB. When all the information required has been received by the 1553 I/O processor, the task can be executed by the EGPC. When its execution is complete, output occurs. Then the task waits until it is rescheduled. In this paper we will not go into further detail with the sub-pages of the EGPC substitution transition, as these pages are essentially identical to the corresponding pages in the generic CPN AMS model [15,16].

4.4 The 1553 I/O Processor Card Model

Figure 8 depicts the page IOProcessorCard which is the sub-page of the 1553 I/O Processor Card substitution transition in Figure 7. As we will see in Section 4.6, this means that the modelling of the MIL-STD-1553B data bus itself is simplified.

The places IOQueue and IOStatus are the same as those on page DMS (see Figure 7). The substitution transition MinorFrame models the timing on the I/O processor card as determined by the minor frames. The place Minorframe contains a token, which indicates a data value corresponding to the current minor frame.

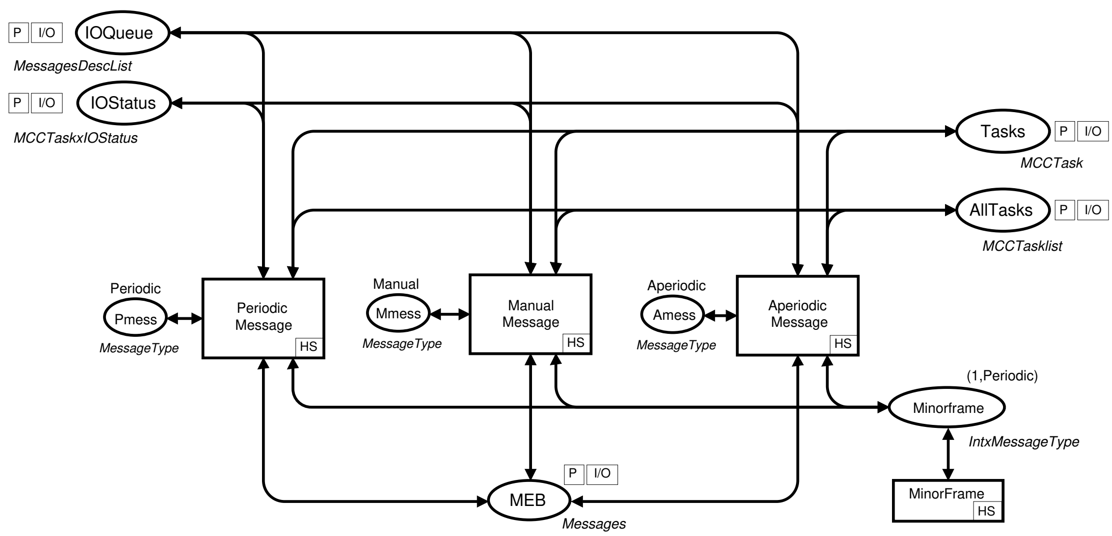

Figure 8. 1553 I/O Processor Card – Page IOProcessorCard

The substitution transitions Periodic Message, Manual Message, and Aperiodic Message model the transfer of messages across the data bus. The priority mechanism on the I/O processor card is such that all periodic messages for a given minor frame will be transmitted first. Once all periodic messages for the current minor frame have been transmitted, manual messages for the minor frame (if any) are transmitted. Then all aperiodic messages will be transmitted in the remaining time of the minor frame.

4.5 The Device Model

Figure 9 depicts the Device page modelling the abstract sub-systems of the AP-3C. Place Bus Interface models the connection to the serial data bus, which can be either the MEB or the AEB. Place State is used to model the nature of the device and contains a token corresponding to the specific device in question. The transition StartTransfer models data transfer between the DMS and the devices. This transition event is triggered by an IOstart signal from the serial data bus, and will put the device in a Transfer State for the amount of time specified as part of the data transfer request. Once this time has elapsed, the device will signal the completion of the data transfer by sending an IOcomplete signal to the serial data bus. This IOcomplete signal will be passed to the I/O Processor Card by the data bus interface.

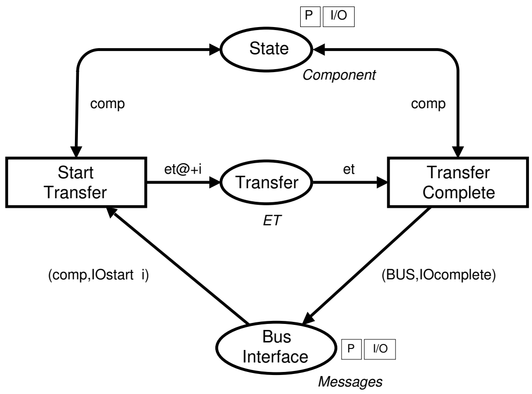

Figure 9. Abstract device – Page Device

4.6 The Data Bus Model

Figure 10 depicts the page MILSTD1553 modelling the MIL-STD-1553B data bus. The places Idle and Busy model the state of the serial data bus. The transition StartTransmit models the start of data transfer across the bus. The serial data bus will then change its state from idle to busy by placing a token in place Busy that specifies the two components (or sub-systems) between which data is being transferred. The serial data bus will then put the receiving/transmitting entity (sub-system or device) into a data transfer state by adding a token to place Bus Interface. When the entity has received/transmitted the data, it will signal the completion to the serial data bus by placing a token (BUS,IOcomplete) in place Bus Interface. This will trigger the occurrence of the transition TransmitComplete, change the state of the serial data bus from busy back to idle, and send a signal (srccomp,IOcomplete) to the I/O processor to indicate that the data transfer across the serial data bus is complete.

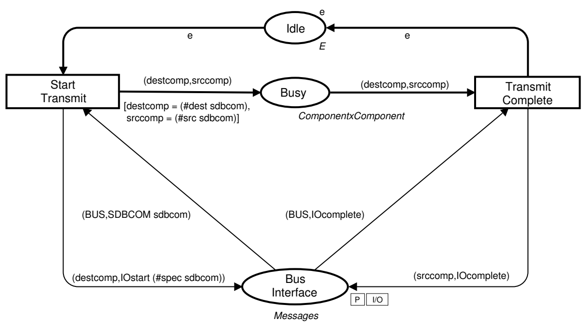

Figure 10. Serial data bus – page MILSTD1553

5. Conclusions and Future Work

This paper provides a summary of our analysis of a CPN model of a generic Airborne Mission System [2] and then our application of this model framework [15,16] to provide an initial model for the AP-3C Orion Maritime Patrol Aircraft AMS.

We have demonstrated how analysis of scheduling and input/output queues of an Airborne Mission System can be done using state spaces. This modelling approach creates a parametric CPN model, which simplifiesanalysis of different task sets and scheduling mechanisms, without major modifications to the model.

Our initial model of the AP-3C AMS is designed to allow ease of updates with only minor changes. The model provides a basis to perform analysis of the AP-3C and is focused on the Data Management System task scheduling and data transfer across the Mission Equipment Bus. The structure of the model has been presented. Selected parts of this model are then described, to provide insight into the approach and the constructs used to model various AMS features. We have taken advantage of system commonalities (for example in devices and the serial data buses) to provide a modular and maintainable model. The model has been designed to facilitate the incorporation of enhancements. For example, any number of controllers (EGPCs) can be included by instantiation of the EGPC#3 page the required number of times.

This model has been developed at a level of abstraction that as easily amenable for analysis and evaluation.

The next step in our work is to validate this model against real data. In order to reduce the complexity of mission system behaviour we intend to adopt the following strategy: select a relevant mission scenario such as a sea search and rescue mission; and define the operational environment, identifying the mission activities and their related tasks performed on the DMS. Once this mission is modelled we can validate it by comparing the CPN model predictions with the data obtained for the same scenario simulated using the RAAF AP-3C Integrated Test and Training Facility [22].

Once the CPN model is validated it will be used to determine the performance of the MEB. We then intend to extend the model with key details of the AMS to predict its behaviour under a range of conditions using the analysis approach outlined in this paper.

7. Acknowledgements

The authors would like to acknowledge the technical discussions with members of our teams, in particular, Scott Simmonds of Adacel Technologies and Rodney Dodd of Air Operations Division. We would also like to acknowledge our reviewer Flight Lieutenant John Postle, AP-3C DMS Manager Royal Australian Air Force.

8. References

[1] *Major Capital Equipment Project Delays or Cost Overruns*, 2001, Additional Budget Estimates – Defence Portfolio, Senate Foreign Affairs, Defence and Trade Legislation Committee, Canberra, Australia.

[2] Locke, C.D., Vogel, D.R., Goodenough, J. B., 1990, *Generic Avionics Software Specification*, Technical Report CMU/SEI-90-TR-8, Software Engineering Institute, Carnegie Mellon University, PA.

[3] Liu, J., 2000, *Real-Time Systems*, Prentice-Hall, NJ.

[4] Sun, H., 2001, *Modelling and Schedulability of Real-Time Tasks with Timed Automata*, Proceedings of International Conference on Parallel and Distributed Processing Techniques and Application, Vol. 4, CSREA Press, pp. 2146-2151.

[5] Amnell, T., Fersman, E., Mokrushin, L., Pettersson, P., Yi, W., 2002, *Times – A Tool for Modelling and Implementation of Embedded Systems*, Proceedings of 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2002), Grenoble, France, Lecture Notes in Computer Science, Vol. 2280, Springer-Verlag.

[6] Behrmann, G., Fehnker, A., Hune, T., Larsen, K., Peettersson, P., Romijin, J., 2001, *Efficient Guiding Towards Cost-Optimality in UPPAAL*, Proceedings of 7th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2001), Genova, Italy, Lecture Notes in Computer Science, Vol. 2031, Springer-Verlag.

[7] Fehnker, A., 1999, *Scheduling a Steel Plant with Timed Automata*, Proceedings of 6th International Conference on Real-Time Computing Systems and Applications (RTCSA), IEEE Computer Society, pp. 280-286.

[8] Desel, J., Reisig, W., 1998, *Place/Transition Petri Nets*, Lectures on Petri Nets I: Basic Models, Lecture Notes in Computer Science, Vol. 1491, Springer-Verlag, pp. 122-173.

[9] Burns, A., Wellings, A. J., Burns, F., Koelmans, A., Koutny, M., Romanovsky, A., Yakovlev, A., 2000, *Towards Modelling and Verification of Concurrent Ada Programs Using Petri Nets*, Department of Computer Science, University of Aarhus, pp. 115-134.

[10] Shatz, S., Tu, S., Murata, T., Duri, S., 1996, An *Application of Petri Net Reduction to Ada Tasking Deadlock Analysis*, IEEE Transactions on Parallel and Distributed Systems, Vol. 7, No. 12, pp. 1309-1324.

[11] Berthelot, G., Petrucci, L., 2001, *Specification and Validation of a Concurrent System: An Educational Project*, International Journal on Software Tools for Technology Transfer, Vol. 3, No. 4, pp. 372-381.

[12] Jensen, K., 1997, *Coloured Petri Nets. Basic Concepts, Analysis Method and Practical Use* (*Vol. 1-3)*, Monographs in Theoretical Computer Science, Second Edition, Springer-Verlag.

[13] Kristensen, L.M., Christensen, S., Jensen, K., 1998, *The Practitioner's Guide to Coloured Petri Nets*. *International Journal on Software Tools for Technology Transfer,* (2). Springer-Verlag, pp. 98-132.

[14] *Design/CPN Online*, 1996, http://www.daimi.au.dk/designCPN/.

[15] Kristensen, L.M., Billington, J., Qureshi, Z.H., 2001, *Modelling Military Airborne Mission Systems for Functional Analysis*, Proceedings of 20th Digital Avionics Conference, Daytona Beach, FL, 14-18 October.

[16] Petrucci, L., Kristensen, L.M., Billington, J., Qureshi, Z. H., 2002, *Towards Formal Specification and Analysis of Avionics Mission Systems*, Proceedings of Workshop on Formal Methods Applied to Defence Systems, Adelaide, Australia, June 2002, Conferences in Research and Practice in Information Technology, Vol. 12, pp. 95-104, Australian Computer Society.

[17] Ullman, J., 1998, *Elements of ML Programming*, Prentice-Hall, NJ.

[18] *F/RF-111C Avionics Update Program, 1992, Software Design Document for the Mission Computer Operational Flight Program,* Volume I, Rockwell International Corporation.

[19] Locke, C. D., 1992, *Software Architecture for Hard Real-Time Applications: Cyclic Executive vs. Fixed Priority Executives,* The Journal of Real-Time Systems, Vol. 4, pp. 37-53.

[20] *The Royal Australian Air Force (RAAF) Maritime Patrol Group (MPG)* http://www.defence.gov.au/raaf/mpg/, Accessed 25 July 2002

[21] *The official newspaper of the Royal Australian Air Force, May 23, 2002. Up, up and away, Transition to AP3C Orion complete.* http://www.defence.gov.au/news/raafnews/editions/4409/story05.htm, Accessed 25 July 2002.

[22] *Defence Materiel Organisation, Projects, Air 5276 Phase 2A - P3 Update Implementation*, http://www.defence.gov.au/dmo/asd/air5276/air5276p2.cfm Accessed 25 July 2002

[23] *Acronym Finder*: http://www.acronymfinder.com/ Accessed 25 July 2002.

[24] *Mission Equipment Bus (MEB) Protocol Specification.* Report F6250.00.151, 11 March 1996, CDRL-ENG-46-ACS-ICD-0074, Rev D 10 Jan 2000. *Project Air-5276* Royal Australian Air Force, Department of Defence, Australia.

[25] *DMS Sub-system, Vol 17,* Report 7371933, Rev D, 30 September 1997, CDRL-ENG-46-CDR-DMS-0206, 14 Nov 2000. *Project Air-5276* Royal Australian Air Force, Department of Defence, Australia.

[26] *Software Design Document for the Operating Systems CSCI*, Report 7371902, Rev B, 26 February 1999, SDRL-ENG-65D-03-03, CDRL-ENG-32-DMS-SDD-0423, 2 July 1999. *Project Air-5276* Royal Australian Air Force, Department of Defence, Australia.

[27] *Runtime Systems Guide VADScross M68000*, ver 6.2.3.0, June 1995, Rational Software Corporation, Santa Clara, California, USA.

[28] *MIL-STD-1553B Notice 3*, 31 Jan 1993, Digital Time Division Command/Control Multiplex Data Bus.