Submission for the 9th International Command and Control Research and Technology Symposium

Topic	C2 Assessment Tools & Metrics	
Title	COAST – An Operational Planning Tool for Course of Action Development and Analysis	
Authors	Dr Lin Zhang Command and Control Division Defence Science and Technology Organisation PO Box 1500, Edinburgh SA, 5111, Australia Dr Lars Michael Kristensen Department of Computer Science University of Aarhus IT-parken, Aabogade 34 DK-8200 Aarhus N, Denmark Mr Brice Mitchell Command and Control Division Defence Science and Technology Organisation PO Box 1500, Edinburgh SA, 5111, Australia Mr Guy Gallasch Computer Systems Engineering Centre School of Electrical and Information Engineering University of South Australia Mawson Lakes SA 5095, Australia Mr Peter Mechlenborg Department of Computer Science University of Aarhus IT-parken, Aabogade 34 DK-8200 Aarhus N, Denmark Dr Chris Janczura Command and Control Division Defence Science and Technology Organisation PO Box 1500, Edinburgh SA, 5111, Australia	Telephone: +61 8 8259 5501 Fax: +61 8 8259 5619 Email: Lin.Zhang@dsto.defence.gov.au Telephone: +45 8942 5686 Fax: +45 8942 5624 Email: kris@daimi.au.dk Telephone: +61 8 8259 5685 Fax: +61 8 8259 5619 Email: Brice.Mitchell@dsto.defence.gov.au Telephone: +61 8 8302 3832 Fax: +61 8 8302 3384 Email: Guy.Gallasch@postgrads.unisa.edu.au Telephone: +45 8942 5635 Fax: +45 8942 5624 Email: metch@daimi.au.dk Telephone: +61 8 8259 6744 Fax: +61 8 8259 5619 Email: Chris.Janczura@dsto.defence.gov.au
Point of Contact	Brice Mitchell	

COAST – An Operational Planning Tool for Course of Action Development and Analysis

Lin Zhang, Brice Mitchell and Chris Janczura
Command and Control Division
Defence Science and Technology Organisation
PO Box 1500, Edinburgh, SA 5111, AUSTRALIA

Guy Gallasch
Computer Systems Engineering Centre
School of Electrical and Information Engineering
University of South Australia
Mawson Lakes SA 5095, AUSTRALIA

Lars M. Kristensen and Peter Mechlenborg
Department of Computer Science
University of Aarhus
IT-parken, Aabogade 34, DK-8200, Aarhus N, DENMARK

Abstract

This paper presents a Course Of Action Scheduling Tool (COAST), a military operational planning tool for course of action (COA) development and analysis. COAST supports automated sequencing and scheduling of military tasks developed during an operational planning process. It is desirable that task sequences and schedules, also known as lines of operation (LOPs), are suitable and feasible. An LOP is suitable when the execution of the task sequence logically leads to the achievement of a predefined end state, and feasible when all tasks are resourced, and conflicts of resource requirements by the tasks are preempted. The properties of suitability and feasibility of an LOP generated with COAST are proved through the use of a Coloured Petri Net (CPN) model of COAs and the associated formal analysis methods.

COAST has a client-server architecture that consists of a JAVA graphical user interface (Client) and an underlying CPN model (Server) that provides a semantic foundation for COAST. CPNs are a mathematically rigorous modelling language, and the core analysis capabilities of COAST are based on the analysis methods of CPNs. The use of the CPN model and associated analysis techniques is transparent to the user. This is important since the users are not expected to be able to use CPNs.

1. Introduction

Operational planning is concerned with the design, organisation, sequencing and scheduling of military campaigns and major operations to achieve strategic objectives. There exist doctrinal processes that guide commanders and planning staff to develop operational plans. Common to many planning processes is the following procedure.

- The planners identify a set of *conditions* that constitute the strategic objectives. This set of conditions is also called an *end state*.
- The planners decide on the operational objectives that must be achieved to reach the desired end-state. The set of conditions that constitute the

operational objectives are the desired *effects* of military operations. Military *tasks* are then developed to achieve the desired effects. The tasks must also be allocated appropriate *resources*, and coordinated with other tasks and events. Some tasks may require certain conditions to be met for them to be prosecuted, and the required conditions may in turn be the effects of other tasks to be developed. The effects/conditions, tasks, and resources must be organised into *lines of operation* to achieve the end state. We consider lines of operation as detailed representations of *courses of action* (COAs).

- The COAs are analysed for improvement. Wargames and quantitative analysis are conducted at this stage to provide commanders with assessments to assist COA selection.
- Decisions are made on the selection of a COA, and the selected COA is executed. The situation is continuously monitored, and a new situation may trigger another round of operational planning, i.e. dynamic re-planning.

Operational planning presents a major challenge to the military. Planning is often conducted in the presence of time pressure and uncertainty, and in large, distributed groups. Tens or sometimes hundreds of military tasks are developed, resourced, sequenced, and scheduled for the achievement of conditions or effects, and ultimately the desired end state in order to attain strategic objectives. There exists complex causal interdependency among tasks induced by effects and conditions; there are precedence ordering and temporal constraints between tasks that must be satisfied in lines of operation; and finally there are potential conflicts in resource requirements that must be addressed for any feasible line of operation. All these factors can make planning a very complex task.

Computer systems have been developed to support operational planning. The emphasis, however, has been in the areas of collaborative planning tools, data and information management, and tools to facilitate situation awareness. Among decision support systems for planning, the Centre Of Gravity Network Effects Tool(COGNET) [Priest, 2002] is a strategy development tool that supports the modelling and analysis of relationships between such elements as capabilities and requirements fundamental to operational planning for the identification and prioritisation of critical elements. CAESAR II/EB [Wagenhals, 1998] is another planning support system that aims at analysing timing and probability profiles of COAs based on static influence models of tasks incorporating temporal information. Concepts and techniques for sequencing and scheduling of military tasks have been investigated in the past [Mulvehill and Caroli, 2000], but none of them has become operational. The predominant planning support system used in many military headquarters has been MS Project. MS Project is not a planning tool as it only presents pre-defined schedules of tasks with pre-defined views. The representation of a plan (schedule) in MS Project does not allow it to be analysed with more rigorous methods.

This paper presents a Course Of Action Scheduling Tool (COAST), a COA development and analysis tool for operational planning. COAST supports the development, resourcing, sequencing and scheduling of tasks within planning. The key question that COAST intends to answer is this: given a list of tasks that are designed by military planners developing a COA, is there one or more lines of operation that when executed, will lead to the achievement of a mission without violating the constraint of given resources? The essence of COAST is a conceptual representation of the military planning domain, formalised with a dynamic modelling language called Coloured Petri Nets (CPNs). The CPN model of the planning domain is amenable to formal analysis methods that can help provide answers to planning questions such as the existence of suitable and feasible COAs. The formal representation can also be used for quantitative analysis of COA probability of success, risks and costs.

This paper is presented as follows. Section 2 describes the conceptual model of the planning domain through the use of a simple planning example. Section 3 explains the formal CPN model of the domain and analysis techniques for sequencing and scheduling of tasks in COAs. Section 4 presents the COAST tool, its design and capabilities, through the use of the example introduced in Section 2. Discussions and ideas of future work are provided in Section 5.

2. Conceptual Modelling of the Planning Domain

This section describes the conceptual model of the planning domain. A simple planning problem is used to illustrate the conceptual model. The model is based on previous work [Zhang et al, 2001] and [Zhang et al, 2002].

A typical operational planning problem comprises four major elements: a desired end state, current conditions, available resources, and limitations. As discussed in the previous section, an end state comprises a set of conditions to be achieved in order to attain strategic objectives. Current conditions are the set of initially valid conditions. One may associate a measure of uncertainty, i.e. probabilities, to current conditions. For example, a condition that is true with certainty has a probability value of 1; and a condition that is probably true has a probability value between 0 and 1. In military terms, they are called *facts* and *assumptions*. There can be political, social, economical, military and physical (e.g. weather) conditions, etc., in a planning problem. The available resources describe primarily military assets and force elements, for example, aircraft, ships and troops. They may also include non-military resources such as law enforcement agencies. Limitations specify the parameters within which operations are planned. Limitations can be categorised into restrictions and constraints. Restrictions are imposed from a superior commander, while constraints are physical characteristics that cannot be changed that affect the conduct of operations.

Let us consider a fictitious and simple problem. A friendly force commander is assigned a mission to recover an island from the occupation of an opposition force through an amphibious operation. The desired end state for the friendly force commander is "Amphibious forces successfully landed". The current conditions include the force dispositions and their readiness levels. The available resources include the force elements that are assigned to this mission, including troops, aircraft, and ships of different types. The detailed resource list is discussed later in this section. The limitations include the restrictions of not trespassing in third party airspace and territorial waters, and the constraints of certain weather conditions that are not favourable for conducting amphibious operations.

Given the end state, and the guidance of conducting an amphibious operation to achieve the end state, the commander and the planers will then consider the conditions that must be established in order to achieve the end state through executing the task of conducting an amphibious operation, and the resources required to execute the task. For instance, the typically required conditions for an amphibious operation include the following.

- Local air control established
- Local sea surface control established
- Local sea sub-surface control established
- En route sea mines cleared
- Point of entry (POE) established

The planners must also consider necessary resources that are required to execute the task of the amphibious operation. We consider for this example the following resources: three types of ships (2 LPA[1], 1 LSH[2], 6 LCH[3]) and three battalions of troops (3 BN). These resources must be a subset of the available resources, and some of them may be lost depending on the level of risk of the operation. Other parameters of the task include the intended start time, the intended duration, the probability of success given the conditions and resources. The specification of the amphibious assault task is illustrated in Figure 1.

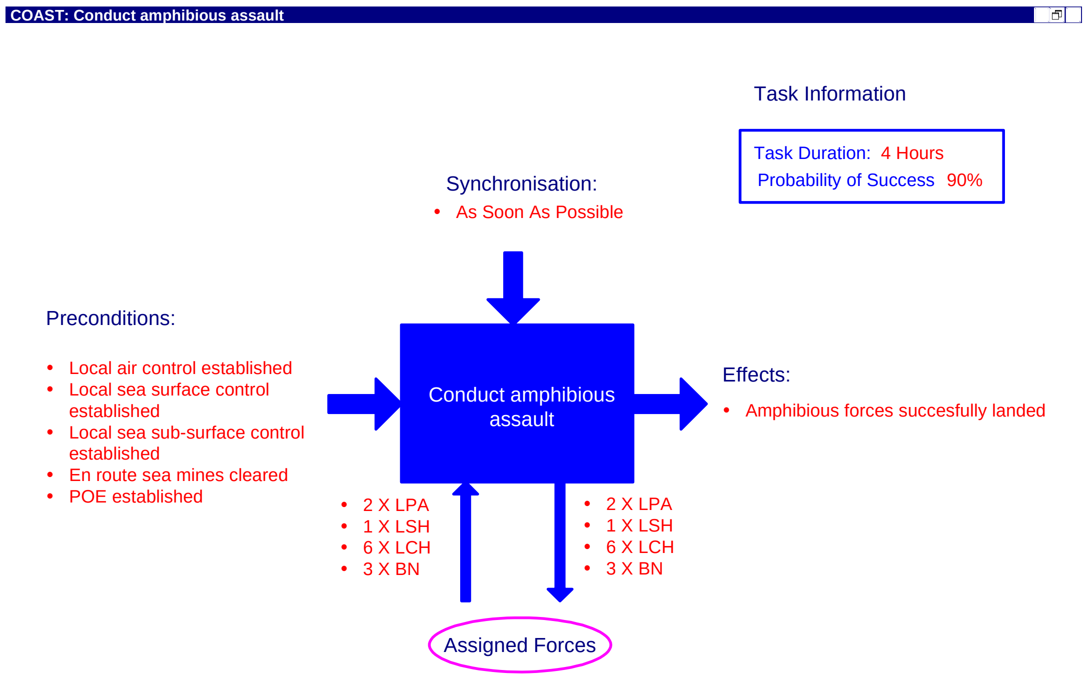

COAST: Conduct amphibious assault
Task Information
Task Duration: 4 Hours
Probability of Success 90%
Synchronisation:
• As Soon As Possible
Preconditions:
• Local air control established
• Local sea surface control established
• Local sea sub-surface control established
• En route sea mines cleared
• POE established
Conduct amphibious assault
Effects:
• Amphibious forces succesfully landed
• 2 X LPA
• 1 X LSH
• 6 X LCH
• 3 X BN
• 2 X LPA
• 1 X LSH
• 6 X LCH
• 3 X BN
Assigned Forces

Figure 1: The conduct amphibious assault task

We observe that the conditions necessary for the amphibious operation must be established through the design and prosecution of other tasks. For example, the establishment of local air control may be achieved through the conduct of a combat air patrol (CAP) operation; it may also be achieved through offensive counter air

[1] Landing Platform Amphibious
[2] Landing Ship Heavy
[3] Landing Craft Heavy

(OCA) operations; and either of these methods (tasks) will require further conditions to be satisfied, and resources to be present (that may be subject to losses). This way, the tasks are causally related through effects and conditions; they draw resources from the same pool; and may be temporally constrained by each other. We consider this method of planning consistent with the Effects Based Planning concept. With this approach, assume that the planners consider the following tasks.

- Conduct amphibious assault
- Conduct combat air patrol
- Conduct Anti-Submarine Warfare (ASW) operations in the Area of Operation (AO)
- Conduct airborne operations
- Conduct maritime escort operation
- Conduct mine clearance operation
- Establish Forward Operating Base (FOB)
- Establish Forward Arming & Refueling Point (FARP)
- Provide Air to Air Refueling (AAR)

The problem is to find suitable sequences of tasks that logically achieve the end state, and feasible schedules of execution that pre-empt conflicts of requirements for resources. The premises of COAST are that if the planners have specified the other 8 tasks in the above list to the same level of detail as the task of conducting amphibious operation, a formal representation of the planning domain can be obtained and mathematical analysis methods can be applied to obtain answers to the planning problem, i.e. suitable and feasible COAs. The detail of the formalism is described in the next section. We generalise the task specification in Figure 1 into a task schema below.

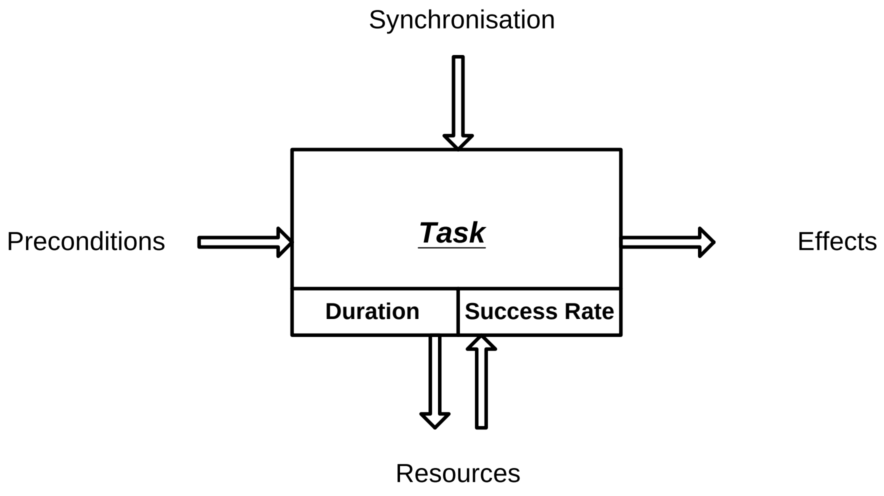


Figure 2: COAST task schema

A task is defined by its preconditions, effects, synchronisation (timing and precedence information), resources (required resources and possible lost resources due to attrition, for example), duration and the success rate. Note that we now use the term *preconditions* to denote those conditions that must be met for a task to execute, and use the term conditions to mean both preconditions and effects.

It is important to model and analyse temporal aspects of COAs, hence the temporal properties of a task must be extensively represented in the model. We allow resources in Figure 2 to have intervals of availability, and specify 4 types of preconditions and 5 types of effects in accordance with the temporal behaviour and task outcomes (success, failure and abort). The types of preconditions and effects are given below, and illustrated in Figure 3.

Preconditions

- *Normal Precondition (NP)*: giving the conditions that need to be fulfilled in order for the task to start.
- *Sustaining Precondition (SP)*: giving the conditions that must be satisfied for the task to be executed. A task will not start unless all its sustaining preconditions are satisfied. If one of the sustaining conditions becomes invalid while the task is executing, the task will have to be *aborted*.
- *Termination Precondition (TP)*: giving the conditions that have to be satisfied for the task to be terminated.
- *Vanishing Precondition (VP)*: giving the subset of the normal preconditions that become invalid when the task starts to execute.

Effects

- *Instant Effect (IE)*: giving the conditions that become valid when the task begins to execute.
- *Post Effect (PE)*: giving the conditions that become valid upon a successful completion of the task, i.e. normal termination of the task.
- *Sustained Effect (SE)*: giving the conditions that are valid as long as the task is executing. When the task terminates, the sustained effects will no longer be valid.
- *Abortion Effect (AE)*: giving the conditions that become valid if the task aborts.
- *Failure Effect (FE)*: giving the conditions that become valid if the task fails.

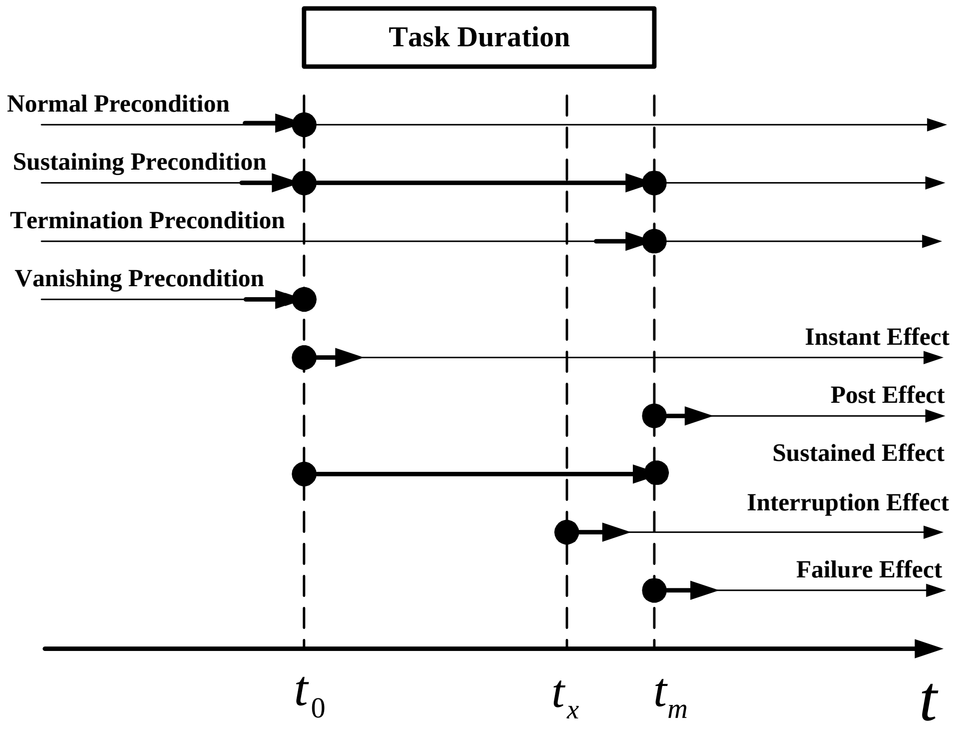


Figure 3: Types of preconditions and effects

The fully developed tasks are given in Table 1. Note that the specifications of task attributes other than the task name are intentionally optional. In Table 1, two tasks do not need preconditions, and the lost resources and durations are specified for only some of the tasks.

	Task name	Preconditions	Effects	Resources	Lost Res.	Duration	Sync. Info.
T1	Conduct amphibious assault	•Local air control established (SP) •Local sea surface control established (SP) •Local sea sub-surface control established (SP) •En route sea mines cleared (NP) •POE established (NP)	•Amphibious forces successfully landed (PE)	2 LPA 1 LSH 6 LCH 3 BN		4 Hours	As Soon As Possible
T2	Conduct combat air patrol	•FOB established (NP) •Fighter aircraft deployed to the AO (NP) •En route refueling provided (SP)	•Local air control established (SE)	12 FA 18		As required	
T3	Conduct ASW operations in the AO	•FOB established (NP) •Local air control established (SP)	•Local sea sub-surface control established (SE)	2 MPA		As required	
T4	Conduct airborne operations	•Local air control established (SP) •FOB established (NP) •FARP established (NP)	•POE established (PE)	12 Blackhawk 2 ABN BN	2 Blackhawk	8 Hours	
T5	Conduct maritime escort operation	•Local air control established (SP)	•Local sea surface control established (SE)	4 FFH		As required	
T6	Conduct mine clearance operation	•Local air control established (SP) •Local sea surface control established (SP) •Local sea sub-surface control established (SP)	•En route sea mines cleared (PE)	•4 Mine Hunters		48 Hours	
T7	Establish FOB		•FOB established (PE)	•1 ECSS		60 Hours	
T8	Establish FARP		•FARP established (PE)	•1 Eng Coy		40 Hours	
T9	Provide AAR	•FOB established (NP) •AAR aircraft deployed to the AO (NP)	•En route refueling provided (SE)	•4 AAR		As required	

Table 1: Task details of an operational planning example

3. Modelling and Analysis of COAs

The CPN modelling language is aimed at modelling systems that can be viewed as concurrent systems and where synchronisation and resource allocation are the key elements [Jensen, 1992]. A CPN model of a system is both action and state oriented in that it captures the current state of the system and the possible event that may occur in a given state. CPN also includes a time concept which makes it possible to describe the time taken by events in the system. This section provides an overview of the CPN model and analysis algorithms that provide the formal semantic foundation of COAST. The CPN model formally capturing the execution of tasks according to their attributes, set of currently valid conditions and available resources is presented in Section 3.1. The analysis algorithms for computing feasible and suitable LOPs are presented in Section3.2.

3.1 CPN Modelling of COAs

The purpose of the CPN model is to formally capture the semantics of task execution in COAs. The basic idea in COAST is to use a CPN model for modelling the execution of tasks according to the preconditions and effects of tasks, imposed synchronisations, and available resources. In the case of the COAST framework, the state of the system corresponds to the set of tasks (which may be idle, executing or done), the set of currently valid conditions, and the set of currently available resources. The events that may happen in a given state correspond to the start and termination of tasks, change of the valid conditions, and change of available resources. The CPN model captures how the state of the system changes when these events occur.

Figure 4 shows the hierarchy page of the CPN model. The hierarchy page provides an overview of the pages (modules) constituting the CPN model and their relationship. Each node in Figure 4 represents a page in the CPN model, and is labelled with a page name and a page number. As an example, the page node at the top of Figure 4 is named CoastServer and has page number 20. Page CoastServer is the most abstract page in the CPN model and constitutes the top-level of the CPN model. An arc between two nodes indicates that the destination page is a subpage (submodule) of the source page.

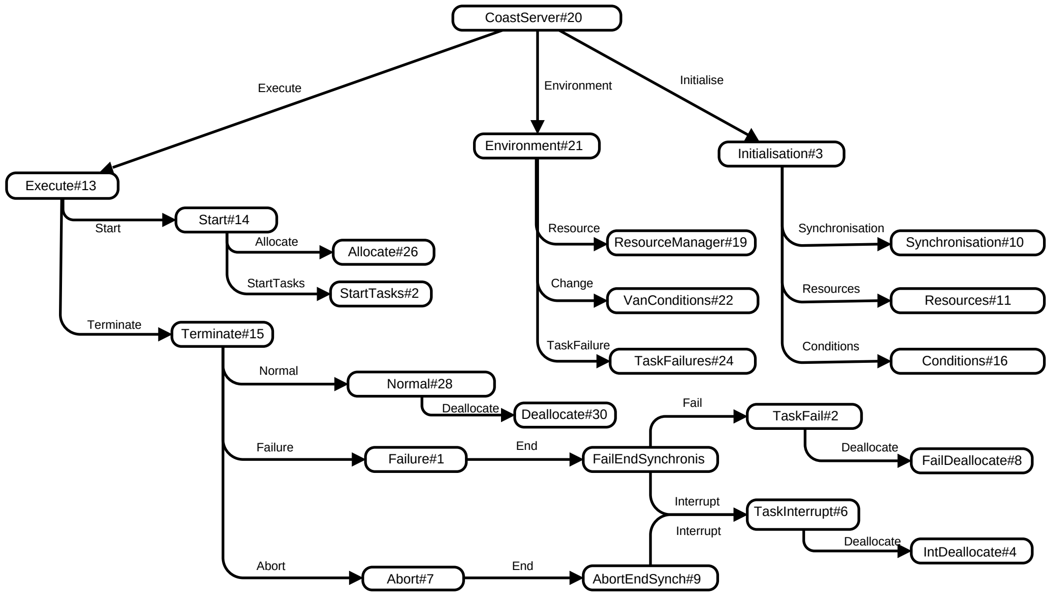


Figure 4: Overview of the CPN model.

The CPN model consists of three main parts.

- Page Execute and its 14 subpages model the execution of tasks, i.e, start, termination, abortion, and failure of tasks according to the set of tasks, resources, conditions, and synchronisation in the plan.
- Page Environment and its 3 subpages model the environment in which tasks execute, and is responsible for managing the availability of resources over time, change of conditions over time, and task failures.
- Page Initialisation and its 3 subpages are used for the initialisation of the model according to the concrete set of tasks, conditions, synchronisations, and resources in the COA. The CPN model has been parameterised with respect to the set of tasks, resources, conditions, and task synchronisations. This ensures that a given set of tasks, conditions, resources, and task synchronisations can be analysed by setting the initial marking (initial state) of the CPN model accordingly, i.e. no changes to the structure of the CPN model are required to analyse different COAs.

Figure 5 shows the top level page of the CPN model with the three main parts of the CPN model represented as the substitution transitions (drawn as boxes): Initialise, Execute, and Environment. Each of these substitution transitions has associated subpages that describe the details of the task execution, the environment, and the initialisation, respectively. The state of a CPN model is represented by means of places (drawn as ellipses), and corresponds to the distribution of tokens (called a marking) on the places of the CPN model. There are five places in Figure 5. The Resources place is used to model the available resources in a given state, the Tasks place is used to model the tasks that are yet to be executed, the Execute Status place is used to model the task currently being executed, and the Conditions place is used to model the current set of valid conditions. Place Init is used to trigger the initialisation of the CPN model. Figure 5 depicts a marking for a sample plan with six tasks. There are two tokens on the Resources place, six tokens on the Tasks place, and one token on the Conditions place. The values carried by these tokens describe the state of the system. The two tokens on the Resources place carry data values that capture the current set of available and busy resources, respectively; the six tokens on the Tasks place have values that corresponds to the six tasks in the COA, and the tokens on the Conditions place have a data value which is a list describing the truth value of each of the six conditions (E1 to E6) in the COA. The dynamics of the CPN model corresponds to the execution/occurrences of transitions that change the marking of the CPN model.

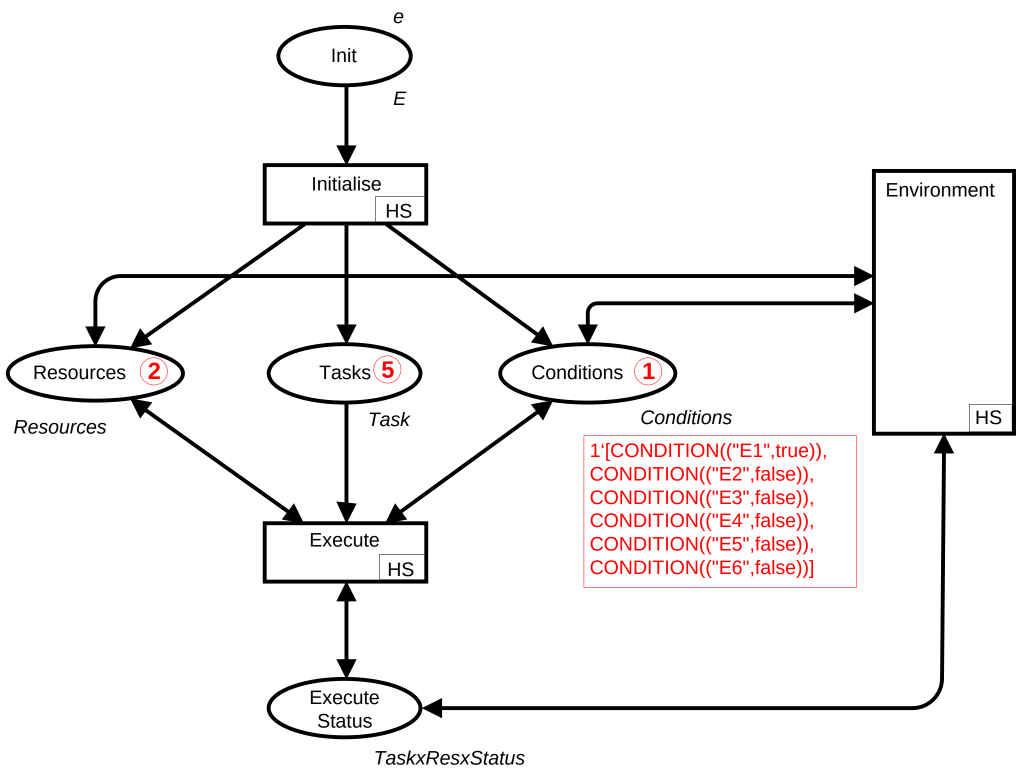


Figure 5: Page CoastServer – top level of the CPN model.

3.2 Analysis Methods

The main analysis capability of COAST is the generation of feasible and suitable LOPs. A LOP is a specification of start and end times for the tasks in the COA. The LOP generation is based on constructing the state space of the CPN model when initialised with the set of tasks, resources, and conditions that constitute the COA. The state space of a CPN model is a directed graph where the set of nodes corresponds to the set of states reachable from the initial state of the CPN model and the arcs correspond to occurring events. The state space hence has a node for each state that can be reached by occurrences of transitions starting from the initial state. There is an arc corresponding to an event b between two nodes n1 and n2, if the event b may occur in the state represented by n1 and if the occurrence of the event b in n1 causes the system to enter the state represented by n2. Paths in the state space hence correspond to the executions of the CPN model, and the state space of a CPN model represents all the possible executions of the CPN model and hence all the possible orderings in which tasks can be executed in the given COA.

Figure 6 shows a sample state space with the node 1 representing the initial state and node 21 representing a desired end-state. The labels on the arcs represent the start and termination of tasks (T1-T6) and the number after the colon is the time at which the event occurs. Arcs without labels represent internal events in the CPN model. The LOPs corresponds to the possible ways to reach state 21 from state 1. The LOPs are obtained through generating a state space for the CPN model and extracting paths in the state space leading from the initial marking to markings representing end-states.

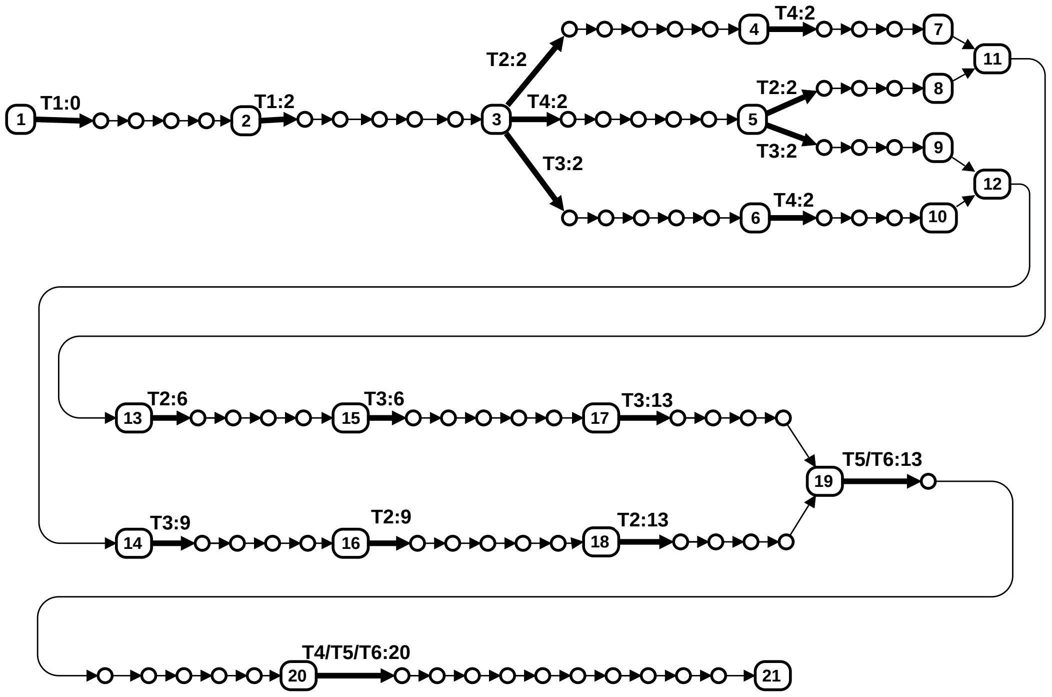


Figure 6: State space of an example plan.

The LOP generation algorithm in COAST consists of two phases. The first phase is a depth-first generation of the state space where complete and incomplete LOPs are reported as they are found. The second phase is a breadth-first traversal of the state space where the LOPs not found in the first phase are computed. The second phase is required since not all LOPs may be reported by the depth-first generation. The LOP generation algorithm is complete in that it will report all the possible LOPs.

The LOP generation algorithm has been implemented in such a way that it is possible to terminate the LOP generation when a certain number of LOPs (specified by the user) has been found, as it is not always possible to compute all the LOPs given time pressures. The fact that the first phase proceeds in a depth-first manner means that LOPs can be found without having to generate the full state space. It is possible to combine the depth-first generation with various kinds of heuristics search techniques.

4. Course Of Action Scheduling Tool (COAST)

This section presents the COAST tool that implements the COA model and analysis methods described in the previous two sections. Section 4.1 introduces the design of COAST focussing on the Client/Server architecture. Section 4.2 shows how COAST is used to develop LOPs for the simple planning problem introduced in Section 2.

4.1 COAST Design

Figure 7 shows the client-server software architecture of COAST. The COAST client, including a domain-specific graphical user interface (GUI) as seen in Figure 8, is implemented in Java, whereas the COAST server is implemented in Standard ML(SML) via the embedded CPN model that forms the core of the COAST server. Communication between the client and the server is based on Comms/CPN and Comms/JAVA [Gallasch and Kristensen, 2001], a library supporting TCP/IP communication between CPN models and external applications. An *SML session* layer has been implemented on top of Comms/CPN. This layer allows the client to invoke functions available in the server and receive the corresponding results. The *SML Session* layer is implemented by allowing the client to submit SML code to the server for evaluation. The received SML code is then executed (evaluated) by the server, and results are sent back to the client. The SML code sent to the server corresponds to the invocation of the SML functions made available by the server through the *COA Analysis* module.

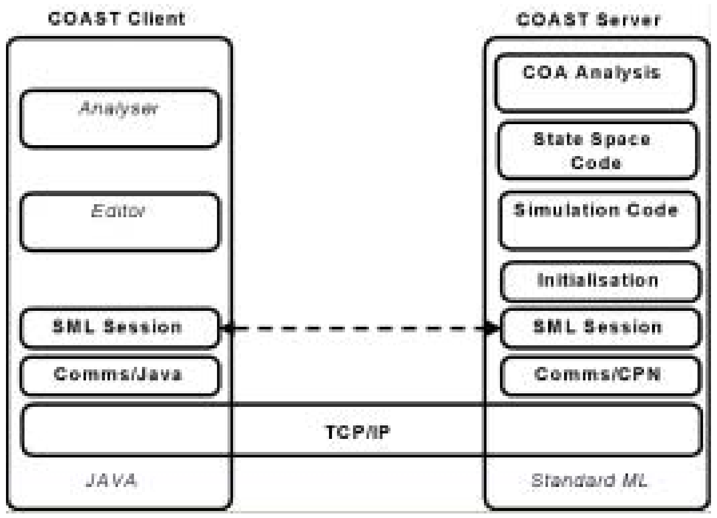


Figure 7: COAST Client/Server Architecture

The COAST client consists of two main parts: an *Editor* for creating and editing plans, and an *Analyser* for the analysis of plans. The COAST server consists of four main parts. The *Initialisation* module allows the CPN model to be initialised according to the plan to be analysed. The *Simulation Code* module is for executing the

CPN model and consists of the simulation code generated by the CPN computer tools for executing CPN models. Similarly, the *State Space Code* module consists of the code for generation of state spaces in the CPN computer tools. The *State Space Code* and *COA Analysis* modules support the generation of state spaces and LOPs.

4.2 Planning with COAST

To illustrate the use of COAST in planning, we consider the simple planning problem introduced in Section 2. Given the desired end state "*Amphibious forces successfully landed*", the planners develop a task "*Conduct amphibious assault*" to achieve the desired end state. Let T1 denote the "*Conduct amphibious assault*" task. As there are five preconditions that must be present for T1 to be prosecuted, five more tasks are developed to achieve the effects required by T1 as preconditions (See Table 1):
T2: Conduct air patrol;
T3: Conduct ASW operations in AO;
T4: Conduct airborne operations;
T5: Conduct maritime escort operation; and
T6: Conduct mine clearance operation.

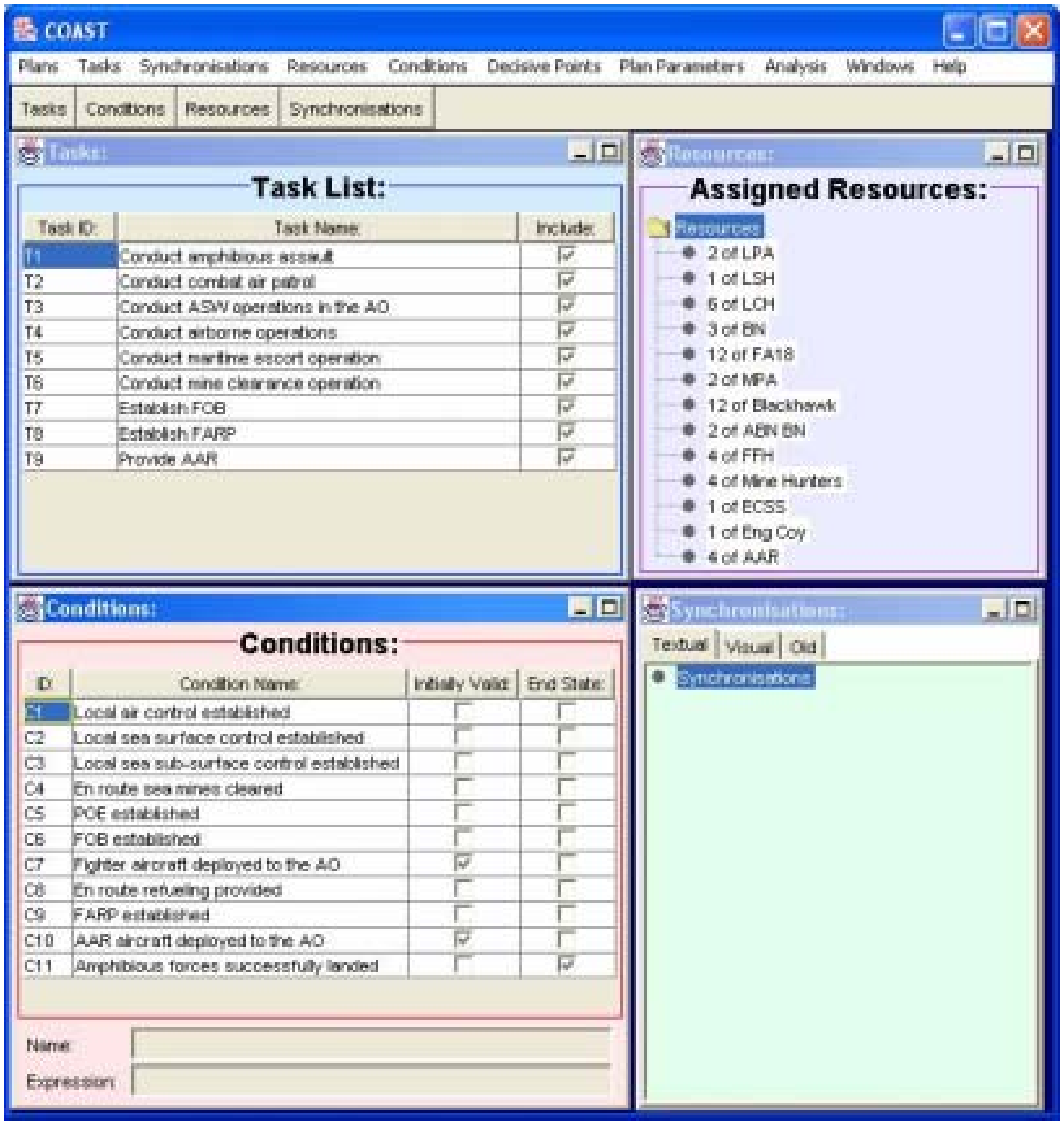
These five tasks will in turn require more effects produced by other tasks as preconditions. Consequently more tasks are developed. Figure 8 shows the GUI for the planning problem in terms of a task list, conditions, assigned resources, and synchronisations.

Tasks in the *task list* window are devised as previously described. It is important to note that for a given effect more than one task can be developed as alternatives of achieving the desired effect. Tasks may also produce undesired effects that can be shown in the GUI. For simplicity, these aspects of planning are not discussed in the example.

The end state, effects that tasks produce, and preconditions that tasks require, are all listed in the *conditions* window. There are three things to note regarding conditions. First, the end state is defined as a set of conditions that must be met for a campaign (or an operation) to terminate. Although only one condition is considered as the end state in the example, in more general situations more than one condition can be selected to constitute a desired end state. Second, in most cases except for the end state, conditions are generated as tasks are being developed. Third, some conditions are considered initially valid, and they constitute the initial state of the planning problem. In the example, we consider "*Fighter aircraft deployed to the AO*" and "*AAR aircraft deployed to the AO*" initially valid.

The *Assigned Resources* window in Figure 8 shows the forces and assets that have been assigned to the campaign/operation under planning. The resources can be added, deleted, and given periods of availability. In this example, the resources are assumed available for the entire operation.

The precedence and temporal relations between tasks are edited and shown in the *synchronisations* window. Fourteen synchronisation patterns are provided to represent such relations as "Task A occurs *d* time units after Task B". Details of synchronisations are not discussed in this paper for brevity. In an ideal planning situation, all necessary precedence and temporal relations could be represented through the task conditions relationship. Hence the example considered does not have to use synchronisations.



Task List:

Task ID:	Task Name:	Include:
T1	Conduct amphibious assault	✓
T2	Conduct combat air patrol	✓
T3	Conduct ASW operations in the AO	✓
T4	Conduct airborne operations	✓
T5	Conduct maritime escort operation	✓
T6	Conduct mine clearance operation	✓
T7	Establish FOB	✓
T8	Establish FARP	✓
T9	Provide AAR	✓

Assigned Resources:

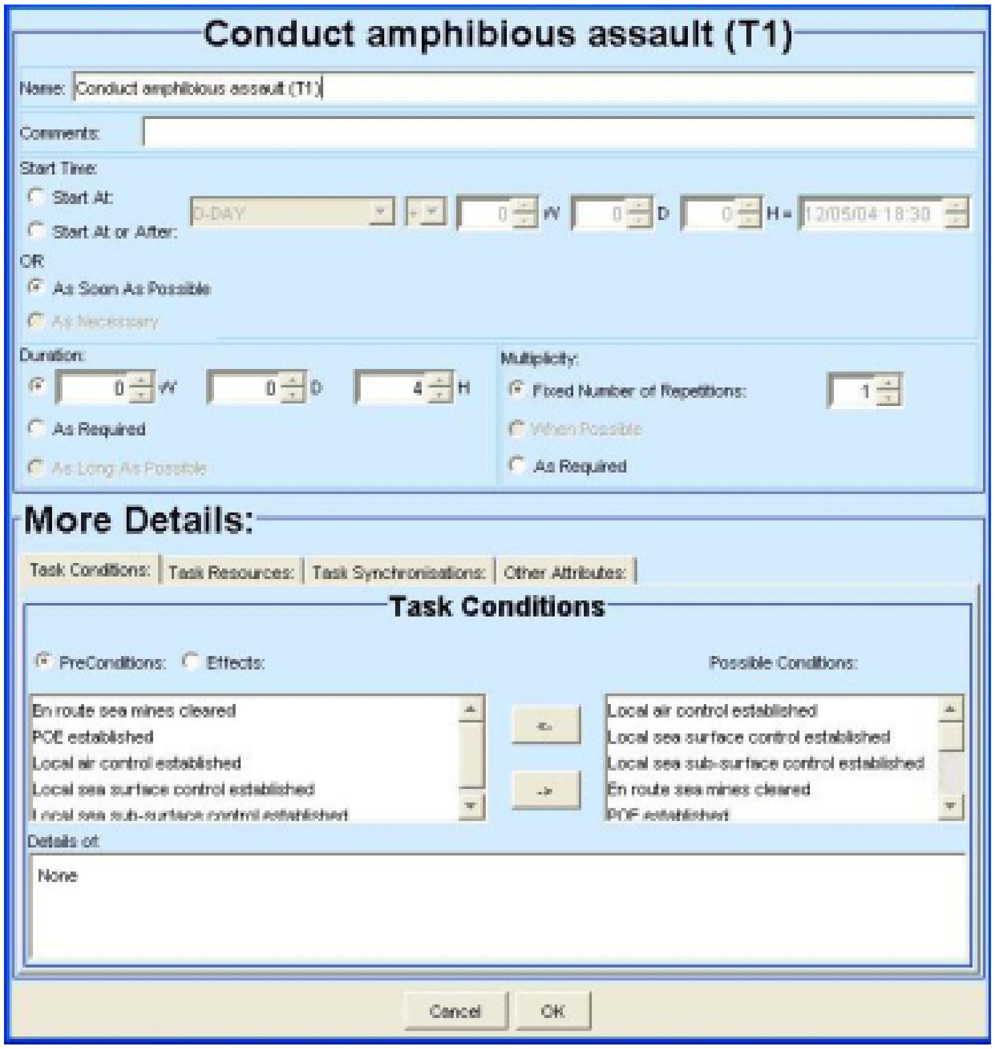
- Resources
  - 2 of LPA
  - 1 of LSH
  - 6 of LCH
  - 3 of BN
  - 12 of FA18
  - 2 of MPA
  - 12 of Blackhawk
  - 2 of ABN BN
  - 4 of FFH
  - 4 of Mine Hunters
  - 1 of ECSS
  - 1 of Eng Coy
  - 4 of AAR

Conditions:

ID:	Condition Name:	Initially Valid:	End State:
C1	Local air control established		
C2	Local sea surface control established		
C3	Local sea sub-surface control established		
C4	En route sea mines cleared		
C5	POE established		
C6	FOB established		
C7	Fighter aircraft deployed to the AO	✓	
C8	En route refueling provided		
C9	FARP established		
C10	AAR aircraft deployed to the AO	✓	
C11	Amphibious forces successfully landed		✓

Figure 8: Main GUI window of COAST

Figure 9 shows the COAST task editor. Information that can be entered through the task editor includes Task Name, Comments, Start Time, Duration, Conditions (both preconditions and effects), Resources (required and potentially lost resources due to attrition, for example), synchronisations, and others that are for record-keeping purposes. Tabs are used to organise the input information, and temporal aspects of a condition (e.g. normal precondition, sustained effect) are edited in the "Details of" area when the condition is selected. It is important to note in the task editor that COAST can sequence and schedule tasks with information such as "As Soon As Possible" and "As Required" duration in the context of a planning problem.



Conduct amphibious assault (T1)

Name: Conduct amphibious assault (T1)

Comments:

Start Time:

Start At: / Start At or After: D-DAY + 0 W 0 D 0 H = 12/05/04 18:30

OR

[✓ As Soon As Possible]

As Necessary

Duration: [✓ 0 W 0 D 4 H] / As Required / As Long As Possible

Multiplicity: [✓ Fixed Number of Repetitions: 1] / When Possible / As Required

More Details:

Task Conditions: | Task Resources: | Task Synchronisations: | Other Attributes:

Task Conditions

[✓ PreConditions:] Effects:

	Possible Conditions:
En route sea mines cleared	Local air control established
POE established	Local sea surface control established
Local air control established	Local sea sub-surface control established
Local sea surface control established	En route sea mines cleared
Local sea sub-surface control established	POE established

Details of:

None

Figure 9: Task Editor

Figure 10 shows the causal relations between the tasks by means of preconditions and effects. An arrow pointing from one task (e.g., T9) to another task (e.g., T2) indicates that the first task (T9) produces effect(s) that are needed as precondition(s) by the second task (T2). A black solid arrow indicates that the duration of the first task (T9) is "as required" (asreq) because the effect(s) that it produces is of the type of "Sustained Effects" (SE) and they are needed by the second task as the "Sustaining Preconditions" (SP). The red dotted arrows indicate causal relations between tasks achieved through precondition/effects pairs other than the pair of SP/SE. It is worth noting the complexity of the causal relations between tasks even with this simple example of nine tasks. The ability of deducing the fixed duration of a task from the "as required" specification through the construct of the SP/SE pair in a dynamic planning environment is significant. It means that much of the inconsistency in planning due to distributed staff developing tasks can be avoided.

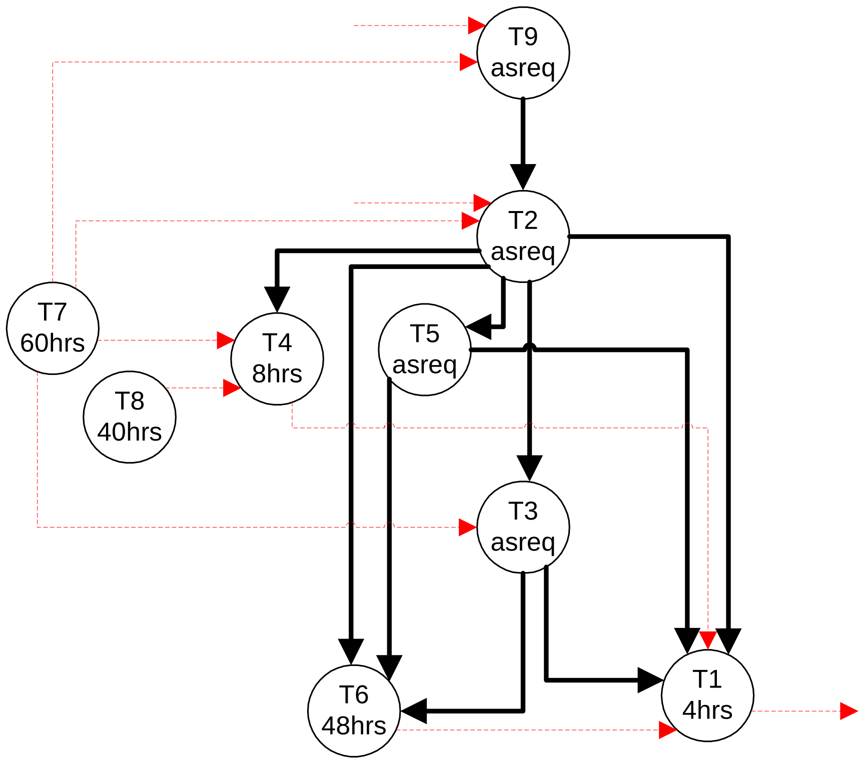
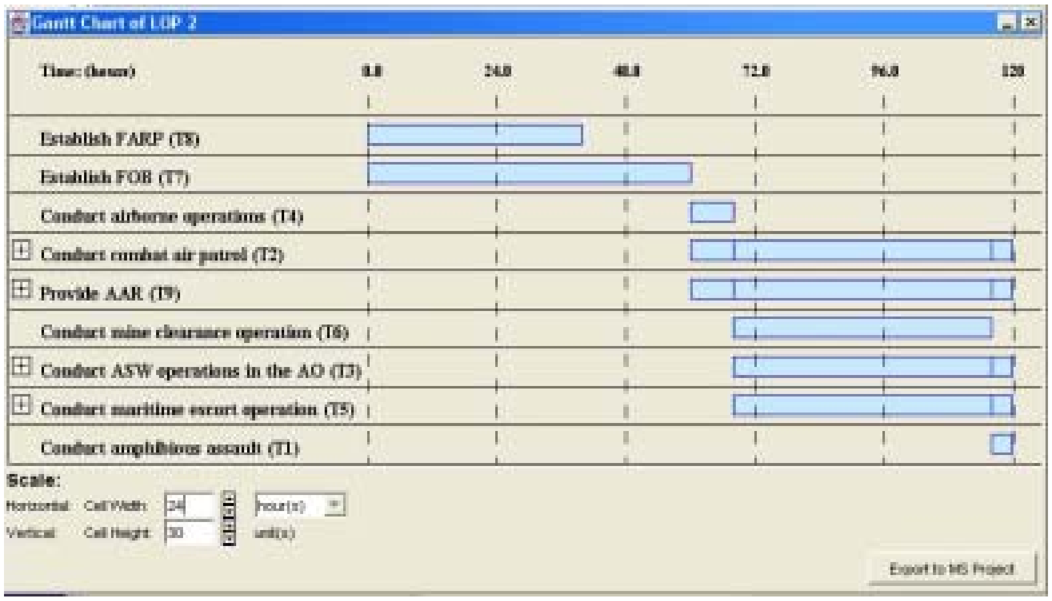


Figure 10: Causal Relationships between tasks

With the planning example, COAST generates two suitable and feasible LOPs. Figure 11 depicts the second LOP (LOP2) in a GANTT chart. These LOPs lead to the achievement of the desired end state through the precondition/effect relations, and pre-empt any potential conflicts of resource requirements by tasks.

By examining the LOP in Figure 11 against the task causal relations in Figure 10 we can see that the LOP in the GANTT chart obeys the causal relationships. Task 7 (Establish FOB) and Task 8 (Establish FARP) can occur immediately as no preconditions are required and there are sufficient resources for both tasks to execute. These tasks produce conditions that are required by Tasks 2, 3, 4, and 9. Sixty hours into the operation after both Task 7 and Task 8 have executed, and Task 4 (Conduct airborne operations) and Task 6 (Conduct mine clearance operation) can occur which both require the sustaining precondition of “Local air control established” which is a sustained effect produced by “Conduct combat air patrol” (Task 2). Due to resource requirements (12 FA18’s) of Task 2 however, it is only possible for Task 2 to provide support for one task. Therefore, a sequential execution is necessary where either Task 4 executes then Task 6 (LOP2), or Task 6 executes then Task 4 (LOP1). If more resources were available, Tasks 4 and 6 could occur concurrently, which would become the third LOP. Note that the “Conduct air patrol” task (T2) and the “Provide AAR” task (T9) are completely synchronised in Figure 11 due to the causal relationship between them through the condition of “En Route refuelling provided” being the sustained effect (SE) of T9 and a sustaining precondition (SP) of T2. The completion of Tasks Conduct airborne operations (T4) and Conduct mine clearance operation (T6), and the execution of Tasks Conduct combat air patrol (T2), Conduct ASW operations in the AO (T3), Conduct maritime escort operation (T5), and Provide

AAR (T9) produce conditions needed to execute the Conduct Amphibious Assault (T1) task. T1 produces the end state “Amphibious forces successfully landed”, which is obtained 120 hours into the operation.


Gantt Chart of LOP 2
Time: (hours)
0.0
24.0
48.0
72.0
96.0
120
Establish FARP (T8)
Establish FOB (T7)
Conduct airborne operations (T4)
Conduct combat air patrol (T2)
Provide AAR (T9)
Conduct mine clearance operation (T6)
Conduct ASW operations in the AO (T3)
Conduct maritime escort operation (T5)
Conduct amphibious assault (T1)
Export to MS Project

Scale:
Horizontal: Cell Width: 24 hour(s)
Vertical: Cell Height: 30 unit(s)

Figure 11: GANTT chart of LOP2

5. Conclusions and Discussions

In this paper, we presented COAST as an operational planning tool for COA development and analysis. Through the development of COAST, a conceptual representation of the military planning domain was developed and validated. The conceptual representation was then formalised with the CPN modelling tool, and state space analysis techniques applied to generate suitable and feasible lines of operation. The COAST software was developed with a Client/Server architecture that allows flexible and component based development. The client comprises the GUI, consistency checking and static causal analysis of tasks. The server consists of the formal CPN model, simulation code and analysis algorithms. The GUI makes the advanced modelling, planning and analysis techniques transparent to the military planning staff.

The current COAST capability enables logical and feasibility analysis of military COA for the development of suitable and feasible task sequences and schedules. In essence, it addresses the issue of existence of suitable and feasible COAs that achieve a commander’s mission in terms of the desired end state. Future research and development of COAST will focus on the quantitative aspects of COA analysis including optimisation.

Acknowledgements

The authors wish to acknowledge the insights from subject matter experts in the Australian Defence Force (ADF), particularly the staff from the ADF Warfare Centre and the Deployable Joint Force Headquarters.

References

[CPN Group, 1996] CPN Group (1996): *Design/CPN Online*. www.daimi.au.dk/designCPN

[Falzon et al, 2001] Falzon, L., Zhang, L. and Davies, M. (2001): Hierarchical Probabilistic Models for Operational-Level Course of Action Development. In Proceedings of the *6th Command and Control Research and Technology Symposium*, Annapolis, MD, VA, June 2001.

[Gallasch and Kristensen, 2001] Gallasch, G. and Kristensen, L.M. (2001): Comms/CPN: A Communication Infrastructure for External Communication with Design/CPN. In Proceedings of the *3rd Workshop and Tutorial on CPNs and CPN Tools*, pp. 79-93. DAIMI PB-554, Department of Computer Science, University of Aarhus.

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

[Kristensen et al, 2002] Kristensen, L.M., Mitchell, B., Zhang, L. and Billington, J. (2002): Modelling and Initial Analysis of Operational Planning Processes using Coloured Petri Nets. In Proceedings of the *Workshop on Formal Methods Applied to Defence Systems*. Adelaide, June 2002.

[Mortensen, 2000] Mortensen, K.H. (2000): Automatic Code Generation Method based on Coloured Petri Net Models Applied to an Access Control System. In Proceedings of the *International Conference on Application and Theory of Petri Nets*. Vol. 1825 of Lecture Notes in Computer Science, pp. 367-386. Springer-Verlag.

[Mulvehill and Caroli, 2000] Mulvehill, A. M., and Caroli, J. A. 2000. JADE: A tool for rapid crisis action planning. In Proceedings of the *5th International Command and Control Research and Technology Symposium*.

[Priest, 2002] Priest, J., Smallwood, R., Falzon, L., Zhang, L., Lumsden, S. (2002): A Centre of Gravity Analysis Tool to Support Operational Planning. In Proceedings of the *7th International Command and Control Research and Technology Symposium*, Quebec City, QC, Canada, September 2002.

[Wagenhals, 1998] Wagenhals, L., Shin, I. and Levis, A.H. (1998): Creating Executable Models of Influence Nets with Coloured Petri Nets. In *International Journal on Software Tools for Technology Transfer*, Volume 2, Issue 2, pp. 168-181. Springer-Verlag.

[Zhang et al, 2001] Zhang, L., Mitchell, B., Falzon, L., Davies, M., Kristensen, L.M. and Billington, J. (2001): Model-based Operational Planning Using Coloured Petri Nets. In Proceedings of the *6th International Command and Control Research and Technology Symposium*, Annapolis, MD, VA, June 2001.

[Zhang et al, 2002] Zhang, L., Kristensen, L.M., Janczura, C., Gallasch, G., and Billington, J. (2002): A Coloured Petri Nets Based Tool for Course of Action Development and Analysis. In Proceedings of the *Workshop on Formal Methods Applied to Defence Systems*. Adelaide, June 2002.