Model-based Operational Planning Using Coloured Petri Nets

Lin Zhang, Brice Mitchell, Lucia Falzon and Mike Davies
Information Technology Division
Defence Science and Technology Organisation
PO Box 1500, Salisbury, SA 5108, Australia
Email: Lin.Zhang@dsto.defence.gov.au

Lars M. Kristensen and Jonathan Billington
Computer Systems Engineering Centre
School of Electrical and Information Engineering
University of South Australia, Mawson Lakes, SA 5095, Australia

Abstract

This paper presents a framework for modelling military operational plans with Coloured Petri Nets (CPN) and its supporting computer tool, Design/CPN. A mapping relationship between key concepts of an operational plan and CPN constructs is established. Parts of an exercise campaign plan were modelled using the proposed framework, and state space analysis of the model conducted to assess operational readiness of the friendly force.

1. Introduction

The planning process in a command and control environment at the operational level of war includes Course of Action (COA) development and analysis as a prelude to the development and analysis of operational plans. The main outcome of COA development and analysis is a suitable COA typically specifying a sequence of parallel and/or sequential operations which attempt to fulfil the defined objectives of the military campaign when executed, i.e., resulting in the desired end-state. Probabilistic models such as Influence Nets and Bayesian Nets [1] may be used to help develop and analyse COAs [2,3]. The main outcome of the subsequent development and analysis of operational plans is then a practical proposal for the employment of available resources to accomplish the tasks and operations of the chosen COA. As such, an operational plan can be seen as an implementation of a chosen COA. In general there are many possible ways in which a COA can be implemented. It is therefore of interest to support the development of operational plans by developing tools and techniques that make it possible to co-ordinate activities in order to optimise resource usage, identify critical resources, and check feasibility of operational plans with respect to timing and resource constraints, as well as compare operational plans in terms of specific metrics. This paper focuses on tools and techniques for the development and analysis of operational plans.

Coloured Petri Nets (CPNs) [4,5] are a graphically oriented modelling language capable of expressing concurrency, non-determinism, and system concepts at different levels of abstraction. CPNs combine Petri Nets and programming languages within the same mathematical framework. Petri Nets are used to model concurrency, synchronisation, and resource sharing and allocation, whereas a functional programming language is used to model data manipulation and to create compact and parameterised models. A software package, Design/CPN, supports the modelling, simulation and analysis of CPNs [6]. CPNs and Design/CPN have a wide range of application areas such as data networks and communication protocols [7], hardware design, embedded systems [8] and decision-making organisations [9].

In this paper we present a framework for representing operational level plans and show how the key concepts of this framework can be represented by CPN. As a representative case study we present selected parts of a CPN model of an operational plan covering the deter phase of a fictitious military campaign. CPN models of operational plans can be analysed, and we give some examples of the kind of results that can be obtained for the deter phase.

Previous work in using CPNs and Design/CPN for COA development and evaluation is found in [10] where Wagenhals et al studied the effect of timing and sequencing on the probability of success of a particular COA. In [10], a set of actionable events is initially identified and their impact on the achievement of operational objectives analysed in a probabilistic equilibrium manner using an influence net modelling tool. A formal procedure is then devised to convert the probabilistic equilibrium model of actionable events together with timing information into a CPN model of the situation. Upon execution, the model generates a timed sequence of probabilities of success for an acceptable sequence of actions that defines a COA. The benefit of using CPNs [10] is the ability to derive probabilities of success (or failure) over time that represent the risk of a COA varying over the entire period of execution. This will facilitate choosing a COA that has a low risk during execution as well as a high overall probability of success. Our paper differs from the work of Wagenhals et al [10] in that our focus is primarily on the feasibility of a predefined COA with the given resource and time constraints.

This paper is organised as follows. Section 2 briefly introduces CPN terminologies that are used in this paper. Section 3 proposes a framework for representing operational plans, including a discussion of key concepts that can be modelled by CPN constructs. Section 4 describes selected parts of a CPN model of an operational plan covering the deter phase of a military campaign. Section 5 provides examples of the kind of questions that a CPN campaign model can help answer based on state space analysis. We draw conclusions and discuss future work in Section 6.

2. CPN Preliminaries

This section gives an informal introduction to CPN constructs that are used in this paper. A formal definition of coloured Petri nets can be found in [4].

A coloured Petri net can be described in terms of a net structure, colorsets (eg, data types), initial marking, and enabling and occurrence rules. The schema of a simple CPN is shown in Figure 1.

There are three components in a CPN net structure: places, transitions and arcs, depicted in Figure 1 as ellipses, boxes and arrows, respectively. *Places* can hold *tokens* that model the states of a CPN. *Transitions* represent actions that can be executed to change CPN states. *Arcs* connect places and transitions.

Tokens in a place represent data. The types of data that a place can hold are specified by a set of *colours* (eg, a CPN *colorset*). The *marking* of a CPN place specifies the numbers and colours of tokens in that place. A *CPN marking* includes markings of all places. A CPN marking in a particular instant represents the state of the net in that instant. An *initial marking* represents the initial state of a CPN.

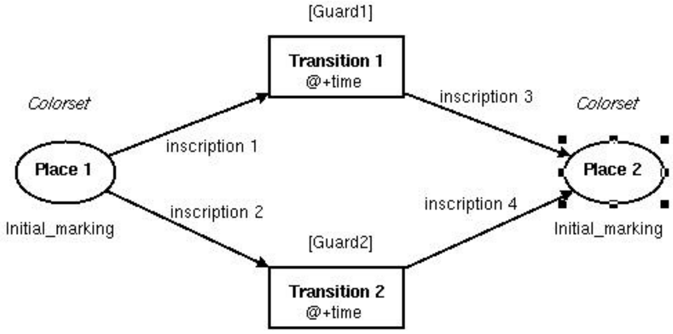

Figure 1 Schematic diagram of a coloured Petri net example

Given a CPN net structure and an initial marking, the dynamics of a CPN is determined by the enabling and occurrence rules, modelled by *arc inscriptions* and *guards* of transitions. There are two types of arcs in respect to transitions: incoming arcs and outgoing arcs. Incoming arcs connect from *input places* to a transition, and outgoing arcs connect from a transition to *output places*. Inscriptions on incoming arcs specify the numbers and colours of tokens from the input places that must be in place in order for a transition to be enabled; and, once the transition occurs, to be consumed. Inscriptions on outgoing arcs specify tokens that the occurrence of a transition produces and puts into the output places.

Transition guards are optional, and if present, impose additional conditions for transitions to be enabled and to occur. Enabled transitions can occur either concurrently or sequentially, and in various orders, depending on the numbers of available tokens in places. It is also possible that the occurrence of some transitions changes the state of a net such that the conditions of other enabled transitions are no longer met.

The occurrence of a transition may take time. The duration of a transition is prefixed by the CPN symbol “@+”, as shown in Figure 1. A CPN that also models time is known as a *timed CPN*.

3. A Framework for Representing Operational Plans

We define an operational plan as a description of military *tasks*, in a prescribed *order*, that are intended to reach a desired operational *end state*, normally within a given time. The key words in this definition are tasks, ordering, and an end state. We discuss these and associated key concepts in the following subsections. CPN constructs are used to represent the concepts where appropriate. Our purpose is to present a conceptual framework for constructing executable models of an operational plan in order to assess the feasibility of a COA given resource and time constraints.

3.1 Tasks

For military planning at the operational level, the notion of a task is a central concept. For an operational level commander and his/her staff, a planning process is formally initiated when a task is received, often in the form of a warning order or a planning directive. The commander and staff will then conduct mission analysis in order to fully understand the task at hand. As well as a full appreciation of the situation including the enemy, friendly and neutral forces, and physical environment, mission analysis also involves the study of the superior commander's task. Having identified their own task, the rest of the planning process is to develop tasks for subordinates, properly ordered, timed, and phased in order to achieve the overall mission. In this sense, tasks are hierarchical, and they are the basic building blocks of an operational plan. As a high level modelling language, CPNs provide two facilities to model hierarchies of tasks [4]: substitution transitions and port places. A *substitution transition* is a more detailed, lower level representation of a transition (eg, a sub-model); and *port places* enable several places across different levels of a hierarchy to behave like a single place. Examples of hierarchical representations of tasks are given in the next section.

A task is rarely defined by its mechanism, eg, how to carry out the task. Normally a task is defined by characteristics external to the task itself. In this paper and in accordance with planning principles at the operational level [11], we define a task by its pre-conditions, triggers, resources, duration and effects, as illustrated in Figure 2. For example, one can specify an amphibious landing task as

Pre-conditions:	Air and Maritime Superiority, favourable weather
Trigger:	On order
Resources:	Amphibious Ready Group
Effect:	Beachhead assaulted
Duration:	4 Hours

We may associate a *transition* of a timed CPN with a military task. The resulting CPN model would be a timed Petri net with the time aspect of the model representing the task duration.

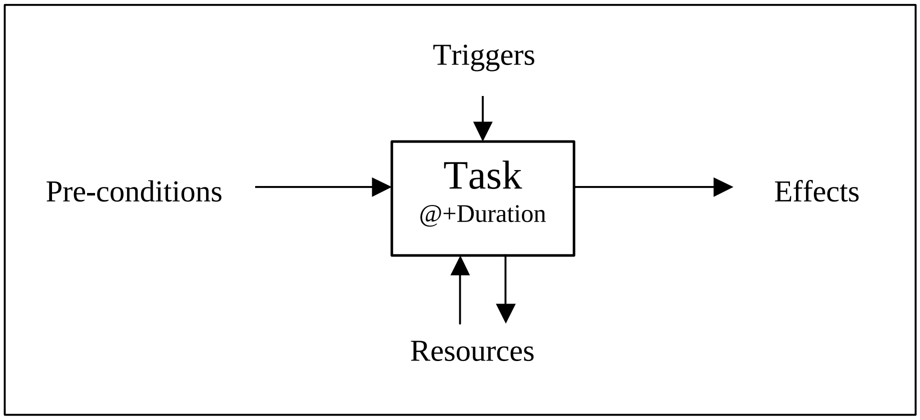

Figure 2 Task representation

3.2 Pre-conditions, Triggers and Effects

Pre-conditions of a task set standards under which the task can be carried out. As in the example of the amphibious landing task, the pre-conditions include the establishment of air and sea control, and the existence of favourable weather conditions. A trigger represents an external event that sets off a task, eg, timing control. An example of trigger is the authorisation from a superior for a task to be carried out. It is possible that a trigger is not required. In this case, tasks can be carried out spontaneously once the pre-conditions are met.

In terms of the CPN, one can associate the concept of pre-conditions of a task with pairs of input places and input arc-inscriptions of a transition, and triggers with either additional pairs of input places and input arc inscriptions and/or with a guard. Similarly, the effect that a task produces when executed, changes the nature of the system and environment, and can be modelled by pairs of output arc inscriptions and output places of a transition. We will call CPN places that hold tokens for representing pre-conditions, triggers and effects, *logical places*.

3.3 Resources

Resources, both military and civil, represent the key constraint and one of the most important factors in operational planning. At the onset of planning, the operational commander is assigned resources including forces, platforms and other physical resources to achieve a designated task. Throughout the process, the scope of COA development is limited by the availability and consumption of resources (including casualty and financial loss). In the modelling of operational plans using CPNs, resources are represented by *resource places*. In this paper, we use one single resource place with a composite data type for convenience.

3.4 End State

End-state is another important concept in military planning. A national end-state is the set of desired conditions, incorporating the elements of national power, that will achieve the national objectives. A military end-state is the set of desired conditions beyond which the use of military force is no longer required to achieve national objectives.

Generally, a national end-state can be defined in terms of social, political, economic, geographical, environmental and military conditions. It is the military conditions that constitute the military end-state. In this sense, a military end-state can be seen as one dimension of a national end-state, although the military dimension is not necessarily orthogonal to other dimensions of the national end-state.

In the state space analysis of a CPN model, there may be certain states in which no transitions are enabled. These states are known as *dead markings*. An end state should be revealed as one of the dead markings of the state space of the CPN model. If the end state is not one of the dead markings, then there is a serious problem with the plan. If more than one dead marking is present in the state space, then there are alternative end states, some of which may be undesirable. In the ideal case, there would be just one dead marking, corresponding to the desired end state. The state space analysis of Design/CPN automatically provides a report that identifies all dead markings. Thus there is a very good match between the desired outcomes of an operational plan and the results that can be obtained by CPN modelling and analysis.

3.5 Ordering

Tasks in an operational plan are partially ordered. While orders are imposed in planning through the use of pre-conditions, triggers and effects, flexibility in the order of execution should be allowed, and sometimes maximised, for subordinate commanders to initiate responses under local circumstances, and to achieve synchronised effects. Within the constraint of the logical order, the number of possible implementations of a COA is only limited by the availability of resources. The CPN model of an operational plan, when validated, enables enumeration of all possible orderings of tasks; the orderings that lead to both desirable and undesirable dead markings eg, the end state, can also be identified.

4. Coloured Petri Net Modelling

In this section, we demonstrate the modelling concepts introduced in the previous section by presenting a CPN model of parts of a campaign plan that was developed during a planning exercise.

The scenario for planning is such that there are signs of a fictitious threat force preparing to invade a fictitious neighbouring friendly country of Australia, and an operational Headquarters of the Australian Defence Forces (ADF) is asked to develop a campaign plan to negate the threat in order to maintain sovereignty of both Australia and the neighbouring country. A high level abstraction of the campaign plan is depicted in Figure 3 where the campaign is designed to have three phases: *Deter*, *Defeat* and *Develop*, corresponding to the three transitions connected to the *Resource* place. An additional transition *DecisionPoint* in Figure 3 decides whether the situation has deteriorated to an open conflict, or the *Deter* phase has achieved its full effect such that the threat has abandoned its expansionist strategy, in accordance with the triggers, *invade* or *withdraw*, respectively, in the *Triggers* place. In the latter case, the military end state would have been achieved, and the *Defeat* and *Develop* phases will not have to be executed.

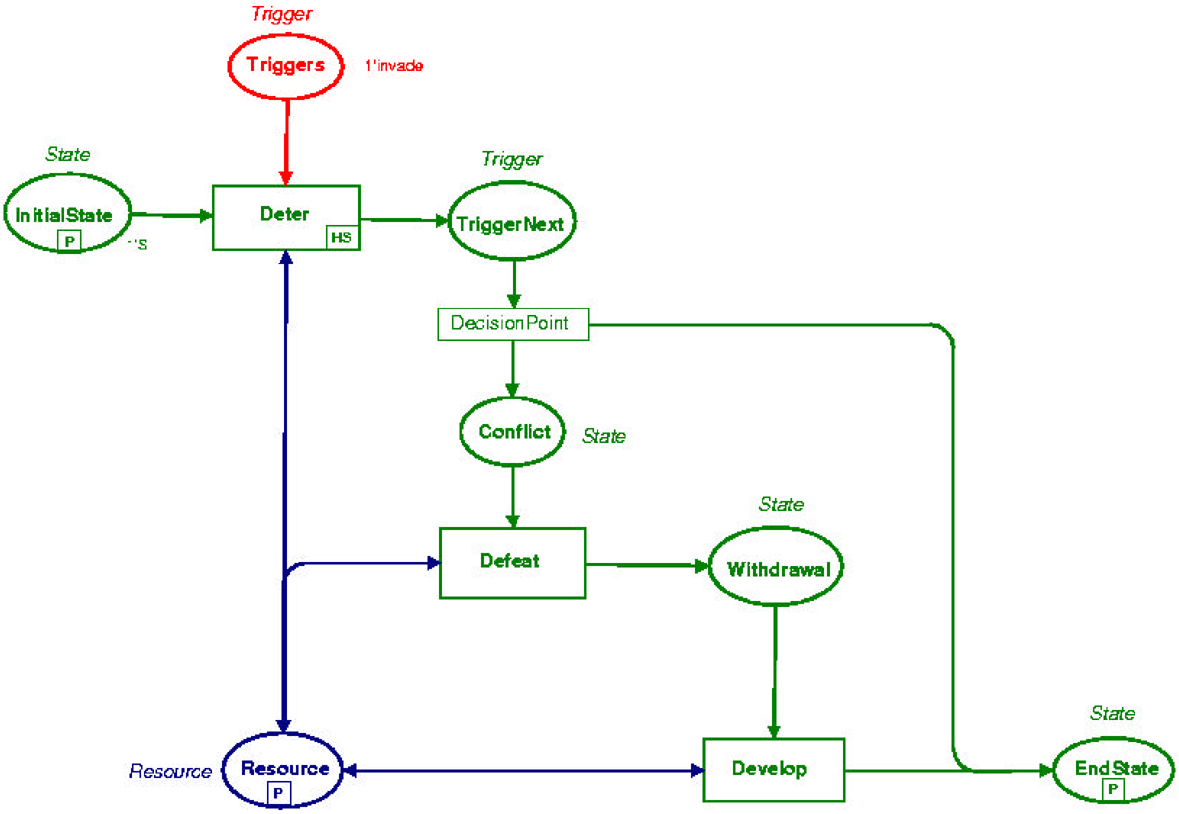

Figure 3 A High Level CPN Model of an Operational Plan.

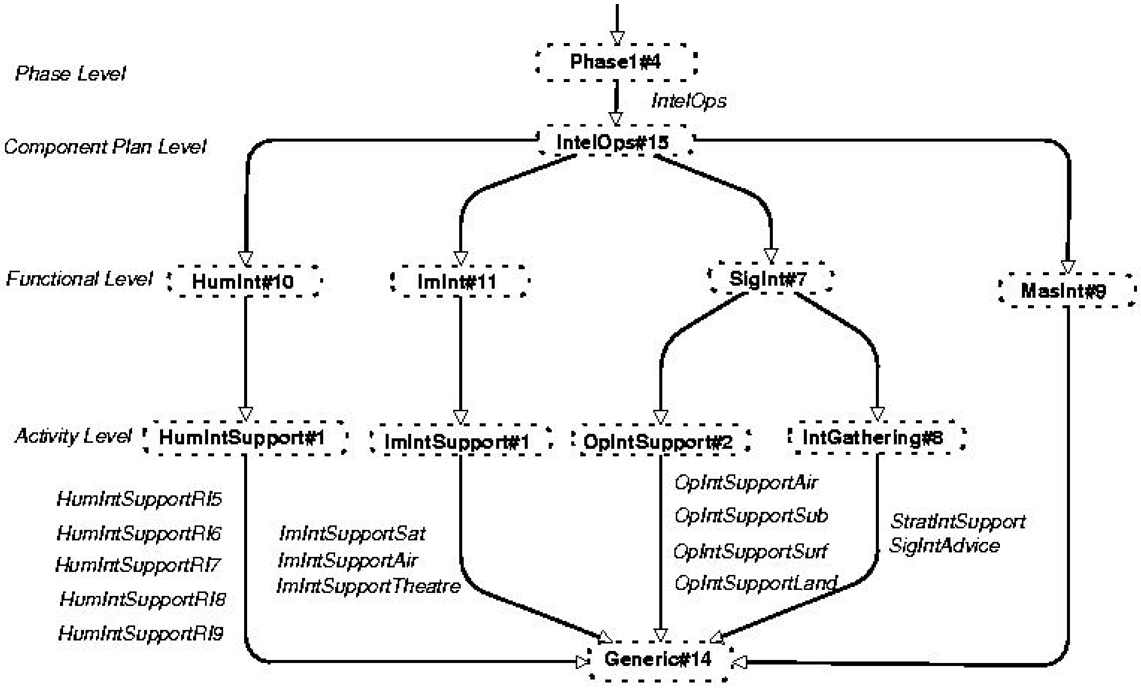

Figure 4 CPN Hierarchy for the Deter Phase (Phase 1)

It is important to note at this point that phases are high-level tasks that can be decomposed into lower-level tasks. Phases are also CPN models, but represented as modules of the top-level CPN model in the form of a hierarchy.

Figure 4 represents part of the hierarchy from *Deter (Phase1)* to a *generic* implementation of intelligence operation tasks. There are five levels in this hierarchy: Phase Level, Component Plan Level, Functional Level, Activity Level and Implementation Level. Due to the volume of the campaign plan, we have only considered Phase1 (*Phase1)* of the plan; and within Phase 1, we have only implemented the component of Intelligence Operations (*IntelOps*). There are four intelligence functions in the intelligence operations component: Human Intelligence (*HumInt*), Imagery Intelligence (*ImInt*), Signal Intelligence (*SigInt*), and Measurement Intelligence (*MasInt*). Similarly, there are four intelligence activities at the Activity Level: Human Intelligence Support (*HumIntSupport*), Imagery Intelligence Support (*ImIntSupport*), Operational Intelligence Support (*OpIntSupport*), and Intelligence Gathering (*IntGathering*). There is only one generic implementation named *Generic*, and the inscriptions on arcs that connect activities to *Generic* indicate the instances (eg, *HumIntSupportRI5*) that call the implementation. These instances are the lowest level Tasks in the Activity modules.

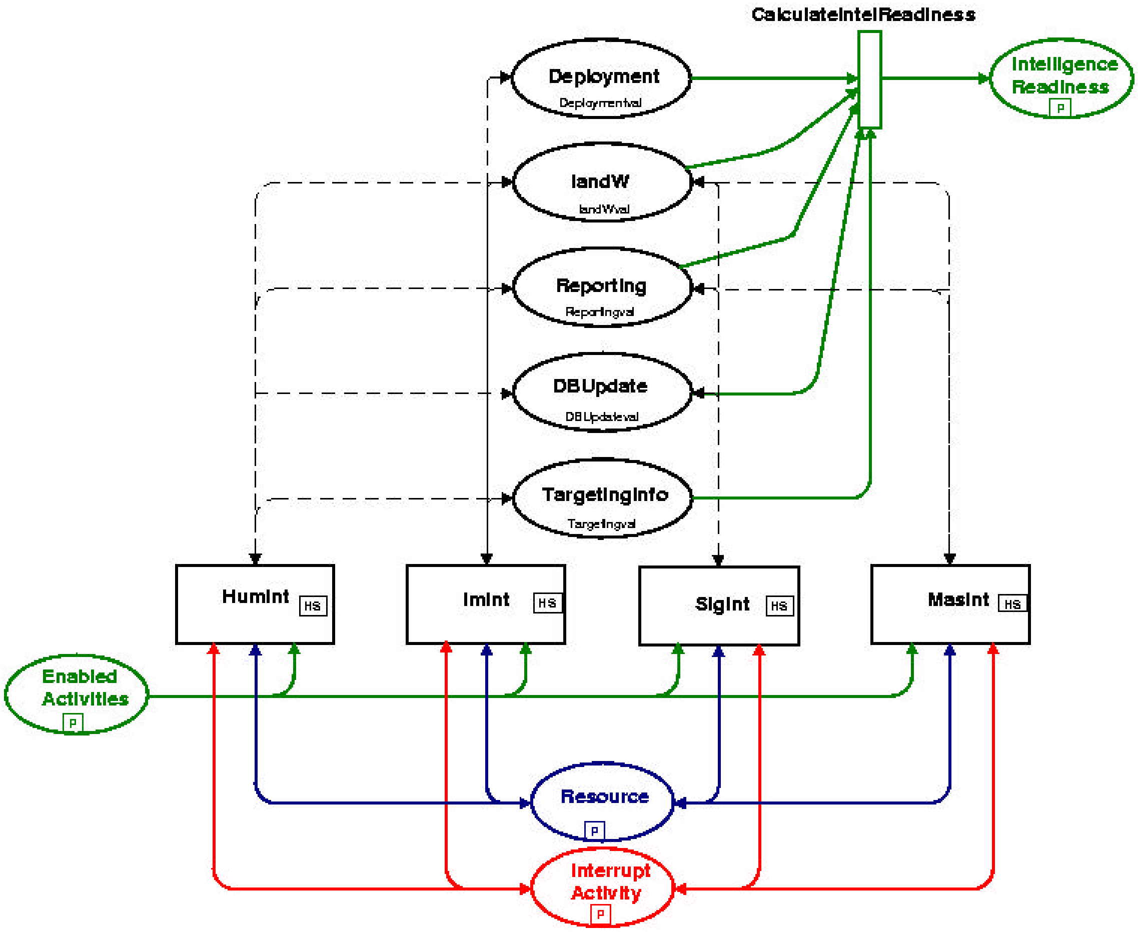

Figure 5 CPN Model of Intelligence Component Plan

Design/CPN [6] is used to model the campaign plan. The ability of Design/CPN to represent a complex system hierarchically makes it convenient to model the plan. The hierarchical structure is facilitated through “Port Places” (noting the port place label *P* in some places of the subsequent models) and substitution transitions (noting the substitution transition label *HS* in some transitions).

The CPN model for the intelligence operations component plan (*IntelOps*) in the *Deter* phase (*Phase1*) is depicted in Figure 5. In this model, the four intelligence functions (*HumInt*, *ImInt*, *SigInt* and *MasInt*) share resources from the *Resource* place. The *Enabled Activities* place holds the tokens of activities that need to be executed by the intelligence functions. These tokens will be checked against the inscriptions of arcs connecting the *Enabled Activities* place to the intelligence functions. As described in Section 2, the arc inscriptions represent preconditions of tasks, and they are implemented in the *Generic* page. The *Interrupt Activity* place holds triggers that interrupt the intelligence functions in case of triggers going off early, eg, invasion or withdrawal.

The effects of the intelligence functions are modelled by the extent to which each of these functions, when executed, affects the level of readiness in five broad categories of readiness for Phase 1: Deployment (*Deployment*), Indicators and Warnings (*IandW*), Operational Reporting (*Reporting*), Intelligence Database Update (*DBUpdate*), and Targeting Information (*TargetingInfo*). When a trigger goes off, execution of the model ensures that each readiness level is updated, and an aggregated *Intelligence Readiness* value is calculated. Another feature of the model is that the resources that are engaged in Phase 1 are made available for use in the following phases of the campaign once the trigger goes off.

Figure 6 shows that execution of the Signal Intelligence (*SigInt*) function affects three of the five readiness indicators: Intelligence Database Update (*DBUpdate*), Indication and Warning (*IandW*) and Operational Reporting (*Reporting*). There are two hierarchical transitions in *SigInt*: Operational Intelligence Support (*OpIntSupport*) and Intelligence Gathering (*IntGathering*). The update of the readiness levels is achieved through two update transitions (*Update* and *Update2*) that take the existing readiness tokens in the form of the *Record* datatype from the three effects places (shown as incoming arc inscriptions *oldReadl1* through to *oldreadl3*), and return updated tokens with a function *update* (outgoing arc inscriptions). There are three interim places (*Completed*, *Interim* and *Completed2*) that temporarily store information on the degree of task completion when a trigger sets off. Figure 6 also shows the structure of resources as a set of records. For example, a maritime resource (*MARRES*) has class, name, functions (*funcslist*), location and notice to move (*ntm*).

In addition to the places of preconditions (*Enabled Activities*), triggers (*Interrupt Activity*) and resources (*Resource*), there is a place (*Config*) that provides configuration for tasks within each of the two compound tasks in this model. A configuration is required as the actual implementation of the tasks within *OpIntSupport* and *IntGathering* calls a generic net model, as shown in Figure 7.

The operational intelligence support task modelled in Figure 7 consists of four lowest level tasks: Air, Subsurface, Surface and Land components of operational intelligence support, represented by *OpIntSupportAir*, *OpIntSupportSub*, *OpIntSupportSurf* and *OpIntSupportLand* transitions, respectively. Each of these transitions is associated with an identification place (eg., *ActivityAir*) that together with the *Configuration* place, determines the behaviour of the task transition when a *Generic* implementation of the task is called.

Figure 8 represents a generic implementation for lowest level intelligence tasks. The purpose of this implementation is to allow tasks to be appropriately interrupted when a trigger goes off before the tasks are completed. There are two features in this implementation. 1. Degrees of task completion are produced as tokens in terms of ratios of the trigger value and specified task durations and are placed in the *Completed* place in order to update the intelligence readiness values as in Figure 5. 2. Resources that are assigned to tasks for specified durations can return to the *Resource* place such that they can be used in the following phases of the campaign.

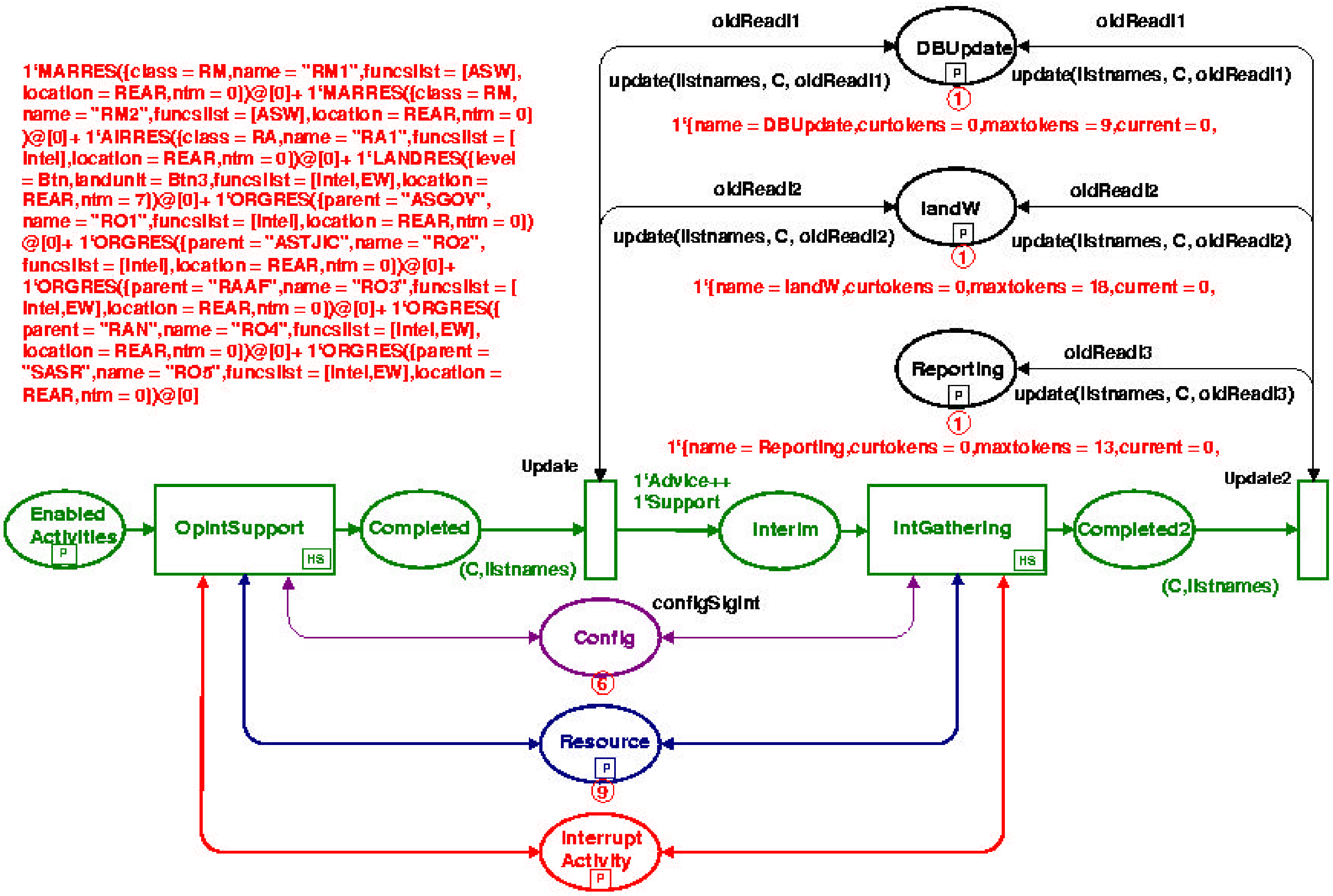
1'MARRES({class = RM,name = "RM1",funcslist = [ASW],
location = REAR,ntm = 0})@[0]+ 1'MARRES({class = RM,
name = "RM2",funcslist = [ASW],location = REAR,ntm = 0]
})@[0]+ 1'AIRRES({class = RA,name = "RA1",funcslist = [
Intel],location = REAR,ntm = 0})@[0]+ 1'LANDRES({level
= Btn,landunit = Btn3,funcslist = [Intel,EW],location =
REAR,ntm = 7})@[0]+ 1'ORGRES({parent = "ASGOV",
name = "RO1",funcslist = [Intel],location = REAR,ntm = 0})
@[0]+ 1'ORGRES({parent = "ASTJIC",name = "RO2",
funcslist = [Intel],location = REAR,ntm = 0})@[0]+
1'ORGRES({parent = "RAAF",name = "RO3",funcslist = [
Intel,EW],location = REAR,ntm = 0})@[0]+ 1'ORGRES({
parent = "RAN",name = "RO4",funcslist = [Intel,EW],
location = REAR,ntm = 0})@[0]+ 1'ORGRES({parent =
"SASR",name = "RO5",funcslist = [Intel,EW],location =
REAR,ntm = 0})@[0]
oldReadl1
oldReadl1
DBUpdate
P
update(listnames, C, oldReadl1)
update(listnames, C, oldReadl1)
1
1'{name = DBUpdate,curtokens = 0,maxtokens = 9,current = 0,
oldReadl2
oldReadl2
landW
P
update(listnames, C, oldReadl2)
update(listnames, C, oldReadl2)
1
1'{name = landW,curtokens = 0,maxtokens = 18,current = 0,
oldReadl3
Reporting
P
update(listnames, C, oldReadl3)
1
1'{name = Reporting,curtokens = 0,maxtokens = 13,current = 0,
Update
Update2
Enabled Activities
P
OpIntSupport
HS
Completed
(C,listnames)
1'Advice++
1'Support
Interim
IntGathering
HS
Completed2
(C,listnames)
configSigInt
Config
6
Resource
P
9
Interrupt Activity
P

Figure 6 CPN Model of Signal Intelligence Function

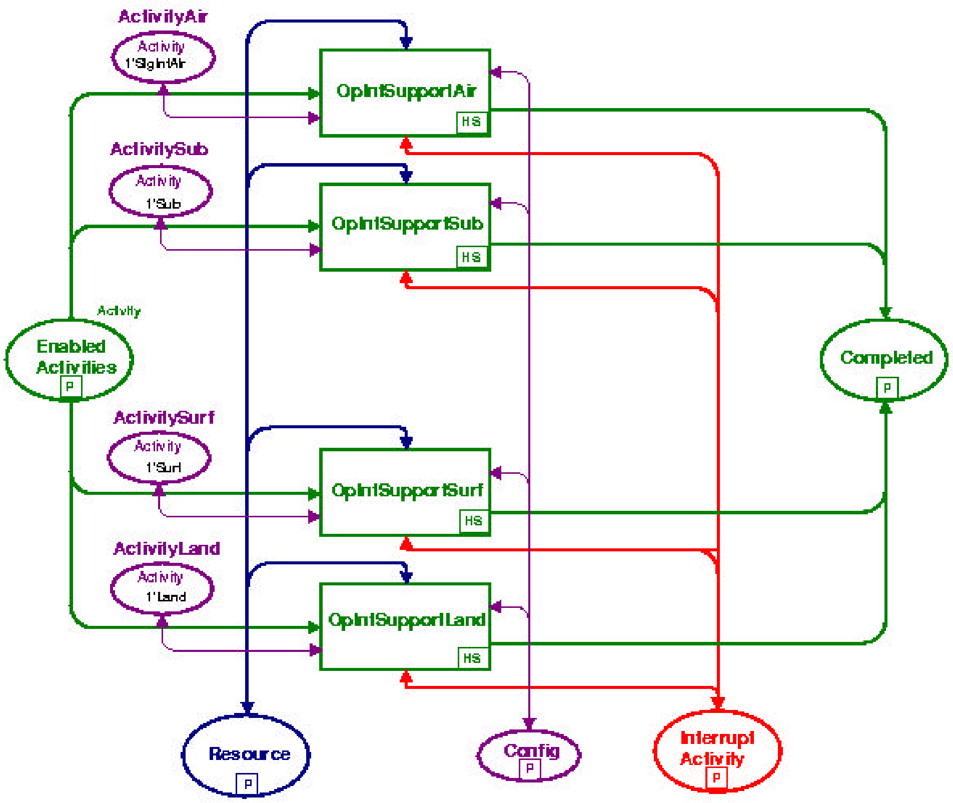

Figure 7 CPN Model of Operational Intelligence Support.

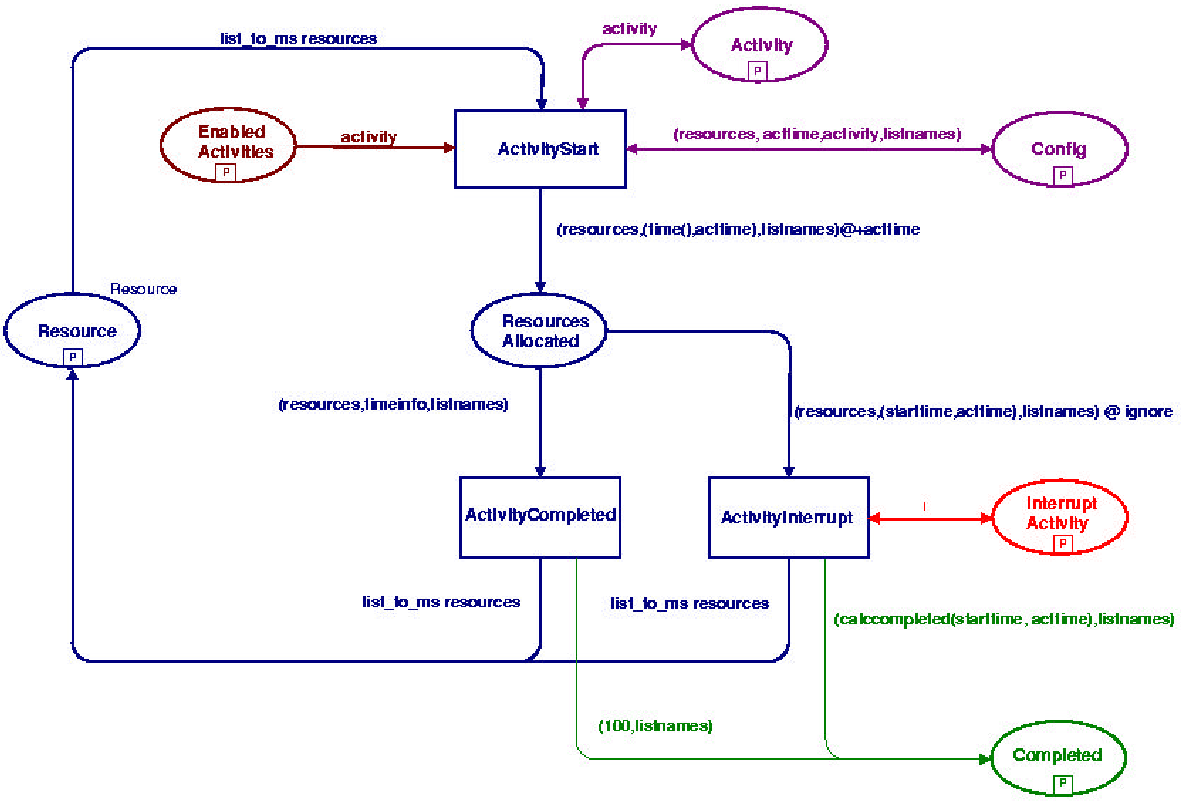

Figure 8 A Generic Implementation for Lowest Level Intelligence Tasks

5. Coloured Petri Net Analysis

One of the analysis questions of interest for the CPN model of the deter phase is to investigate the relationship between the readiness measures and the time at which the trigger sets off. CPN and Design/CPN support two main analysis methods: simulation and state space analysis. The basic idea behind simulation is to make a number of (random) executions of the CPN model and then based on these simulations extract results concerning the properties of the operational plan. Since a set of simulation traces represents only a subset of the possible executions of the operational plan, it cannot be used to obtain definitive answers about properties of the operational plan. The CPN analysis of the operational plan has therefore focused on the use of state spaces. The basic idea behind state space analysis is to compute all reachable states and state changes of the system under consideration (in this case the operational plan) and then use the structure obtained (called the state space) to reason about the operational plan. Since a state space represents all possible executions, definitive answers can be obtained. For the CPN model of the intelligence operations component of the deter phase, state spaces were used to obtain *all* the possible values of the readiness measures for different values of when the trigger sets off. With a simulation-based analysis it would only have been possible to obtain a subset of these. The state space of a complex operational plan is typically very large. To alleviate this complexity problem, we apply partial state spaces. This allows us to ignore parts of the state space that are not relevant for the analysis.

In this section, we present a number of CPN analysis examples that use the model to answer the single question: how ready would our force be at the point when the threat force begins to take hostile action against us? The ability of a commander to answer this question has a profound impact on managing operational risks and on comparing alternative courses of action. A state-space analysis of the CPN campaign model is carried out for each of the possible trigger values ranging from zero to three hundred when all tasks are supposed to be complete, at intervals of ten. As we have only modelled the Intelligence Operations in the *Deter* phase, the following results represent the levels of friendly force readiness in areas of intelligence operations corresponding to different trigger values.

Figure 9 shows the level of readiness in Intelligence Database Update of the *Deter* phase varying as a function of trigger values that represent the time between now and when the threat force commences open hostility that requires military action. As shown in the figure, at each point of the trigger value, the readiness level can take a range of values due to the number of possible ways of executing the tasks. Similarly, Figure 10 and Figure 11 show the levels of readiness in Indicators and Warnings and in Operational Intelligence Reporting, respectively, varying as a function of trigger values.

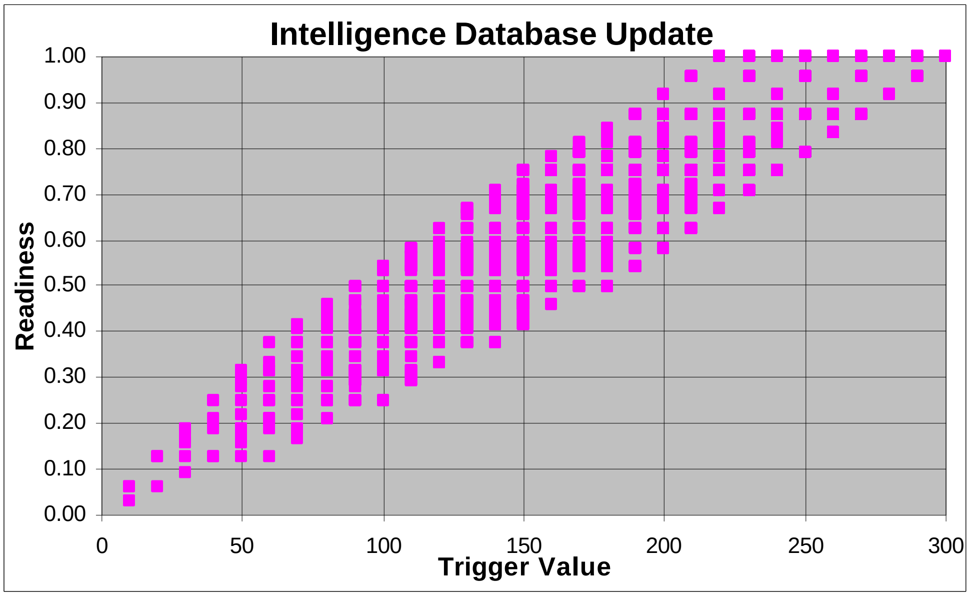

Figure 9 Readiness Level of Intelligence Database Update versus Trigger Values

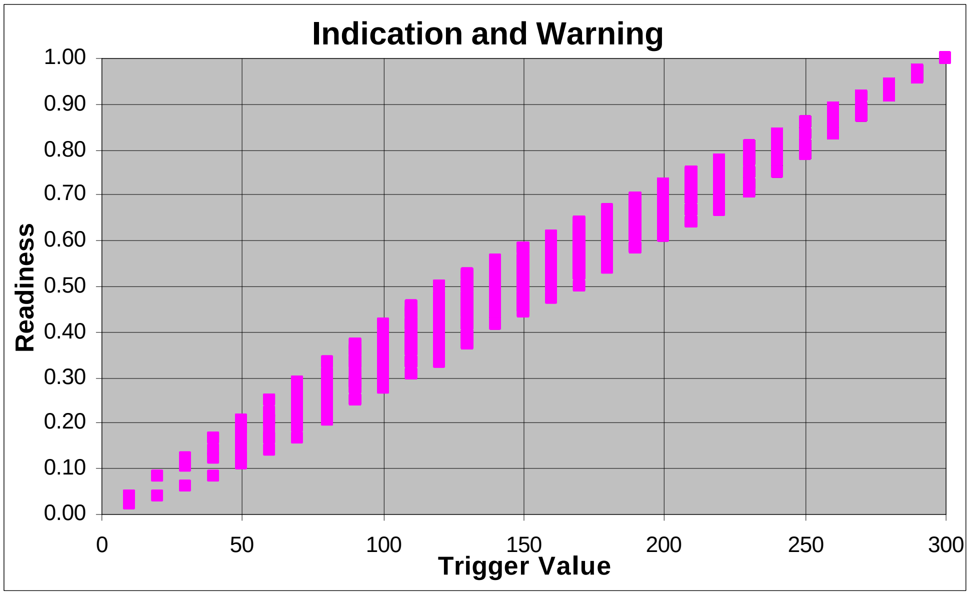

Figure 10 Readiness Level of Indicators and Warnings versus Trigger Values

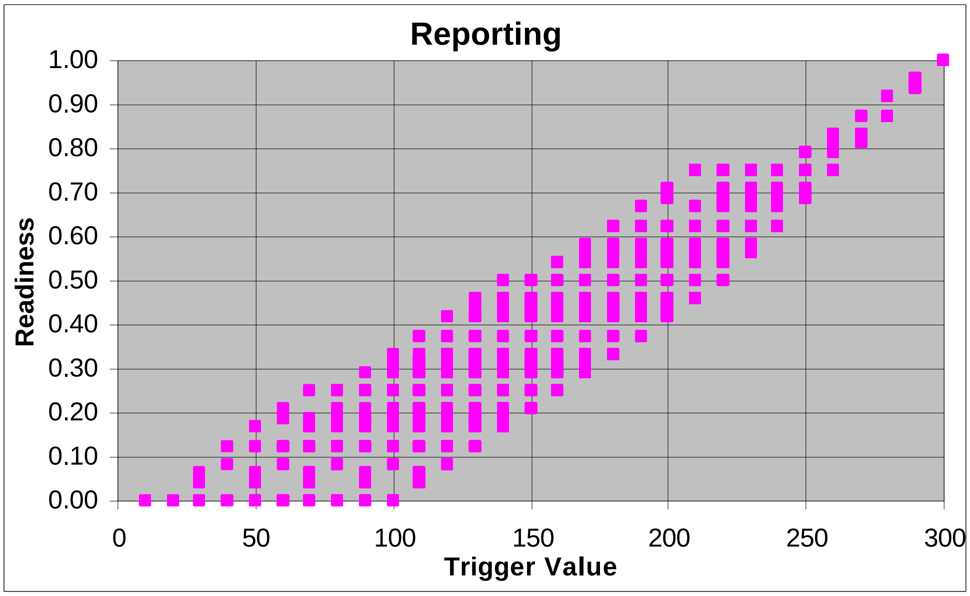

Figure 11 Readiness Level of Operational Intelligence Reporting versus Trigger Values

6. Conclusions and Future Work

We presented in this paper a framework for modelling military campaign plans at the operational level using Coloured Petri Nets (CPN) and Design/CPN. An operational plan is a description of military tasks, in a prescribed (partial) order, that is designed to achieve a desired end state. Central to any operational plan are tasks, which can be modelled as CPN transitions. A task is predicated by pre-conditions corresponding to logical input CPN places, and incoming CPN arc inscriptions. When pre-conditions are satisfied, the actual occurrence of a task (transition) may need to be activated by triggers that can be modelled by additional pairs of input places and incoming arcs and/or a transition guard. Execution of a task produces effects that can be modelled by pairs of CPN output places and outgoing arc inscriptions. A task consumes resources, and this is modelled by CPN places representing resources.

We applied the proposed framework to model the intelligence operation component of the deter phase of an exercise campaign plan. The model is constructed hierarchically, and provides a logical representation of the levels of intelligence tasks that need to be carried out during the deter phase of the campaign. The CPN model of the deter phase consists of fourteen hierarchically structured modules. The CPN model is timed since the time taken to complete activities is an important aspect in the analysis of operational plans. The modules of the CPN model range from modules describing the operational plan at a very abstract level to the most detailed level describing the allocation of resources to activities and the time spent on completing activities with the allocated resources. It was proved during the development of the campaign model that the proposed framework provided a logically convenient way to represent a campaign plan; and some of the structural problems in the original plan were discovered during the modelling process.

We then used the CPN model and conducted a series of state-space analyses to try to answer an important planning question of how prepared would the friendly force be at the point when an adversary commences hostile actions. The CPN model of the deter phase captures all the possible ways in which the operational plan for the deter phase can be executed, i.e., all possible orderings in which resources can be allocated for the different activities. Because of this, at each point of trigger value, there were a range of readiness levels resulting from different possible orderings of tasks and resource allocations. Our analysis can potentially identify a path in all possible orderings that has the highest level of readiness at a point of trigger value as dictated by the operational commander. The problem of optimisation can be investigated in the future using state space analysis, possibly combined with the use of a recently developed state space method [12] allowing us to consider only small segments of the state space at a time and yet guarantee that the properties of interest can be determined.

7. References

[1] J. V. Jensen. An Introduction to Bayesian Networks. London UCL Press, 1996, pages 48-52.

[2] L. Zhang, L. Falzon, M. Davies and I. Fuss. On Relationships between Key Concepts of Operational Level Planning, Proceeding, 5th International Command and Control Research and Technology Symposium, Australia War Memorial, Canberra ACT, Australia, 24-26 October 2000.

[3] L. Falzon, L. Zhang and M. Davies. A Policy Analysis Approach to Operational Level Course of Action Analysis, Proceeding, 5th International Command and Control Research and Technology Symposium, Australia War Memorial, Canberra ACT, Australia, 24-26 October 2000.

[4] K. Jensen. Coloured Petri Nets. Basic Concepts, Analysis Methods, and Practical Use. Volumes 1-3. Monographs in Theoretical Computer Science. Springer Verlag 1992-1997.

[5] L.M. Kristensen and S. Christensen and K. Jensen. The Practitioner's Guide to Coloured Petri Nets. In [8] pages 98-132

[6] Design/CPN Online. www.daimi.au.dk/designCPN.

[7] J. Billington, M. Diaz and G. Rozenberg. Application of Petri Nets to Communication Networks, Lecture Notes in Computer Science, Vol.1605, Springer-Verlag, 1999.

[8] K. Jensen (Ed.) International Journal on Software Tools for Technology Transfer. Special Section on Coloured Petri Nets, Volume 2, Issue 2, December 1998. Springer Verlag.

[9] Perdu, D.M. & A.H. Levis. A Petri Net Model for Evaluation of Expert Systems in Organizations, Automatica, Vol. 27, No. 2, March 1991.

[10] L. W. Wagenhals, I. Shin, and A. H. Levis. Creating Executable Models of Influence Nets with Coloured Petri Nets. In [8] pages 168-181.

[11] *Joint Military Appreciation Process*. Australian Defence Force Publication, Operations Series, ADFP 9, Joint Planning, Chapter 8, 1999.

[12] S. Christensen and L.M. Kristensen and T. Mailund. A Sweep-Line Method for State Space Exploration. Proceedings of *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2001.