Defence Research and Development Canada | Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

An Introduction to the IP/PCT Model Implementation in IPME

Keith Hendy

Defence R&D Canada
Technical Report
DRDC Toronto TR 2010-040
March 2011

Canada

An Introduction to the IP/PCT Model Implementation in IPME

Keith C. Hendy

Defence R&D Canada

Technical Report

DRDC Toronto TR 2010-040

March 2011

Author

Original signed by K. Hendy

Keith Hendy
Deputy Director General

Approved by

Original signed by D. Reding

Dale Reding
Director General, DRDC Toronto

Approved for release by

Original signed by J.V. Baranski

Dr. Joseph V. Baranski
Chair, Knowledge and Information Management Committee
Chief Scientist

Abstract

This document provides a description of the Information Processing (IP)/Perceptual Control Theory (PCT) model implemented in the Integrated Performance Modelling Environment (IPME) software by Micro Analysis and Design. The current document is an edited and reduced version of an earlier report. In the current document, an attempt has been made to improve readability through a reorganisation of the material and the elimination of content that is not central to understanding the function of the IP/PCT model within IPME. The essence of the IP model is that all factors that impact on human cognitive workload can be reduced to their effects on the amount of information to be processed and the amount of time available before the decision has to be actioned. From this position, it can be shown that if humans are limited at the rate at which they process information then operator workload, performance, and error production are all functions of the time pressure. The IP Model is about time and the information to be processed.

The PCT model argues that humans behave as multi-layered closed loop control systems. The set points for these control loops are our perceptual goals (or how we want to see, hear, feel, taste, or smell the state of the world). According to PCT, we sense the world state, forming a perception of that state which we then compare with our goal. If there is a difference between our perceived and desired states, we formulate an action. This action is implemented in order to operate on the world so as to drive the perceived state of the variables of interest towards the goal. The perceptual processes and the decisional processes draw on internal knowledge states that transform sensation to perception, and difference to action. Our attentional mechanism shifts our focus from loop to loop to loop. The PCT model is therefore about Goals, Attention, Knowledge and Feedback.

Résumé

Ce document fournit une description des modèles de traitement de l’information (TI) et de la théorie du contrôle perceptuel (TCP) mis en œuvre dans le logiciel Environnement intégré de modélisation des performances (EIMP) par Micro Analysis and Design. Ce document est une version réduite et modifiée d’un précédent rapport. Dans le présent document, nous avons tenté d’améliorer la lisibilité grâce à une réorganisation du matériel et à une élimination du contenu qui n’est pas essentiel à la compréhension de la fonction des modèles de TI et de la TCP au sein de l’EIMP. L’essence du modèle de TI est que tous les facteurs qui touchent la charge de travail cognitive de l’être humain peuvent être réduits à leur effet sur la quantité d’information à traiter et le temps requis pour prendre la décision. De ce point de vue, il est démontré que si les êtres humains sont restreints à la vitesse à la quelle ils traitent l’information, alors la charge de travail, la performance et la production d’erreurs de l’opérateur sont toutes fonction des contraintes de temps. Le modèle TI porte sur le temps et l’information à traiter.

Selon le modèle de la TCP, les êtres humains se comportent comme des systèmes de commande en boucle fermée (ou asservi) à de multiples couches. Les valeurs de réglage de ces boucles de contrôle sont nos objectifs perceptuels (ou comment nous voulons voir,

entendre, toucher, goûter ou sentir l'état du monde). Selon la TCP, nous sentons l'état du monde, formons une perception de cet état, et nous le comparons ensuite à nos objectifs. En cas de différence entre l'état perçu et l'état désiré, nous formulons une action. La mise en œuvre de cette action vise à agir sur la réalité en vue de faire évoluer l'état perçu des variables d'intérêt vers l'objectif visé. Les processus perceptuels et les processus décisionnels se fondent sur le savoir en mémoire pour transformer une sensation en perception, et un écart en action. Nos mécanismes attentionnels dirigent notre attention de boucle en boucle. Le modèle de la TCP est donc centré sur les objectifs, l'attention, le savoir et la rétroaction.

Executive summary

Hendy, K.C. 2010. An introduction to the IP/PCT Model implementation in IPME. DRDC Toronto TR 2010-040. Defence R&D Canada — Toronto.

The Integrated Performance Modelling Environment (IPME) is a software package for predicting human performance, operator workload and error production in complex human-machine systems. Using data from some form of task or activity analysis, task networks can be built from which human and system performance can be predicted. The addition of models of human information processing capability reduce the burden of the analyst in making the task networks sensitive to human capabilities and limitations. IPME is a joint development between *QinetiQ* of the United Kingdom and *Defence R&D Canada*. IPME is available from Micro Analysis and Design of Boulder Colorado.

This document provides a description of the Information Processing (IP)/Perceptual Control Theory (PCT) model implemented in the IPME. The current document is an edited and reduced version of an earlier report that formed the basis of the software specification for the IP/PCT implementation. In the current document, an attempt has been made to improve readability through a reorganisation of the material and the elimination of content that is not central to understanding the function of the IP/PCT model within IPME. The essence of the IP model is that all factors that impact on human cognitive workload can be reduced to their effects on the amount of information to be processed and the amount of time available before the decision has to be actioned. From this position, it can be shown that if humans are limited at the rate at which they process information then operator workload, performance, and error production are all functions of the time pressure. The IP model is about time and the information to be processed.

The PCT model argues that humans behave as multi-layered closed loop control systems. The set points for these control loops are our perceptual goals (or how we want to see, hear, feel, taste, or smell the state of the world). According to PCT, we sense the world state, forming a perception of that state which we then compare with our goal. If there is a difference between our perceived and desired states, we formulate an action. This action is implemented in order to operate on the world so as to drive the perceived state of the variables of interest towards the goal. The perceptual processes and the decisional processes draw on internal knowledge states that transform sensation to perception, and difference to action. Our attentional mechanism shifts our focus from loop to loop to loop. The PCT model is therefore about Goals, Attention, Knowledge and Feedback.

Sommaire

Hendy, K.C. 2010. An introduction to the IP/PCT Model implementation in IPME. DRDC Toronto TR 2010-040. Defence R&D Canada — Toronto.

L'Environnement intégré de modélisation des performances (EIMP) est un progiciel qui permet de prédire la performance humaine, la charge de travail de l'opérateur et la production d'erreurs dans les systèmes complexes humain-machine. À l'aide de données à partir d'une sorte d'analyse de tâche ou d'activité, des réseaux de tâches peuvent être créés grâce auxquels ont peut prédire la performance d'un humain et d'un système. L'ajout de modèles de traitement de l'information humaine réduit le fardeau de l'analyste en rendant les réseaux de tâches sensibles aux capacités et aux contraintes humaines. L'EIMP est un système développé conjointement par *QinetiQ* du Royaume-Uni et *Recherche et développement pour la défense Canada*. Il est possible de se procurer l'EIMP auprès de Micro Analysis and Design de Boulder Colorado.

Ce document fournit une description des modèles de traitement de l'information (TI) et de la théorie du contrôle perceptuel (TCP) mis en œuvre dans le logiciel Environnement intégré de modélisation des performances (EIMP) par Micro Analysis and Design. Ce document est une version réduite et modifiée d'un précédent rapport. Dans le présent document, nous avons tenté d'améliorer la lisibilité grâce à une réorganisation du matériel et à une élimination du contenu qui n'est pas essentiel à la compréhension de la fonction des modèles de TI et de la TCP au sein de l'EIMP. L'essence du modèle de TI est que tous les facteurs qui touchent la charge de travail cognitive de l'être humain peuvent être réduits à leur effet sur la quantité d'information à traiter et le temps requis pour prendre la décision. De ce point de vue, il est démontré que si les êtres humains sont restreints à la vitesse à la quelle ils traitent l'information, alors la charge de travail, la performance et la production d'erreurs de l'opérateur sont toutes fonction des contraintes de temps. Le modèle TI porte sur le temps et l'information à traiter.

Selon le modèle de la TCP, les êtres humains se comportent comme des systèmes de commande en boucle fermée (ou asservi) à de multiples couches. Les valeurs de réglage de ces boucles de contrôle sont nos objectifs perceptuels (ou comment nous voulons voir, entendre, toucher, goûter ou sentir l'état du monde). Selon la TCP, nous sentons l'état du monde, formons une perception de cet état, et nous le comparons ensuite à nos objectifs. En cas de différence entre l'état perçu et l'état désiré, nous formulons une action. La mise en œuvre de cette action vise à agir sur la réalité en vue de faire évoluer l'état perçu des variables d'intérêt vers l'objectif visé. Les processus perceptuels et les processus décisionnels se fondent sur le savoir en mémoire pour transformer une sensation en perception, et un écart en action. Nos mécanismes attentionnels dirigent notre attention de boucle en boucle. Le modèle de la TCP est donc centré sur les objectifs, l'attention, le savoir et la rétroaction.

Table of contents

List of figures

List of tables

This page intentionally left blank.

Introduction

This document provides a description of the Information Processing (IP)/Perceptual Control Theory (PCT) model originally implemented in the Integrated Performance Modelling Environment (IPME; http://www.maad.com/index.pl/ipme). The current document is an edited and reduced version of an earlier report (Hendy, 1994a) which formed the specification for the software development. In the current document, an attempt has been made to improve readability through a reorganisation of the material and the elimination of content that is not central to understanding the function of the IP/PCT model within IPME. It does of course represent one of many possible instantiations of the basic construct and recognizes that IP/PCT is just one of many theoretical foundations for building a human performance modelling environment. It is not the intent of this report to further argue this position, but rather document how these ideas were used to construct an algorithmic representation of IP/PCT.

Although considerable time has elapsed since the IP/PCT model was originally described and implemented in IPME, this document is a record of the assumptions that drove the software implementation. As such the publication of this document will aid in the interpretation of the IP/PCT model implementation in IPME, particularly for those who may wish to improve or otherwise modify the algorithm.

The IP model is described in detail elsewhere (Hendy, East, and Farrell, 2001; Hendy, Liao, and Milgram, 1997). The essence of the IP model is that all factors that impact on human cognitive workload can be reduced to their effects on the amount of information to be processed and the amount of time available before the decision has to be actioned. From this position, it can be shown that if humans are limited at the rate at which they process information then operator workload, performance, and error production are all functions of the time pressure. Time Pressure is defined as follows:

$$Time\ Pressure = \frac{\text{Time to process information}}{\text{Time available}},$$

which, at a constant rate of processing, reduces to

$$Time\ Pressure \propto \frac{Amount\ of\ information\ to\ be\ processed}{Time\ available}.$$

The IP model is about time and the information to be processed (knowledge). The IP model applies everywhere in the human cognitive system where information is being processed.

A fundamental assumption of the IP model is that information is processed serially within a given structure. Interference in multiple concurrent task performance is assumed to depend on the amount of physical overlap between the structures involved in processing each task. It is expected that task interference, assuming a constant strategy, will manifest itself as an increase in the processing or decision time for one or more of the tasks (Hawkins, Olbrich-Rodriguez, Halloran*, et al*., 1979). This is a direct consequence of the competition for serial resources. Note that the concept of *overlap* is assumed to exist at the neural level. This view is consistent with the architecture described by Detweiler and Schneider (1991) for a connectionist model of skill acquisition.

In the IP model, the selection of a particular information processing strategy is assumed to involve a specific set of processing structures. Different strategies will, in general, involve different structures. The selected strategy also characterises the *depth of processing* and sets the total amount of information to be processed, and hence the time to arrive at a decision. Time and the amount of information to be processed are always related linearly by the fixed processing rate.

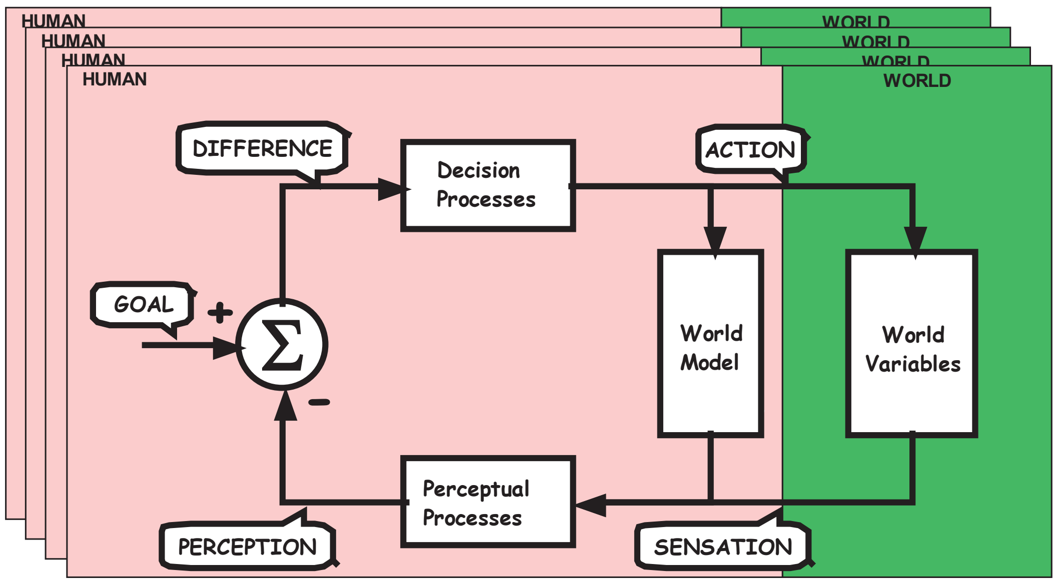

Figure 1. *The multi-layered Perceptual Control loop for a human operator interacting with the world.*

The PCT model (Powers, 1973) argues that humans behave as multi-layered closed loop control systems (see Figure 1). The set points for these control loops are our perceptual goals (or how we want to see, hear, feel, taste, or smell the state of the world). According to PCT, we sense the world state, forming a perception of that state which we then compare with our goal (as shown by the ∑ sign in Figure 1 which represents the mathematical summing operation). If there is a difference between our perceived and desired states, we formulate an action. This action is implemented in order to operate on the world so as to drive the perceived state of the variables of interest towards the goal. The perceptual processes and the decisional processes draw on internal knowledge states that transform sensation to perception, and difference to action. Our attentional mechanism shifts our focus from loop to loop to loop. The PCT model is therefore about Goals, Attention, Knowledge and Feedback.

The IP model acts wherever there are data transformation or information processing actions. These occur in the perceptual processes, the decisional processes and in the internal world model processes. Combining the IP and PCT models one can say that human decision-making depends on the management of time, knowledge and attentional resources (Hendy and Lichacz, 1999). The IP and PCT models are complementary. The dynamic behaviour of the PCT model is bandwidth limited with this limitation coming from the lags and delays in the terms of the transfer functions (Decision and Perceptual Processes). The IP model provides a mechanism for explaining these delays and shows how strategy selection provides a trade-off between speed of response and absolute accuracy of performance. By going to a less accurate, less computationally intensive strategy, transport delays will be less and the dynamic response will increase due to the increased bandwidth. Transport delays are a product of the time required to process the information (Bits) associated with selecting and forming an action at a finite processing rate (Bits per second).

This page intentionally left blank.

Multiple Task Performance

In the IP/PCT model, elements of multiple concurrent tasks that draw on the same processing structures are assumed to be processed serially by time-multiplexing. The first assumption to be made, and perhaps the most fundamental, is that operators will service no more than 2 tasks concurrently for which the degree of interference is non-zero (Hendy, 1994b). While the literature on dual task performance is abundant (e.g., Wickens, 1992, p. 364ff.), information on multiple (more than 2) task performance is less prodigious. While the restriction to dual tasking probably provides a conservative prediction, overt multiple task performances appear to be rare in operational systems (Shaffer, Hendy, and White, 1988). The restriction to dual tasking will be limited to tasks that require higher level processing, say at the level of Rasmussen's rule-based and knowledge-based activities (Rasmussen, 1983). There is no limit set on the number of purely skill-based activities (these are designated Category 1 in Table 3, see p. 14) that can be performed in concert, provided there are no structural interference limitations (see discussion on p. 11).

Time-multiplexing

Suppose that the performance of tasks *i* and *j* overlap in the time domain. Then it is assumed that processing two tasks that share a common structure will occur by rapidly time-multiplexing within that structure as illustrated in Figure 2 (zero switching time is assumed). In Figure 2, Tasks 1 and 2 are shown to be processed on successive processing intervals (assuming equal priority is given to both tasks). The reaction times of both tasks will be delayed by this form of processing. Note that:

T_i is the task completion time of the *i*th task when performed in isolation, and

T_{ij} is the task completion time of the *i*th task when performed in combination with the *j*th task.

Suppose that, instead of successively switching from one task to the other, there is a probability associated with the allocation of a processing structure to each task within a given interval (Kinchla, 1980). Assume that in any processing interval, the probability that the processing structure is assigned to the *i*th task is p_i. Then on average, over any given time period, a proportion p_i of the processing time is devoted to task *i*, while a proportion $p_j = (1-p_i)$ is devoted to task *j*.

In general, suppose tasks *i* and *j* do not require the same processing structures for all of their processing time, but share a common structure for a proportion (as defined by the coefficient $\mathbf{c}_{ij}$) of the period of their overlap in the time domain. Two cases need to be considered:

> CASE 1 — in which the processing of one of the tasks (assume it to be task *j*) is entirely embedded within the processing time of the other (task *i*); and
>
> CASE 2 — in which the tasks partially overlap (the processing time of task *i*, remaining after task *j* starts, is entirely embedded within the processing time of task *j*).

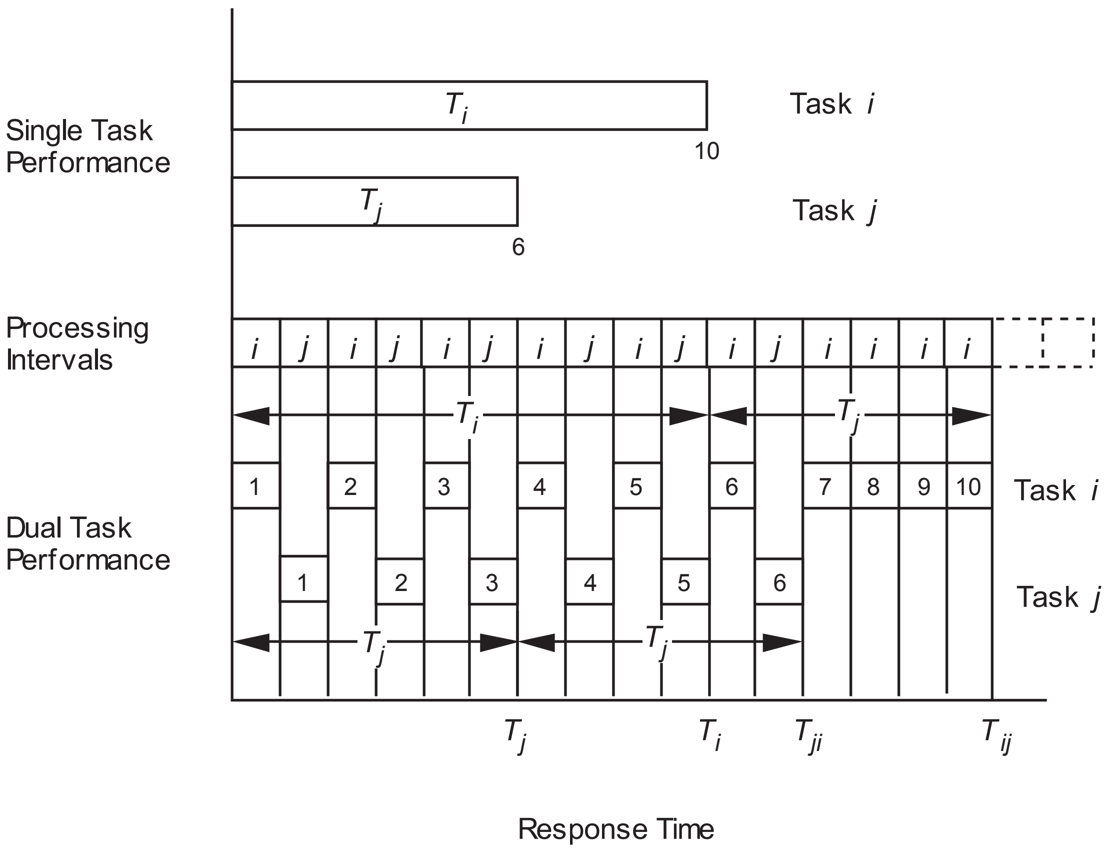

Figure 2. **An example of time-multiplexing in 'concurrent' task processing (CASE 1 - two tasks completely overlapping)**

Let (for a more detailed discussion of the following derivation see Appendix 1):

$t_i(\mathrm{s})$ be the starting time of the *i*th task,

$t_i(\mathrm{e})$ be the ending time of task *i*, when performed alone,

$t_{ij}(\mathrm{e})$ be the ending time of the *i*th task when performed in combination with the *j*th task.

$\mathbf{c}_{ij}$ be the proportion of time that the overlapping tasks share a common processing structure (for simplification, this time is assumed to be evenly distributed throughout the overlap), and

p_i be the probability that in any given time interval, processing resources will be devoted to task *i*, rather than task *j* (note that $p_i = (1 - p_j)$).

Then for CASE 1, $t_j(\mathrm{s}) \geq t_i(\mathrm{s})$ and $t_{ij}(\mathrm{e}) \geq t_{ji}(\mathrm{e})$, and assuming that the requirement to share common structures is distributed evenly throughout the period of overlap, it can be shown that

$$T_{ij} = T_i + \frac{\mathbf{c}_{ij}(1 - p_i)}{1 - p_i \mathbf{c}_{ij}} T_j, \tag{1a}$$

$$T_{ji} = \frac{T_j}{1 - p_i \mathbf{c}_{ij}}, \text{ and} \tag{1b}$$

$$T_i - T_j\left[\frac{1-\mathbf{c}_{ij}(1-p_i)}{1-p_i\mathbf{c}_{ij}}\right] - \{t_j(\mathrm{s}) - t_i(\mathrm{s})\} \geq 0. \tag{1c}$$

Similarly, for CASE 2, $t_j(\mathrm{s}) \geq t_i(\mathrm{s})$ and $t_{ij}(\mathrm{e}) < t_{ji}(\mathrm{e})$, and

$$T_{ij} = \frac{[T_i - p_j\mathbf{c}_{ij}\{t_j(\mathrm{s}) - t_i(\mathrm{s})\}]}{1-p_j\mathbf{c}_{ij}}, \tag{2a}$$

$$T_{ji} = \frac{(1-p_j\mathbf{c}_{ij})T_j + \mathbf{c}_{ij}(1-p_j)T_i - \mathbf{c}_{ij}(1-p_j)\{t_j(\mathrm{s}) - t_i(\mathrm{s})\}}{1-p_j\mathbf{c}_{ij}}, \text{ and} \tag{2b}$$

$$T_i - T_j\left[\frac{1-\mathbf{c}_{ij}(1-p_i)}{1-p_i\mathbf{c}_{ij}}\right] - \{t_j(\mathrm{s}) - t_i(\mathrm{s})\} < 0. \tag{2c}$$

In these two sets of expressions, the inequality classifies the situation according to case.

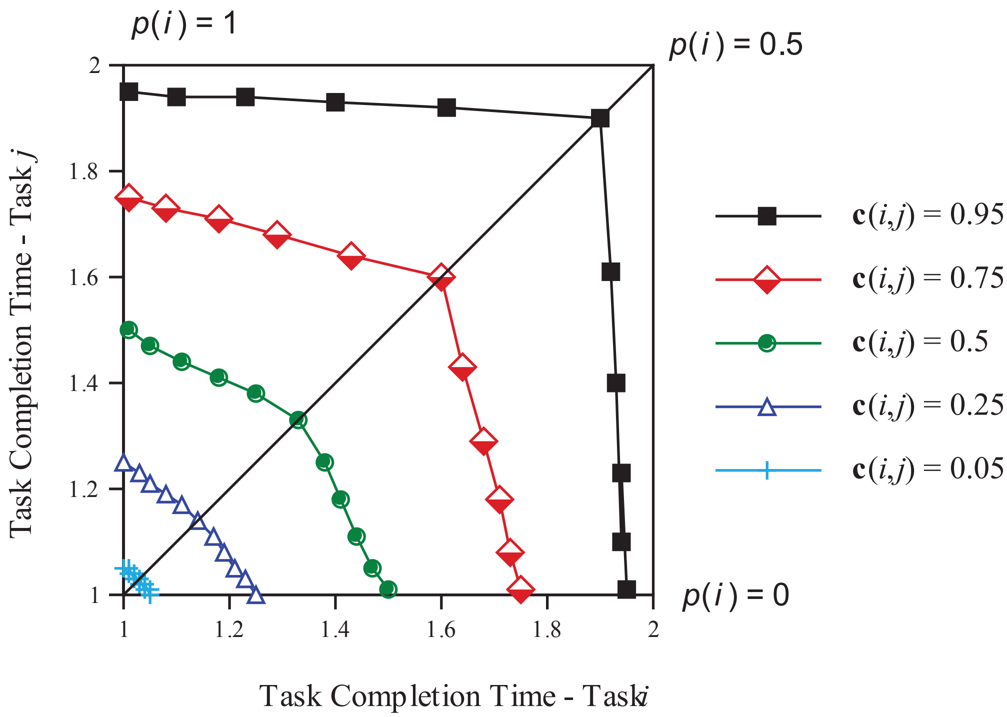

***Figure 3.* Performance Operating Characteristics for the IP/PCT model, for various values of task interference *(c_{ij})* and the probability of attending *(p(i))* to the processing of task *i*.**

The relationships for task completion times under the two cases, discussed above, produce Performance Operating Characteristics (POCs) for the IP/PCT model, as shown in Figure 3. The following assumptions were made in generating these curves;

$$T_i = T_j = 1\,\text{sec, and}$$

$$\mathrm{t_i}(\mathrm{s}) = t_j(\mathrm{s}).$$

The five POCs of Figure 3 are for values of $\mathbf{c}_{ij} \in \{0.05, 0.25, 0.5, 0.75, 0.95\}$ and probabilities of attending in the range 0.01 to 0.99. These POCs show the behaviour of the IP/PCT model as one prioritises the processing of one task over the other. Note that as $p_i \to 1$ or $p_i \to 0$, the IP/PCT model asymptotes to values which are consistent with the prediction of serial processing. Overall the predictions of the IP/PCT model are reasonable, as one would expect the response latency of the lower priority task $(p \to 0)$ to approach $(T_i + T_j)$. Note further that if $c_{ij} = 1$ and $p_i = 0$ or 1, discontinuities exist. At these boundary conditions, behaviour reverts to serial rather than multiplexed processing. On Page 14 it is seen that IPME is forced to serial processing when c_{ij} exceeds a critical value (nominally 0.7). In IPME, for computational reasons, the lowest value p_i can take is approximately 0.001 (see page 31).

If the proportion of the time devoted to processing each task is determined by task priorities P_i and P_j, then

$$p_i = \frac{P_i}{P_i + P_j}. \tag{3}$$

The task priority values (P_i and P_j) are determined by the instantaneous time pressures for the tasks (defined on p. 27).

To allow for task resumption after an interruption it is necessary to keep a running total of the amount of actual processing time devoted to each task and conversely the amount of processing time remaining for each task. At any time $t_j(\mathrm{s}) \le t \le t_{ji}(\mathrm{e})$, the amount of processing time devoted to resumable task *i*, since task *j* commenced, is

$$\Delta T_{ij} = p_i \mathbf{c}_{ij} \{t - t_j(\mathrm{s})\} + (1 - \mathbf{c}_{ij})\{t - t_j(\mathrm{s})\}, \tag{4}$$

and to task *j*:

$$\Delta T_{ji} = (1 - p_i)\mathbf{c}_{ij} \{t - t_j(\mathrm{s})\} + (1 - \mathbf{c}_{ij})\{t - t_j(\mathrm{s})\}. \tag{5}$$

Davis (1971) suggests that the time remaining on an interrupted task should be increased by some factor on resumption, and that the priority of the task should also be increased to avoid constant interruption. This might be considered to constitute a start up penalty. In Davis' model, this factor is chosen from one of four values depending on the percentage of the original task completion time remaining. For the implementation of the IP/PCT model, the penalty factor is a global variable, with the possibility of modification at the task level. It takes the default value of $\mu = 1.05$ (assigned arbitrarily).

Conceptually this penalty factor represents additional information processing, due to task resumption, which must be discharged before the information remaining on the original task can be processed. The calculation of the running total will take this into account. Therefore, each time a task is interrupted two calculations are made.

1 The **actual processing time remaining** on the task $\Delta T_{act}(n)$ at the *n*th interruption. This is equal to the actual task completion time remaining when the task last started or resumed $\Delta T_{act}(n-1)$, minus the amount of processing time logged against the primary task in the current iteration, that is

$$\text{if } \left(\Delta T_{ij}(n) - (\mu - 1)\Delta T_{act}(n-1)\right) > 0, \text{ then}$$

$$\Delta T_{act}(n) = \left(\Delta T_{act}(n-1) - \left(\Delta T_{ij}(n) - \mu \Delta T_{act}(n-1)\right)\right) \quad (6a)$$

else

$$\Delta T_{act}(n) = \Delta T_{act}(n-1)\,. \quad (6b)$$

Note that when $n = 1$,

$$\Delta T_{act}(1) = T_i - \Delta T_{ij}(1)\,. \quad (6c)$$

2 The **effective time remaining on the task** which includes the penalty due to resumption, is

$$\Delta T_{eff}(n) = \mu \Delta T_{act}(n) \quad (7)$$

The best analogy for this process is that of paying back a loan. Suppose you borrow $10000 and agree to pay back $1000 per week (equivalent to single task performance) which is the most you can afford. This will require 10 payments to service the debt. This works fine for the first couple of weeks until the tax man hits you with a bill of $5000. Now you will have to service the two loans concurrently (equivalent to two interfering tasks). You find that $1000 per week is still all you can afford (this is equivalent to a constant processing rate) so you make a deal with both parties to pay back the $1000 from the original loan every second week (still a total of 10 payments required) and $1000 to the tax man (5 payments) on alternate weeks (identical to time multiplexing the payments). Obviously, the date of the final payment will have moved to the right for both commitments compared with servicing the creditors separately.

Say you loose your job for four weeks and no payments are made to one of the creditors (equivalent to an interruption, delay etc.). Your creditor (the one you are defaulting to) is very obliging (no penalty or interest is applied) so the amount of the loan doesn't change, all amounts paid up to the job loss are credited against the loan and the amount remaining is simply the original loan less the amount previously paid. Obviously if a 5% default penalty is applied the actual amount of the loan would increase in dollar terms as well as the date of the final payment (equivalent to the start up penalty).

This page intentionally left blank.

Task Interference

For multiple concurrent task performance it is assumed that two types of interference can occur, namely, *structural* interference and *resource limited* interference. The term *structural* interference is used, quite specifically in this context, to describe interference effects that are due to limitations such as:

- the inability to focus foveally at different images concurrently, when they are widely separated in visual angle;
- those problems associated with operating spatially separated controls with the same hand or limb; and
- the inability to speak two messages at the same time.

Structural limitations, in this context, have little or nothing to do with the **processing structures** involved — at least at the higher levels of processing. They are driven by physical rather than information processing limitations and are therefore assumed to be largely associated with input and output stages, rather than cognition.

Structural interference is assumed to be *all or nothing*, that is $\mathbf{c}_{ij} = 1$ or 0. Matrices of interference coefficients for the *visual*, *auditory*, *cognitive*, and *manual*/*kinesthetic* domains are shown in Tables 1, 2, 3 and 4, respectively. Default values are shown for completeness. Note that these assignments have been made somewhat arbitrarily and no claims are made for their validity.

Visual domain

The visual domain consists of a single channel for information flow. Tasks are categorised according to whether they require foveal (central) vision or can be processed peripherally (see Table 1). A home area is defined for vision, which is assumed to be the resting position of the eyes in the absence of specific **operator initiated or goal directed** eye movements. At the completion of each task that involves a visual component, or when the task is interrupted or shed, the direction of gaze is returned to this point.

It is assumed that the highest priority visual task is foveated. If, and only if, this is a task that can be performed with peripheral vision, the direction of gaze shifts to the next highest priority central visual task that satisfies the criterion for selection into the active task list (see discussion on the allocation of attention module starting on p. 19). Hence, the direction of gaze is always either the home position or the currently selected, highest priority, visual task (with central tasks taking precedence over peripheral). To accommodate different visual environments (e.g., the use of night vision aids), one can edit the fields containing the visual subtense ranges in Table 1. Note that as some operators might be using night vision aids and others will not, these tables are set at the level of the operator. The default is that Table 1 is identical for all operators.

Table 1. Task interference coefficients (c_{ij}) for a human information processing model — visual domain.

Channel	*Mode*	*Interference*
Vision	Input	Structural ($\mathbf{c}_{ij}$ = 1 or 0)

OPERATOR INITIATED CATEGORIES	ANGULAR SUBTENSE† (DEGREES) BETWEEN TASKS			
	Ø≤ 2	2<Ø≤30	30<Ø≤90	90<Ø≤180
0. None (no visual component)	**0.0**	If a task has no visual component it will not interfere with **any other** task in the visual domain.		
1. Central-Central	**0.0**	**1.0**	**1.0**	**1.0**
2. Central-Peripheral	**0.0**	**0.0**	**1.0**	**1.0**
3. Peripheral-Peripheral	**0.0**	**0.0**	**0.0**	**1.0**
4. No allocation (default) with any visual task	**1.0**	As these tasks are neither allocated as central or peripheral, they will be assumed to interfere with **all** other visual tasks.		

† *These values are for situations involving **operator initiated** eye movements. Note that an externally initiated visual signal, occurring outside a certain angle (say 30°) of visual arc, will not be detected — or will be detected with a certain probability (see Table 7) — and therefore may not be 'serviced' even serially. Tasks not allocated are not serviced.*

For simplicity, each visual task is assigned to an area in the visual scene (*Area* 1, *Area* 2, etc.). A global lookup table sets the approximate (or exact) angular subtense between all pairwise combinations of these areas. These angles are used in the assessment of visual interference coefficients and probabilities of detection for externally cued stimuli. By default, *Home* is *Area* 1, although this is within the analyst's capability to change. The actual number of visual areas is set by the analyst.

Obviously, to obtain the degree of discrimination implied by Table 1, the visual scene would have to be divided up into increments of 2° of solid angle. A coarser analysis might simply assign 0° angular subtense to all tasks that share a common area. Note that the visual scene does not have to be divided evenly. A fine grid can be used in areas where there are multiple central tasks, while a coarse grid can be used elsewhere.

Auditory domain

The auditory domain consists of two channels, namely *audition* and *vocalisation*. Five types of auditory and vocal signals are considered. They are:

- tones or simple auditory cues such as buzzers, bells, chimes and horns,
- speech that is incidental to the current activities — monitored for presence and general content rather than for detailed meaning
- complex auditory signals and patterns (e.g., sonar signals, Morse code)

- attended speech (i.e., speech that is directly relevant to current activities), and
- voice output.

Table 2 lists the interference coefficients for this domain. Note that interference is assumed to be structural ($\mathbf{c}_{ij}$ = 1 or 0). As a starting point, and subject to validation, few interference effects are postulated for the sensor side of the auditory domain. It has been assumed that two simple auditory signals may interfere in the way that two or more musical tones forming a chord are qualitatively different to the individual notes that comprise it, which might be considered to be a form of structural interference. Making two vocal responses simultaneously, on the other hand, is an obvious case of structural interference. It is assumed that most auditory interference effects occur at the higher level of processing (see Table 3) which involve working memory (such as simultaneously speaking and attending to speech).

Table 2. Task interference coefficients (c_{ij}) for a human information processing model — auditory domain.

Channel	*Mode*	*Interference*
Audition	Input	Structural ($\mathbf{c}_{ij}$ = 1 or 0)
Vocalisation	Output	Structural ($\mathbf{c}_{ij}$ = 1 or 0)

CATEGORIES	CATEGORIES 0	1	2	3	4	5	6
0. None (no auditory component)	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
1. Tone or simple auditory signal	**0.0**	**1.0**	**0.0**	**0.0**	**0.0**	**0.0**	**1.0**
2. Speech input (*incidental to the primary task*)	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**1.0**
3. Auditory pattern	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**1.0**
4. Speech input (*attended to*, *salient to the primary task*)	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**1.0**
5. Voice output	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**1.0**	**1.0**
6. No allocation (default)	**0.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

This last point perhaps needs elaboration, as the IP/PCT model tries to make explicit the distinction between structural (physical or non information processing limitations) and cognitive (due entirely to information processing limitations). This is why Table 2 is quite sparse as it is expected that most interference effects would be accounted for in the mandated cognitive components of Table 3.

Cognitive domain

Within the cognitive domain, it is assumed that resource limited performance stems from the competition for common processing structures as discussed in connection with the IP/PCT model. Within this domain the degree of interference is graded, with coefficients taking values in the range 0 to 1 (see Table 3 for suggested values). At some level of interference it seems reasonable to assume that operators will perform tasks in a strictly serial fashion rather than resorting to time-multiplexing. The difference between serial and interleaved performance is seen in the position of tasks on the simulated time-line. For interleaved performance, task start times remain unchanged by interference effects, but the completion times of tasks are delayed. With strictly serial performance, one task is postponed until the other has been completed; therefore, there will be changes (delays) in both the task start and stop times. The times required for the tasks to be processed, however, are not modified in this case. Tentatively, this critical value ($\mathbf{c}_{\text{lin}}$) is set at $\mathbf{c}_{ij} \geq 0.7$.

Table 3. Task interference coefficients (c_{ij}) for a human information processing model — cognitive domain.

Channel	*Mode*	*Interference*
Cognition	Central	Resource Limited (see interference matrix — note that structural interference will take precedence over resource limitations, and the maximum value of $\mathbf{c}_{ij}$, over all domains, will be used to determine the outcome)

CATEGORIES	**CATEGORIES** 1	2	3	4	5	6
1. Automatised, highly learned	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
2. Passive monitoring of auditory signals (e.g., non-salient speech)	**0.0**	**0.0**	**0.2**	**0.0**	**0.1**	**1.0**
3. Verbal encoding, decoding, speech/sound production	**0.0**	**0.2**	**1.0**	**0.2**	**0.5**	**1.0**
4. Spatial encoding decoding, pattern recognition	**0.0**	**0.0**	**0.2**	**1.0**	**0.3**	**1.0**
5. Memorisation/recall, calculation, estimation, deduction, reasoning	**0.0**	**0.1**	**0.5**	**0.3**	**1.0**	**1.0**
6. No allocation (default)	**0.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

Note that the only interference coefficients that exceed 0.7 in Table 3, are those for which $i = j$ or involve the default category *no allocation*. However, as these are arbitrary assignments, the situation may change in the future. **Note that all operator tasks must be assigned a cognitive category**. The development of IP/PCT concepts in terms of Hierarchical Goal Analysis (HGA) (Hendy, Beevis, Lichacz*, et al.*, 2002) made the requirement for ANDed (the logic operator AND) cognitive categories imperative. This capability was first implemented in v3 of IPME.

Psychomotor and kinesthetic domain

The Psychomotor and Kinesthetic domain consists of two channels — tactile input and manual output. Interference is assumed to be structural (see Table 4). In general, as *digit* is a subset of *hand*, tasks involving different digits of the same hand will be assumed to interfere. Therefore, selection of *digit* will automatically invoke *hand*. Compatible combinations, such as might occur with Hand-on-Throttle-and-Stick (HOTAS) systems, will be made by exception (see the discussion on *Compatible Task Pairs* on p. 22); similarly for *Foot* and *Leg*. Keyboarding tasks, such as typing or operating a Control Display Unit (CDU) for a Flight Management System (FMS), involve the whole hand, so the need to consider each digit separately is unnecessary. The operation of rudder pedals and toe brakes is a compatible combination that would be added to the exception list.

Table 4. Task interference coefficients (c_{ij}) for a human information processing model — psychomotor and kinesthetic domain.

Channel	*Mode*	*Interference*
Tactile	Input	Structural ($\mathbf{c}_{ij}$ = 1 or 0)
Manual	Output	Structural ($\mathbf{c}_{ij}$ = 1 or 0)

CATEGORIES

Left Hand	Right Hand
None	*None*
Whole hand	*Whole hand*
Digit 1	*Digit 1*
Digit 2	*Digit 2*
Digit 3	*Digit 3*
Digit 4	*Digit 4*
Digit 5	*Digit 5*

CATEGORIES

Left Leg	Right Leg
None	*None*
Whole leg	*Whole leg*
Foot	*Foot*

No allocation (default)

CATEGORIES	YES	NO
Tasks use the same hand, leg, foot, finger, at least one task is not allocated etc.	**1.0**	**0.0**

Categories in this domain are ANDed (the logic operator AND) for complex tasks such as flight control. Hence, multiple category selections (e.g., right hand, left and right foot for a fixed wing aircraft) may be made for this domain.

Miscellaneous domain

A miscellaneous domain (Table 5) is included to account for effects that are not covered adequately by Tables 1, 2, 3 and 4. For example, one may wish to model team activities rather than individual operator tasks, redefine the categories for one of the domains, or introduce a new domain such as aided-vision. The miscellaneous domain provides some flexibility for accommodating additions such as these to the network of tasks. It is possible to edit both the names of the category fields (up to 15) and the name of the table for a specific application. These changes are reflected globally.

Table 5. Task interference coefficients (c_{ij}) for a human information processing model — a miscellaneous processing domain.

Channel	*Mode*	*Interference*
Miscellaneous	Unspecified	Structural or Resource limited.

CATEGORIES	CATEGORIES 0	1	2	3	4	5	...	*n*
0. None (default)	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**...**	**0.0**
1. Category 1	**0.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**...**	**1.0**
2. Category 2	**0.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**...**	**1.0**
3. Category 3	**0.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**...**	**1.0**
4. Category 4	**0.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**...**	**1.0**
5. Category 5	**0.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**...**	**1.0**
...		**...**	**...**	**...**	**...**	**...**	**...**	**...**
n. Category *n*	**0.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**...**	**1.0**

Combining interference coefficients

Equations (1), (2), (4), and (5) require a single value of $\mathbf{c}_{ij}$ for the calculation of interference effects. The value of $\mathbf{c}_{ij}$ used finally to describe the degree of task interference is the maximum value of the coefficients obtained from all active domains represented by Tables 1, 2, 3, 4, and 5, and the value of $\mathbf{c}_{ij}$ obtained from any transformations. Obviously, structural interference ($\mathbf{c}_{ij} = 1$), when present, will dominate, and will force strictly serial processing. Because all input and output channels (vision, audition, vocalisation, tactile, manual) invoke structural rather than resource limitations, cognition isn't a factor if a source of interference is at the sensor/effector level. Cognition enters the equation only when sensor/effector limitations are absent.

Combining interference coefficients in this way carries with it the assumption that domains (visual, auditory, cognitive, kinesthetic/psychomotor and miscellaneous) are ORed (the logic operator OR). Hence, any task might involve processing resources from one or more of these domains. Indeed, all tasks are assumed to have at least a cognitive component. Categories within domains, in general, are not ANDed (the logic operator AND) with two exceptions. One exception is for the kinesthetic/psychomotor domain where tasks might involve various combinations of digits, hands, arms and legs. The other exception is for cognitive tasks, and was driven by the requirements of the PCT-based HGA. The value of $\mathbf{c}_{ij}$ used in calculations is the maximum value of $\mathbf{c}_{ij}$ calculated for all pairwise combinations of the cognitive categories of each task.

Linking peripheral categories with central categories

Various types of input and output processes can be linked naturally with appropriate central processes (from Table 3) as shown in Table 6.

Table 6. *Natural linkages between input/output and perceptual/central processes.*

INPUT/SENSATION			COGNITIVE/PERCEPTUAL PROCESSES
VISION			
1.	Central		
	1.1	Text, dial reading	Verbal encoding
	1.2	Pattern, spatial relationship, tracking, graphic displays	Spatial encoding, visual pattern recognition
2	Peripheral		Automatized, highly learned perception
AUDITION			
1	Tone or simple auditory signal		Automatized, highly learned perception
2	Speech input (incidental to the primary task)		Passive (pre-attentive) monitoring of auditory signals
3	Auditory Pattern		Semantic (use verbal) decoding.
4	Auditory localization		Spatial decoding
5	Speech input (attended to, salient to the primary task)		Verbal decoding, speech recognition
KINESTHETIC.			
1	Tactile		
	1.1	Simple stimulus	Automatized, highly learned perception
	1.2	Complex stimulus	Spatial encoding.
MEMORY			
1.	Recall from memory		
	1.1	Accessible, familiar	Automatized
	1.2	Verbally coded	Verbal decoding
	1.3	Spatially coded	Spatial decoding
	1.4	Semantically coded	Semantic (use verbal) decoding
	1.5	Complex concept/operation	Recall
OUTPUT/BEHAVIOUR			**COGNITIVE PROCESSES**
VOICE			
1	Voice output		Speech production
PSYCHOMOTOR			
1	Manual output		
	1.1	Simple	Automatized, highly learned response
	1.2	Difficult but familiar	Spatial encoding
	1.3	Complex and or unfamiliar	Memorization/recall, calculation, estimation, deduction, reasoning
MEMORY			
1.	Storage in memory		
	1.1	Familiar concepts	Automatized
	1.2	Verbally coded	Verbal decoding
	1.3	Spatially coded	Spatial decoding
	1.4	Semantically coded	Semantic (use verbal) decoding
	1.5	Complex concept/operation	Memorization, calculation, estimation, deduction, reasoning

Later versions of IPME make these linkages directly by default. In most cases, the instances of input and output categories listed in Table 6 are already reflected in the interference matrices of IPME. Some new categories and sub-categories have been introduced as follows:

VISION domain

- two sub categories have been added under *Central vision*,

AUDITION domain

- a new category *Auditory localization*,

KINESTHETIC domain

- two new sub-categories under *Tactile*

MEMORY domain (Note: it is not necessary to create a new interference matrix or coefficients for MEMORY as it is absorbed into the cognitive domain)

- recall/storage from/in memory is a requirement arising from the move to goal based decompositions of human activities (Hendy, Beevis, Lichacz*, et al.*, 2002).

Allocation of Attention

In general, a task network could be said to simulate the demand placed on the operator by the system, rather than the task load actually serviced by the operator. A task network can have many parallel branches which lead to the generation of multiple concurrent demands. In many cases these demands clearly exceed human capabilities to respond (e.g., CMC, 1992; Glenn, Cohen, Wherry Jr., *et al*., 1994). While the IP/PCT model posits that many loops can be under simultaneous control, tasks that share common processing resources will compete for processing time. For the reasons stated previously, it has been assumed that no more than 2 high level tasks will be processed concurrently. The purpose of an *allocation of attention* module is to schedule the tasks to be performed, either serially or concurrently depending on their levels of interference, at any point in time. The allocation of attention module determines whether a task is performed on demand, interrupted, resumed, postponed, or shed. The allocation of attention algorithm is intended to provide a fair representation of human task selection strategies under competing system demands.

A rule base for the scheduler

When the task network generates a new demand (with the exception of certain special cases — see the discussion starting on p. 22), the following set of rules govern the scheduling of tasks. When new tasks arrive or an ongoing task finishes, a *temporary* queue is generated containing all tasks currently running (**the active task list**), together with any new tasks and tasks awaiting processing. Tasks awaiting processing are retrieved from a *short term memory* (STM) queue plus a *systems* queue for *externally cued visual tasks* (see p. 22). Not all tasks make this transition if memory has decayed (see p. 22 for details of *memory effects*). Note also that some externally cued tasks are special cases, both in the condition of their entry into the temporary queue and in their storage during the pre-attentive stage. A task that is programmed to occur in a cyclical fashion (see the discussion on p. 25 for *continuous* and *repeating* tasks) is not added to the temporary queue if a predecessor remains present in the short term memory queue or in the active task list. Neither will it be transferred to the short term memory queue following the current task scheduling.

Task scheduling is in accordance with the following rules based on *priority*, *interruptability*, *resumability* and *sheddability*. Note that, in general, *priority* might be time or state dependent (e.g., the priority of a display may increase with time since last glance, the priority of a task may change due to the occurrence of some predisposing condition — see discussion on *Task Performance Modifiers* starting on p. 35). Short term memory queue size is tracked. This queue is flushed on a *first opportunity* basis (i.e., as soon as an ongoing task finishes, the queue is examined to see if there are any tasks that can be started). Note that this does not account for physiological or psychological refractory periods which are generally of short duration, compared to the task completion times, and can therefore be ignored.

> *Rule 1*. Tasks transferred from the active task list, which are deemed to be *interruptable*, may be halted if less than C_{crit} complete (tentatively $C_{crit} = 70\%$). An interrupted task is returned to the STM memory queue (subject to Rule 7) and may be *resumed* (applying the start-up penalty as appropriate — see p. 89) or *restarted* later. An *uninterruptable* task, once started, must run to completion (as it is not actually interrupted the start-up

penalty μ is not applied). A task which is *not resumable* is *restarted* if possible (note that the time remaining on the task is set back at T_i when it is returned to the STM queue). Task interruptions are logged, non-resumable tasks are flagged when interrupted and the actual percentage of processing completed is recorded at the point of interruption.

Rule 2. Tasks, including interrupted tasks, are serviced in order of *priority*. Priority is determined by the value of the instantaneous time pressure for the task (see the definition of instantaneous time pressure p. 27). All task postponements are logged. Task(s) of the highest priority value(s) are serviced first with the exception of active uninterruptable tasks which take precedence once started.

Rule 3. Once the highest priority task is selected (taking into account the precedence relationship of uninterruptable tasks), if the 2nd ranked task has a level of interference with the first ranked task such that $\mathbf{c}_{ij} \geq \mathbf{c}_{\text{lin}}$, it is returned to the temporary queue and the next ranked task considered. This process is repeated until a compatible combination is found or until the queue is exhausted. For the purposes of starting a task, if it can be performed with an alternate hand, foot, or limb this is tried before rejecting the combination (see Rule 5). This event is logged. If the $\mathbf{c}_{ij}$ resulting from the initial selection is 0, the next highest priority tasks (down to and including Category 8 tasks in Table 8) is added to the current selection(s), in turn, until **any** pairing of the selections results in $0 < \mathbf{c}_{ij} < \mathbf{c}_{\text{lin}}$ (skipping over tasks for which $\mathbf{c}_{ij} \geq \mathbf{c}_{\text{lin}}$).

Rule 4. If several tasks have the same priority, they are scheduled according to the following hierarchy:

- in order of their originally scheduled start time (including non-resumable tasks that are restarting), and independent of the number of interruptions;
- in order of the least processing time remaining;
- according to least interference; and
- a random selection is made.

Rule 5. If a task can not be started due to interference effects, it is allocated to a less loaded channel if possible. This generally will occur for tactile and manual channels only.

Rule 6. Any remaining tasks in the temporary queue that are category 1 in the cognitive domain (assumed to be automatised or skill-based — see Table 3) are added to the active list provided there are no structural interference effects with the tasks **already** scheduled. Externally queued visual detection tasks (see the discussion on special cases beginning on p. 22), once accepted into the temporary queue, are always added to the active list.

Rule 7. The STM queue is limited to *m* items (tentatively *m* = 3; for example see Moray, 1986, page 40-27) and does not include items in the active task list. On transfer from the temporary queue, tasks are shed from the bottom of the priority list — *sheddable* tasks first — to meet this limit. If the limit can not be met with sheddable tasks, *non-sheddable* tasks are **forced** from the queue (but only after **all** sheddable tasks have been removed). The following hierarchy is used to determine the order of shedding among tasks of equal priority value. That is:

- in order of the most processing time remaining;
- according to most interference; and
- a random selection is made.

Tasks shed are logged. Tasks partially serviced when shed have the actual processing time completed logged (% complete). Repeating and continuous task components that are shed due to an unprocessed or incompletely processed predecessor, are also logged.

When a task is shed, a decision must be made with respect to the linked tasks that follow. Failure to do this will result in a potentially premature termination of a branch or indeed the whole network. When a task is shed (including forcible sheddings) there are 4 potential outcomes:

1. there is no effect on network integrity (no tasks follow);
2. the branch should terminate (the shed task is on a critical path and failure to process the task implies mission failure);
3. the next linked task should be started (this will be the default condition); and
4. another(other) task(s) should be initiated (this might be a deterministic, conditional or probabilistic branching).

Task shedding occurs in response to excessive task-loading, hence it would be inappropriate to start a following task the moment a predecessor is shed. To avoid the propagation of the overload condition through the network, any task linked to a shed task is not initiated until the shed task was due to finish (i.e., time of shedding + $\Delta T_{\text{act}}(n)$). Note that the task status variable *shed* can be used to effect the outcome of future activities in the network through various task performance modifiers (see the discussion starting on p. 35). Note also that Condition 4 includes rescheduling the task if it branches back to itself.

At the moment, the rule base for the scheduler involves crisp sets. Further developments could introduce a fuzzy rule set if this was deemed to be a better model of the human scheduling process. The notion of the human as a fuzzy adaptive controller is a particularly attractive analogue (Mancini, 1988).

Memory

Several interesting possibilities for modelling human performance flow from the scheduling algorithm. For example, the probability of a successful outcome for some tasks may decrease if they are interrupted or delayed, or a task may be dropped from the queue (forgotten) if not serviced within a certain time period — most likely the probability that an item is forgotten would increase with time or the nature of other tasks in the queue (see the discussion on memory in Card, Moran, and Newell, 1983).

A reasonable starting point is a simple memory model that forcibly sheds a task from the STM queue, with a certain probability, rather than transferring it to the temporary queue whenever a new task arrives or an ongoing task finishes. An exponential decay model is used as the default, namely:

$$\textit{probability of shedding} = \left(1 - 1.3591 e^{\frac{-t}{\sigma}}\right), \tag{8}$$

where, t is the elapsed time (in seconds) since the task was first scheduled to start, and σ is the memory decay time constant (i.e., the time to $p = 0.5$). It is possible for the analyst to program other relationships.

Exceptions and special cases

Compatible Task Pairs. There are likely to be pairs of tasks that although predicted to be structurally or resource limited to serial processing (tentatively for values of $\mathbf{c}_{ij} \geq \mathbf{c}_{\text{lin}}$) may be compatible for dual tasking in certain combinations (e.g., controlling aircraft pitch and roll with a joystick or control wheel involves the same hand but is a compatible combination). Allowance is made for these exceptions on a case by case basis. For these selected combinations of tasks, $\mathbf{c}_{ij}$ takes on new values (0 if structurally limited and $< \mathbf{c}_{\text{lin}}$ if resource limited — an internal check ensures that this is so) on a special case basis. This will only effect the $\mathbf{c}_{ij}$ values for these specific task-pairings and will not, in general, effect the $\mathbf{c}_{ij}$ values when these tasks are paired with any other tasks.

Externally Cued Visual Detection Tasks. In general, tasks can be **internally** cued (described in the literature as endogenous, top-down, volitional or goal-directed) or **externally** cued (exogenous, bottom-up, reflexive, or image based). Externally cued tasks must first capture the attention of the operator before they can be serviced. The probability that an externally cued visual task engages attention will determine the likelihood that the task enters the temporary queue and therefore is a candidate for operator processing. This allows the location of externally cued visual stimuli to effect the outcome of the task network simulation, say, through the probability of orienting to a new visual stimulus occurring in an area other than the *home* area.

The issue for the modeller is the probability that attention shifts to the external (exogenous) stimulus location and the dynamics of this process. Note that under the assumptions of Rizzolatti's pre-motor theory (Clark, 1999) the attentional shift occurs prior to any eye movement. The attention shift simply may be in response to the exogenous stimulus or it might start a subsequent endogenous or goal directed visual search in the area of the new attentional focus.

Computational models to describe the dynamic behaviour of attentional shifts are characterised by a "winner-take-all" strategy in which visual elements compete against each other to engage attention (Clark, 1999). Under this strategy, the saliency of one of the elements is maximally enhanced and all other elements are maximally inhibited. The "winner" becomes the new focus of attention and the target for a saccadic eye movement. The probability that an item in the external world becomes the new focus of attention is the probability that this item attains the greatest saliency at the expense of all other items. According to Clark (1999), when the input feature activity changes, the winner-take-all network will take on a new equilibrium and a new winning location will be generated.

Hence, when an externally cued visual task is generated by the network, the orienting component of the externally cued task is added to the temporary queue with a probability that depends on the nature of the task (whether it requires central or peripheral vision to engage the attentional mechanism — see Table 7) and the angular subtense measured either from the *home* region or from the region of the highest priority visual task currently selected (with central tasks taking precedence over peripheral — see the discussion on p. 11). The probability value from Table 7 may have a modifier to account for stimuli that decay or otherwise change in detectability .

EXTERNALLY INITIATED CATEGORIES	**ANGULAR SUBTENSE (DEGREES)**			
	Ø≤ 2	2<Ø<30	30≤Ø<90	90≤Ø≤180
1. Central	**1.0**	**0.5**	**0.0**	**0.0**
2. Peripheral	**1.0**	**1.0**	**0.5**	**0.0**
3. No allocation (default)	**1.0**	**1.0**	**1.0**	**1.0**

Table 7. Probabilities of detection for externally cued visual detection task.

Note that it is the responsibility of the analyst to create the orienting component of the task activity and correctly designate it as an *Externally Cued Visual Task* (from the categories in Table 8) as well as modelling the behaviour this task will initiate. The orientation component is envisaged as a short duration automatised task that merely signals the presence of an interesting visual stimulus. These tasks do not contribute to the instantaneous time pressure or occupy cognitive or short term memory resources. They are best modelled as fleeting transient processes (say of 20-50 ms duration).

An attempt to add the detection task to the temporary queue, is made only after **all** other internally cued task selections have been made — to determine the point of goal-directed vision — but before the selection is implemented. The externally cued visual task occurs prior to the saccade and while it does have a visual area designated to it, it does not change the direction of gaze (the point of gaze has already been determined by the active visual tasks or will default to the *home* position). The task that follows the externally cued visual task will capture the direction of gaze only if it becomes active.

Further attempts are made to add the detection task to the temporary queue, during the time that the visual stimulus is available (subject to the same rules as the first attempt). Until the task enters the temporary queue, it is held in **system** rather than **operator** memory. These further attempts to engage the attentional mechanism can be made only when there is a potential shift in visual attention (i.e., when the currently active visual task is completed, interrupted or shed). If the task fails an attempt to enter the temporary queue it is logged as a failure to detect. If the event marking the presence of the signal disappears from system memory before the detection task enters the temporary queue, it is logged as a missed detection. Obviously, if the detection task initiates a network of activities (the localisation, recognition and action initiation stages of the activity), they will be disabled if the detection phase fails. An externally queued visual detection task is serviced immediately (as an automatised task) it has entered the temporary queue (see Rule 3 of the scheduler on p. 20).

Other components of the activity (post orientation) may require higher level cognitive resources and thus automatic processing can not be assumed. The task components initiated by the orientation task are likely to be goal-directed (internally cued) and therefore will not, generally, fall under the special case described here.

In the IP/PCT implementation of externally cued visual tasks the probability of engaging the attentional mechanism has been assumed to be constant (for a given angular subtense) and independent of the time since the last opportunity to engage attention. This seems to be a reasonable interpretation of the concept that an overt engagement of visual attention (as signalled by a saccade) occurs when fixation at the current focus is released. As described by Clark (1999), this is triggered by transitions of the winner-take-all process. While the generation of a winner-take-all solution will have its own dynamic response characteristics, it has been assumed that the time constants are relatively short even in comparison with inter saccade intervals. The modeller can override these assumptions by applying task performance modifier functions to the probability fields of Table 7.

Externally Cued Speech and Auditory Pattern Recognition. Externally cued or attended speech input is either processed at the time it occurs or shed. Speech inputs will, generally, be externally cued (i.e., their time of occurrence is under the control of the speaker rather than the listener). It is assumed that signal-to-noise ratios are such that the presence of these speech signals passes the detection phase even if it is not possible to attend, due to higher priority concurrent tasks, at levels of processing such that the information content is transferred.

Therefore, as these tasks are assumed to pass the detection phase, they are added to the temporary queue as soon as they are generated by the network. Giving them Category 1 priority (see Table 8) makes them candidates for instant processing. They do not go into interim storage in system memory. Hence, speech is either processed at the time of arrival or forcibly shed from the queue and logged as a missed communication. There is no attempt to reschedule at a later time if the task is not serviced immediately due to the tie breaking provisions of Rule 4 of the scheduler or the presence of tasks with Time Pressure (*TP*) > 1. Hence, those tasks designated as Category 2 and 4 from Table 2 are shed, if externally cued, on the first attempt to transfer from the temporary queue to the short term memory queue. These tasks are shed before pruning the remaining tasks in the temporary queue down to the short term memory queue limit of *m* items. Auditory pattern recognition tasks designated as Category 3 in Table 3, **if externally cued**, are treated in the same way as the equivalent speech input.

Speech and auditory pattern recognition tasks might also be internally cued (i.e., under the control of the listener). These tasks are treated as any other task, rather than under the special case provisions discussed above. That is, they enter the temporary queue at the time of occurrence and are transferred to the STM queue for later processing, under the rules of the scheduler, if they cannot be serviced immediately. The rules associated with continuous and repeating tasks are invoked as appropriate.

Continuous and repeating tasks

Notionally continuous and repeating tasks are identical. The IP/PCT model treats continuous tasks as repetitive cycles of attending and not attending (Hendy, 1994b). It is assumed that, on average, the proportion of any given time interval required for the active processing of a continuous task, to achieve criterion performance, is directly proportional to the rated difficulty of the task. Hence, for continuous tasks, the average duty cycle is set by the difficulty rating D, where $0 < D \leq 1$.

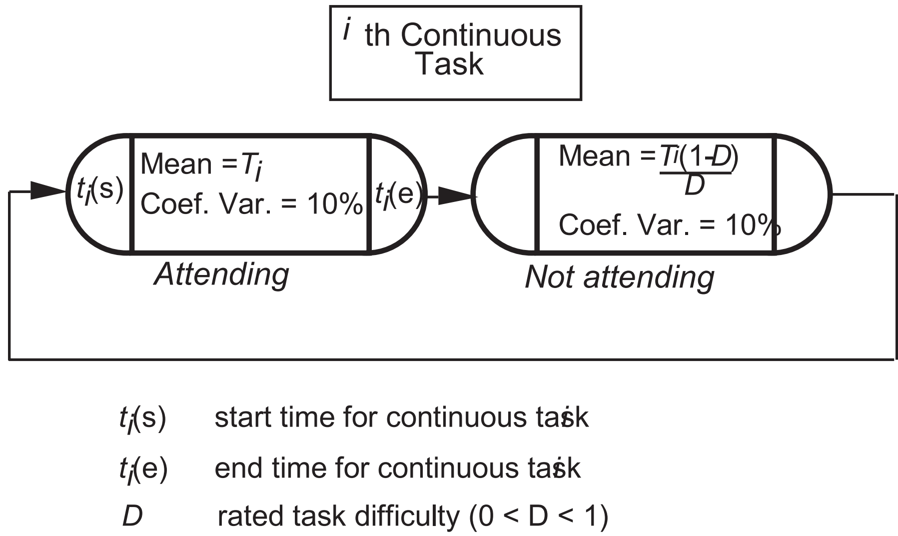

Figure 4. **Network representation of a continuous task. A repeating task differs only in the calculation of the mean non-attending time.**

The mean time interval required for active processing, during each cycle, is set by the analyst. A default value of 0.5 second is used for this dwell time, with a default coefficient of variation of 10%, and a duty cycle of 50% (a *Beta*, instead of a *Normal*, distribution could be used, in which case maximum and minimum values would be set). Hence, the network representation of a continuous task is shown in Figure 4. The starting and ending times for continuous tasks may be set by external factors such as the mission scenario, or from internal network states such as the activation of another task or some parameter taking a particular value or range of values.

Repeating tasks are treated similarly except that the *Attending* time and the mean *Cycle* time are set directly rather than through an intermediate parameter such as D. Hence, given that the mean *Attending* time of the *i*th repeating task is T_i, and the mean *Cycle* time is δT_i, then the mean *Non attending* time is $\delta T_i - T_i$.

Task Priority

Suppose tasks have associated with them a latest acceptable time for servicing ($t_i(\mathrm{L})$). This time might be related to the importance of the task to mission success, to an external event in the scenario, or to the status of certain variables in the network. The latest acceptable time for servicing is calculated once only when the task first arrives for servicing — it does not change with interruption and resumption cycles including tasks that are restarting (even though T_i is recalculated). There is one exception to this rule and that occurs when a task or task segment actually starts when $t \geq T_i(\mathrm{L})$.

Instantaneous time pressure

At any point (t) in time, after its generation by the network, the *instantaneous time pressure* (ITP) associated with the *i*th task is calculated as the ratio of

$$\frac{\text{effective processing time remaining on task } i}{\text{time remaining to latest processing time for task } i}$$

or

$$100\left[\frac{\Delta T_{eff}(n)}{t_i(\mathrm{L}) - t}\right]\%. \tag{9}$$

This process captures the essence of the *time available* part of the time pressure calculation in the IP/PCT model.

The *peak instantaneous time pressure* is the maximum time pressure value found for all tasks in the short term memory queue plus the active tasks. It is assumed that subjective reports of workload will be related to a moving average of the peak instantaneous time pressure calculated over the most recent load history.

Note that the instantaneous, and therefore the peak instantaneous time pressure can exceed 1 according to Equation 9 (but see also the following section on task shedding). While this does not cause computational difficulties, it does suggest that when interpreting peak instantaneous time pressure as predicted subjective workload, values be truncated at say 2.0. This would make it easier to interpret graphical results. While human operators may be able to cope with brief periods of overload, time pressures of 2.0 would most likely overwhelm even the most capable person. Any value over 1 should be flagged for investigation.

Task shedding and tasks that are late

If $\Delta T_{eff}(n) > t_i(\mathrm{L}) - t$, the instantaneous time pressure exceeds 1 for that task and it might be forcibly shed, or be logged as late and retained in the STM queue for future servicing. If it is deemed that the task should be forcibly shed, this condition overrides any other restrictions (for example see Table 8) that might be operating. These tasks are flushed from the temporary queue once the criterion has been met, that is as soon as the *TP* value is calculated and before the current processing interval proceeds. This establishes the time at which task shedding occurs. It also removes the value of ITP from the calculation of peak ITP.

Tasks for which time pressure exceeds 1 should be rare. For this to happen, two conditions have to be satisfied simultaneously, namely: $\Delta T_{eff}(n) > \mathrm{t}_i(\mathrm{L}) - t$; and the task has to be retained even though it won't be completed until after the nominated '*latest acceptable time for servicing*.' If time pressures exceed 1, then one might question whether the latest time for task completion has been set correctly, or if the task really should be shed if it is not completed before the target time.

However, if the task is not forcibly shed, the current task status is set to *late*, and a variable is set to TRUE. The same conditions apply when $t \geq T_i(\mathrm{L})$ with one exception. If the task or task segment is finally serviced with an actual start time that is later than or equal to the previously calculated latest processing time $\mathrm{t}_i(\mathrm{L})$, a new $\mathrm{t}_i'(\mathrm{L})$ is calculated with respect to the actual start time that is equal to $\left(t + \Delta T_{eff}(n)\right)$. This sets the initial value of the time pressure at *TP* = 1 and makes the task or task segment a candidate for immediate processing. These are the only conditions under which the latest processing time is recalculated.

A variable is set TRUE the first time the satisfying conditions apply for the task. It is not modified if these conditions subsequently change, for example, due to the recalculation of $\mathrm{t}_i'(\mathrm{L})$ or if the task is forcibly shed on a future cycle. The task status variable always tracks the current state of the task.

Latest acceptable time for servicing continuous and repeating tasks

For continuous tasks, the expected arrival of the next *attending* interval sets the latest processing time for the purposes of calculating instantaneous time pressure. Hence, for the *n*th occurrence (arriving at time $\mathrm{t}_{in}(\mathrm{a})$) of the *i*th continuous task with rated difficulty level D_i, the expected latest arrival time is given by:

$$\begin{aligned}\overline{\mathrm{t}_{in}(\mathrm{L})} &= \mathrm{t}_{in}(\mathrm{a}) + T_i + \frac{T_i(1-\mathrm{D}_i)}{\mathrm{D}_i} \\ &= \mathrm{t}_{in}(\mathrm{a}) + \frac{T_i}{\mathrm{D}_i},\end{aligned} \tag{10}$$

while, for the *n*th occurrence of the *i*th repeating task,

$$\overline{\mathrm{t}_{in}(\mathrm{L})} = \mathrm{t}_{in}(\mathrm{a}) + \delta T_i. \tag{11}$$

Task categories and priorities

In Table 8, tasks are categorised according to their time criticality. This categorisation establishes the initial priorities and time pressures for all tasks. For interuptable tasks, *interruptability* is temporarily disabled if the instantaneous time pressure for that task equals or exceeds a value tentatively chosen to be 0.8. This is the minimum value for the non-interruptable tasks designated as Categories 1 to 3 in Table 8.

Tasks nominated as *sheddable* do not contribute to the calculation of peak instantaneous time pressure. These are discretionary tasks and therefore do not logically have a $t_i(L)$ associated with them. This is really only an issue with the Category 7 tasks in Table 8 that may be nominated either as *sheddable* (in which case a $t_i(L)$ would not be calculated) or *non-sheddable*. For the sake of convenience in coding, Category 7 and 8 tasks designated as *sheddable* are given an arbitrary instantaneous time pressure, and therefore priority, of 0.001 rather than 0. This allows *sheddable* tasks to time multiplex with each other and avoids dividing by zero in Equation 6.

Task priority for the attentional mechanism is computed from the closeness of the current time to $t_i(L)$ with **tasks having the highest instantaneous time pressure being serviced first** (see Equation 9). Higher priority tasks also capture more of the processing time when competing with other tasks. The probability estimates for time-multiplexing the processing channel use normalised data and can therefore handle *priority* values (instantaneous time pressures) greater than 1.

For the orientation part of an externally cued visual task the issue is whether the stimulus was detected or not, rather than the priority of the task (orientation is assumed to be *pre-attentional*). Hence, these tasks do not contribute to the instantaneous time pressure or occupy cognitive or short term memory resources. Once the orientation is made, however, a string of tasks might follow which do load memory and contribute to the instantaneous time pressure. Because of the special status of the visual orientation task, it is given its own category in Table 8. Note that task shedding, if detection fails during the time the stimulus is present, is from the system memory rather than from human memory.

By giving externally cued speech and auditory pattern recognition tasks Category 1 status in Table 8, they enter the temporary queue with the highest initial priority and therefore are candidates for immediate servicing.

Because the effective time remaining on resumable tasks may be increased by a factor μ when interrupted, the priority of a resuming task can be effectively increased slightly due to an increase in the instantaneous time pressure (e.g., see Davis, 1971). This is over and above the increase in time pressure that results from any delays in task completion.

Table 8. Categorisation of tasks according to their latest processing times $t_i(L)$***.***

CATEGORY	TASK CHARACTERISTICS	$t_i(L)$	INTERRUPTABLE	SHEDDABLE
1	Requires instant reaction. Is critical for crew survival. No delays or interruptions are acceptable. Includes externally cued auditory tasks not because they are necessary for survival but because they are lost if not attended to.	$t_i(a) + T_i$	No	No¥
2	Requires priority attention. User defined delays between 0 and 25% of the task completion time. No	$t_i(a) + \kappa T_i$ $1 < \kappa < 1.25$	No	No

	interruptions are acceptable.			
3 (default)	Requires priority attention. Delays of up to 25% of the task completion times are acceptable. No interruptions are acceptable.	$\mathrm{t}_i(\mathrm{a}) + 1.25T_i$	No	No
4	User defined delays in excess of 25% of the task completion time. Tasks may or may not be interruptable.	$\mathrm{t}_i(\mathrm{a}) + \kappa T_i$ $\kappa > 1.25$	Yes†/No	No
	High Workload (0.7 to 0.8)	$\kappa = 1.4$ to 1.25		
	Medium Workload (0.4 to 0.7)	$\kappa = 2.5$ *to* 1.4		
	Low Workload (0.2 to 0.4)	$\kappa = 5.0$ *to* 2.5		
	Very Low Workload (0.1 to 0.2)	$\kappa = 10.0$ *to* 5.0		
	Low importance (0.001 to 0.1)	$\kappa = 1000$ *to* 10.0 see also Category 8		
5	Latest time for task completion is set by the scenario. Tasks may or may not be interruptable.	set externally	Yes†/No	No
6	Continuous task with rated difficulty level of D_i.	$\mathrm{t}_i(\mathrm{a}) + \dfrac{T_i}{\mathrm{D}_i}$	Yes†/No	No§
7	Repeating task with mean attending time T_i and mean cycle time ∂T_i. Tasks may or may not be interruptable or sheddable.	na€/$(\mathrm{t}_i(\mathrm{a}) + \delta T_i)$	Yes†/No	Yes/No§
8	Discretionary task. May be interrupted or shed.	na€	Yes	Yes
9	Externally cued visual detection task.	na	No	No‡

Notes for Table 8:

¥ *Externally cued speech and auditory pattern recognition tasks will be forcibly shed if they are not processed immediately as special cases of this category. Shedding occurs at the beginning of the processing interval.*

† *Depends also on the value of the instantaneous time pressure. When the instantaneous time pressure for the task exceeds 0.8, task interruptability for this task, if enabled, should be disabled. Interruptability is also disabled when a task is ≥ 70% complete. Neither of these rules will apply to Category 8 tasks.*

€

€ *Instantaneous time pressure is arbitrarily set at 0.001 for all sheddable tasks. This is a convenience to allow low priority discretionary tasks to time multiplex (see discussion p. 29).*

§ *Whenever its predecessor remains unprocessed, a repeating or continuous task will be shed. Shedding occurs at the beginning of the processing interval*.

‡ *Put in a temporary **system** queue (rather than **human** memory) if not serviced at the first opportunity. Finally shed if not serviced during the time the stimulus is present*.

Predicting Operator Workload and Performance

The IP/PCT model posits that operator load depends on the *time pressure*, or the ratio of *time required to process information* to the *time available*. In the context of task network simulation, the time required to process information is given by the task completion time of each activity. The concept of time pressure represents a return to a metric that has a long history in time-line analysis and task network simulation (e.g., see Linton, Plamondon, Dick*, et al*., 1989). The major difference between the current implementation and past usage, with the exception of Wingert's function interlace method (Wingert, 1973), is that with the current implementation, performance is sensitive to the presence of multiple concurrent tasks.

As an alternative to the percent time occupied prediction of operator workload, the value of the peak instantaneous time pressure across all active and memory queued tasks — actually a moving average over its recent time history — is used . This measure is directly related to the notion of time pressure as used in the IP/PCT model, namely the ratio of the actual processing time to the time available. In the words of Linton, Jahns and Chatelier (1977) "...As used by many systems engineers, workload is the extent to which an operator is occupied by a task relative to the time that is available for accomplishing the task."

Unlike the *percent time occupied* metric, an *instantaneous time pressure* measure retains its sensitivity to the composition of the task time-line even when the network is entirely task-driven. Hence, while the percent time occupied metric might speak to the *busyness* of the simulated operator, a moving average of the peak instantaneous time pressure is likely to be more closely related to the operator's perceived task-loading. Because sheddable tasks do not have a latest processing time associated with them, they do not effect the instantaneous time pressure value unless these tasks change status either permanently or temporarily. The IP/PCT model uses the average peak instantaneous time pressure as the predictor of operator workload.

Specifying the point of overload

While not strictly necessary for purely comparative studies, the specification of a load limit (a *redline*), for defining the point of operator *overload*, tends to be the Holy Grail of workload researchers. Typically, values of around 70 to 80% of *time-occupied* are chosen. These values appear to be supported by little more than observations that this marks the point at which load shedding starts. However, empirical evidence is not offered in support of these claims (Meister, 1985, p. 78).

For purposes of illustration, suppose that the problem is framed in terms of a single server queuing problem (Campbell, 1989). Note that in the IP/PCT model, the single server is sometimes taking customers two at a time. In this situation, some predictions might be made as to what would constitute a point of overload. If tasks are assumed to arrive according to a Poisson process at a constant mean rate of λ tasks s^{-1}, and the mean task completion rate also remains constant at a value of μ tasks s^{-1} (assume task completion times are exponentially distributed), then, in a given fixed time interval δt:

mean time occupied = mean number of tasks × mean task completion time

$= \delta t \lambda \mu^{-1}$, and

mean time pressure $= (\delta t \lambda \mu^{-1}) \delta t^{-1}$

$= \lambda / \mu$.

Equating λ with the mean arrival rate of the queuing problem, and μ with the mean service rate (Hillier and Lieberman, 1974), it can be seen that the mean *time pressure* is equivalent to the *utilisation factor* ρ of the queuing problem or a classical single server system, the steady state number of items in the queue is 1 at $\rho = 0.5$, rising to approximately 2 at $\rho = 0.7$, and 4 at $\rho = 0.8$ (Hillier and Lieberman, 1974, Fig 9.6). If the queuing analogy is valid, it seems that a value of *time pressure* around 0.75 is a reasonable limit to set. This would hold the steady state queue size to 2-3 items.

Memory queue size and task shedding

The issue of setting limits on time pressure assumes less importance if overload is redefined in terms of the length and status of the short term 'memory' queue in the *allocation of attention* module. Of particular interest are occasions of forceful load shedding from this short term storage. This approach strikes directly at what might in fact be the underlying problem of operator overload, and avoids problems associated with the arbitrary selection of parameters for the moving average, particularly in the case of the time occupied metric. In view of the need to distinguish between discretionary and non-discretionary tasks when computing operator load, this shift from a traditional workload paradigm to a concern for tasks serviced versus tasks shed is particularly salient.

Basically, the analysis shifts from a concern for *workload* to an interest in *errors* (task shed, delayed, etc.) and the development of system status knowledge by tracking the proportion of tasks serviced that contribute to situation assessment. Using this approach, task shedding is tracked and categorised by the type of information involved. Flags distinguish between tasks that are critical to <u>mission performance</u>, and those that contribute to <u>awareness of the mission or the system</u>. Note that in the IP/PCT model, operator error is associated with information unprocessed or shed (Hendy, *et al.*, 1997).

Task Performance Modifiers (TPMs)

Task network simulation has the potential to be sensitive to a number of task performance modifying influences as might be derived from aspects such as: manpower (crew size), personnel (aptitude, command, experience), training (knowledge, skills), fatigue and other physiological stressors, allocation to an alternative processing channel (another sensory channel or operator), operator adaptation to high information processing loads, and various psychological stressors. The network properties that are available to implement this potential are (Hendy, Koberski and Youngson, 1992):

- individual task inventory;
- various task properties;
- task sequence, including branching due to conditional or probabilistic task outcomes (e.g., resulting from a changed probability of successful task completion);
- task completion time; and
- tasks serviced and shed.

In this Section, a number of TPMs are advanced to account for the effects of:

- time pressure and channelised visual attention;
- predecessor tasks;
- changing priority values;
- aptitude and experience;
- physiological/psychological stressors; and
- the use of non-preferred channels.

These effects are assessed for their influence on: task inventory and sequence through the probability of successful completion and detection probabilities; task completion time; and task attributes such as *priority* and *domain category*.

It is realised that the application of multiple factors has the potential for unwanted cumulative effects. An attempt has been made to resolve the most obvious conflicts by establishing rules for combining multiple factors that effect task completion times or probabilities. At this stage, to implement a starting suite of task performance modifiers, it is sufficient to have access to, or be able to derive, certain variables such as:

- number of interruptions for each task;
- the (mean) peak instantaneous time pressure;
- time since the task segment commenced;
- a list of active tasks and tasks in the short term memory queue;
- number of unsuccessful scheduling attempts for each task;
- % processing completed on each task;
- status of a specific task;
- elapsed time since first scheduled start time of each task; and
- categories (from Tables 1, 2, 3, 4, and 5) of all tasks in the queue.

Where these task performance modifiers effect bounded variables, a check is made to see that parameter values do not stray outside of their limits, for example

- probability values remain in the range {0,1}, and

- category values stay within the range for the parameter etc.

TPMs are called as Functions in IPME.

Application and implementation of TPMs

Generally the TPMs described in this Section are applied whenever a task first becomes available for processing. In some cases, TPMs are applied each time the allocation of attention module is called. It is not intended, in general, that the transformations would be recursive in cases when there are multiple calls during the time a task is active or queued waiting for processing time. To avoid compounding the results of successive transformations, the original values of all task parameters should be used each time the task performance modifier is calculated. Table 9 specifies the conditions that govern the application of the task performance modifiers discussed in this Section.

Table 9. Application of transformations involving Task Performance Modifiers in the IP/PCT model.

		Applies whenever the Allocation of Attention Module is called	
Task Performance Modifier due to	**Applies at Task Start**	**At Beginning of Processing Interval**	**At End of Processing Interval**
Aptitude/Experience	Yes		No current examples
Non-preferred Limb	Yes		
Physiological/Psychological Stressors	Yes		
Task History	Yes		
Time Pressure(speed/accuracy)	Yes		
Task Priority Transformations		Yes	
Time Pressure(visual detection)		Yes	

	Applies to			
Task Performance Modifier due to	**Task completion time**	**Prob. of detection (Table 7)**	**Prob. of Success (p. 37)**	**Task Property**
Aptitude/Experience	Yes		Yes	Yes
Non-preferred Limb	Yes		Yes	
Physiological/Psychological Stressors	Yes		Yes	
Task History	Yes		Yes	
Time Pressure(speed/accuracy)	Yes		Yes	
Task Priority Transformations				Yes
Time Pressure(visual detection)		Yes		

Table 9 lists variables that might be affected by the TPMs. This incomplete list is limited to task completion times, the probability of detecting a visual stimulus, the probability of a successful task outcome, and finally task attributes such as priority and domain category.

For TPMs that apply at the onset of a task a Function should be inserted into the *Task Beginnings Effects* field. For those that apply at the beginning of a processing interval, the Function call should be made from the *Scheduling Effects* field. For those that are evaluated when a task completes, the Function should be inserted in the *Task Ending Effects* field.

Changes in strategy due to time pressure

According to the IP/PCT model, at some critical point, as time pressure increases towards 1, operators will attempt to adopt less accurate but more timely strategies (this is the classical speed/accuracy trade-off). Typically, this will involve a reduction in the monitoring of outcomes from emitted behaviours. Hence, while task completion time might decrease as a function of *TP*, the probability that the loop has been closed to the desired level of accuracy is expected to decrease also. This could be represented in task network simulation terms by associating probabilistic outcomes with all tasks (see Figure 4).

In Figure 4, each task has two potential outcomes. In one case it is assumed, with a probability *p*, that the goal has been achieved to the required accuracy. This is the notion of a task *successfully completed*. Alternatively, with a probability (1 - *p*), it is assumed that the required accuracy has not been met and the task is considered to be *unsuccessfully completed*. In Figure 4, a task performance modifier $\mathrm{tpm}(\overline{TP})$, for task completion time and the probability of goal achievement, are both shown to be functions of the time pressure. The critical value TP_{crit} at which the simulated operator attempts to adapt to increasing time pressure by a change in strategy (e.g., see Sperandio, 1978; Seifert, 1980) might be associated with the 70-80% *redline* value discussed previously. Category 1, 2 and 3 tasks in Table 8 might be exempted from this speed accuracy trade-off as it is assumed that these tasks will always be performed in the most expeditious manner. An automatised skill-based task (Category 1 for the cognitive channel — see Table 3) might also be exempted.

It seems to be a reasonable assumption that the human will respond to global rather than local, or individual task, demands. Therefore, the task performance modifier would be calculated from a moving average of the peak instantaneous time pressure. This creates the potential for generating a hysteresis effect (or a sustained performance decrement following a peak load), such as has been observed in various time critical tasks (Cumming and Croft, 1973; East, 1993; Hicks, 1993; Smolensky, 1990). Because the moving average retains historical information from recently experienced periods of high load, the 'perceived' time pressure will be higher than the actual time pressure until the moving average has time to decay. This will keep performance from recovering when the load is relaxed due to the action of the time pressure related TPM. The window for the moving average is set by the analyst, with a default value of 1 minute. It is possible to set the width of the window for the moving average down to zero.

The Task Performance Modifier (TPM) shown in Figure 5 can be used to change the 'native' task completion time (T_i) of a task as follows. Let the working value for the task completion time of the *i*th task (T_i'), computed at the time an attempt is made to service the task, be given by

$$T_i' = \mathrm{tpm}(\overline{TP})T_i. \qquad (12)$$

$\overline{TP}$ is the mean peak instantaneous time pressure immediately preceding the time at which an attempt is made to service the task, and $\mathrm{tpm}(\overline{TP})$ is a time pressure dependent task performance modifier which is bounded in the range 0 to 1. T_i is the run time value for the current call of the *i*th task, and therefore is a product of the statistical distribution underlying the calculation of task completion times for that task.

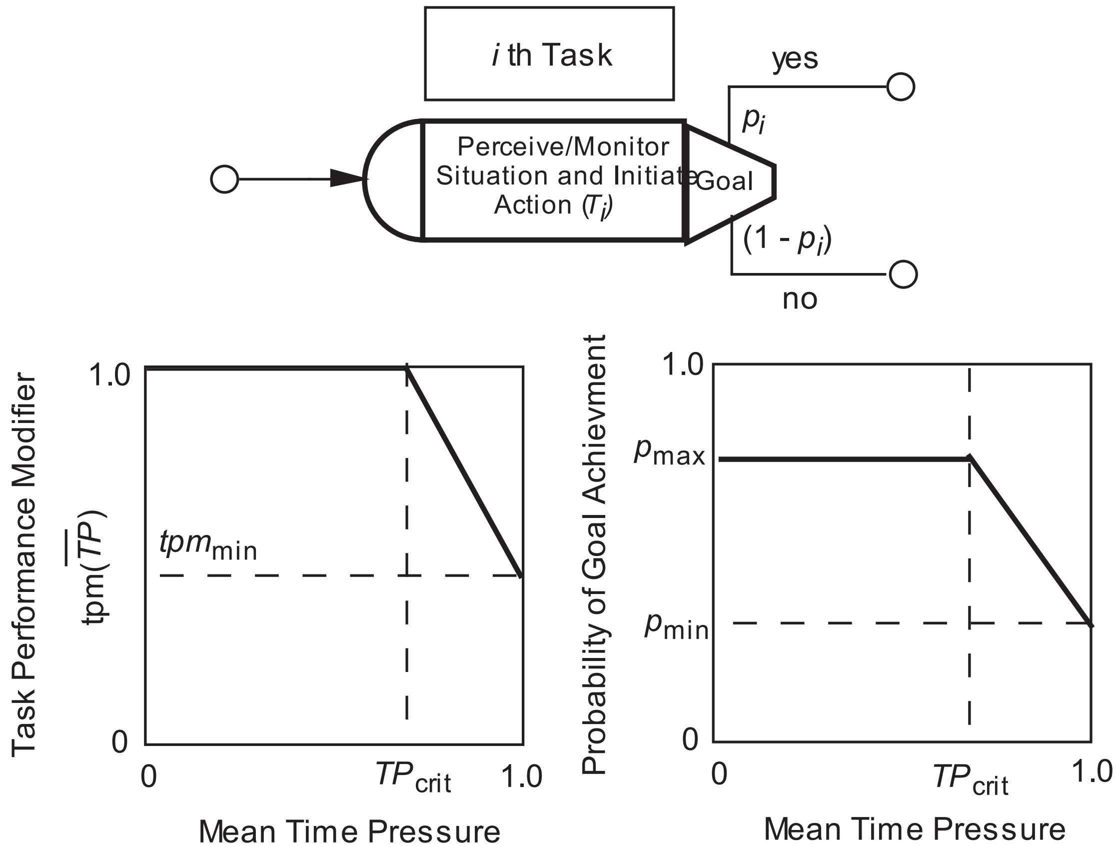

Figure 5. **Incorporating the PCT effects of changing strategies on task completion times and the probability of goal achievement through a Task Performance Modifier.**

Once a task is started T_i' **is not** recalculated if the task is interrupted and **resumed.** However**,** T_i' **is** recalculated if the task is interrupted and **restarted**. Obviously, this correction needs to be applied before the tasks are sorted in order of priority. The correction for the effective time remaining, discussed previously (see p. 10), is applied if the task is subsequently interrupted. A Function inserted into the Task Beginning Effects field would be an appropriate way to implement this type of TPM.

When the outcome of goal achievement on task completion is *no* (setting the *unsuccessful completion* flag), the result may be to repeat the task (this is not the same as interrupting the task and restarting), this time with the task performance modifier due to time pressure set to 1 and the probability of goal achievement set to p_{max} (i.e., reverting to the more accurate but slower strategy). The *no* path may simply rejoin the main network, or, alternatively, initiate a whole new pathway through the network. The value of the status variable can be used at a later date to modulate the future course of the network. For example, the outcome of a future activity might depend on the successful completion of the current task. In essence, these are the same types of outcomes that result from task shedding with the exception that network flow is not necessarily interrupted by an unsuccessful task completion. Therefore task shedding can be seen as a special case of unsuccessful task completion. Note, however, that a task might invoke either effect depending on the circumstances.

Probability of detection and channelised attention

As time pressure increases, operators often channel their attention and become less sensitive to events occurring outside their primary focus. This concept is already built into the implementation for the visual channel where a high task-driven load will capture attention away from the *home* region which can effect the probability of attending to a subsequent visual task if detection is made conditional on the current visual focus. With stochastic scheduling, the probability that an arriving visual task can be attended to depends on the mean time during which the visual channel is allocated to other visual tasks during the expected arrival time of the new task.

In contrast, the probability of attending to a new visual or auditory task could also be made conditional on the value of the mean peak instantaneous time pressure. This represents an attempt to model the inhibitory effect of high task-loading on the attentional mechanism. This effect is seen to be acting pre-attentionally and therefore overrides any prioritisation due to time pressure as it prevents the task from being added to the short term memory queue. For vision, this is easily accommodated by making the delimiting values in Table 7 functions of the time pressure. For example, the *p*-values in Table 7 could be modified as follows:

$$\text{prob. of detection (see Table 2)} = p \times \text{tpm}(\overline{TP}) \,. \tag{13}$$

Alternatively, for stimuli that decay with time:

$$\text{prob. of detection (see Table 2)} = p \times \text{tpm}(\delta t) \,, \tag{14}$$

where, $\text{tpm}(\delta t)$ is some function of the time since the stimulus first occurred (typically an exponential decay function).

Predecessor tasks and task history

As time pressure increases, task shedding will become more prevalent. In the IP/PCT model, error is assumed to depend on the amount of relevant information presented but left unprocessed. At the level of task description used in most network simulations (rarely lower than the button pushing stage), tasks are usually considered completed or not. However, either the time to complete and/or the probability of successful completion of some tasks may depend on the successful completion of various predecessor tasks. For example: the time to respond to an emergency may depend on the extent to which systems states have been monitored recently; a radio can not be set unless the message giving the channel setting was attended to; etc. Hence, the outcome of a task can be conditional on the successful completion of predecessor tasks. The TPM would be assembled from standard algebraic and logical relationships applying to *Function* calls.

Task category transformations

Under certain conditions, one might wish to modify the initial priority values of one or more tasks by changing the Category of a task from Table 8. The IP/PCT implementation supports various transformations that may be applied to change task categories during run time. Note that these transformation will effect only the calculation of $t_i(\mathrm{L})$. Originally assigned values of *interruptability* and *sheddability* are retained. Note that *interruptability* is automatically disabled when the time pressure exceeds a critical value or when the task is greater than a certain percentage completed (see notes for Table 8). Note also that the values of these parameters are set by the analyst, with defaults of 0.8 and 70%, respectively.

Experience and aptitude

The IP/PCT model posits that different levels of experience, knowledge, etc. result in different choices of strategies for processing, which effects both the total amount of information to be processed (hence the processing time) and the processing structure involved (from automatised, perhaps dedicated, structures to algorithmic problem solving involving calculation, recall, use of working memory, etc.). These effects could be modelled within a task network environment by changing the task completion times by an appropriate factor and changing the cognitive category of some tasks from Category 1 (automatised) to Category 4 or 5 (spatial encoding decoding, pattern recognition; or memorisation, recall, calculation, estimation, deduction, reasoning) and from Category 4 to Category 5.

The experience and aptitude values would be set as operator traits. The actual tasks that are effected by experience and aptitude factors are set at the local or task level.

Non-preferred channel

When a task is allocated *a priori* to a preferred processing channel (e.g., a limb or digit), it is assumed that a time penalty may occur if it is re-allocated, at run time, to a non-preferred channel (e.g., performed by the left rather than the right hand). A fixed penalty of 10% might be assumed. More elaborate rules may be developed that are dependent on the target for re-allocation.

Physiological and psychological stress factors

Various physiological and psychological states (temperature, noise, vibration, g-stress, chemical agents, drugs, fear, fatigue, anxiety, motivation, etc.) may work singly or in combination to change task completion times and possibly effect the strategies used to solve problems (the IP/PCT model). This mechanism allows the possibility for linking task models with physiological models (Jensen, 1994). In the simplest form, environmental stressors could be linked to events in the scenario which drive the task network. For example, temperature could increase during a mission, g-stress could be introduced to all tasks associated with an air-to-air engagement, etc.

Suppose various physiological stressors have been defined. The variables could be assigned names and used in TPMs to modify predicted performance, for example:

stressor (1) = acceleration,
stressor (2) = temperature,
stressor (3) = vibration,
stressor (4) = fatigue,
stressor (5) = etc.

As an example of a functional relationship, CREWCUT (see Little, Dahl, Plott*, et al*., 1993) multiplies task completion times by a factor which is a function of the time on task to account for fatigue, as follows;

$$\text{tpm(fatigue)} = \frac{1}{\left[0.25\left(0.93^{t}\right) + 0.75\right]} \tag{15}$$

where, *t* is the number of hours of steady task performance. For those tasks that have a probabilistic outcome, error rate may be affected also. This relationship is given for demonstration purposes only. Other mathematical forms are possible.

Combined stressors

It cannot be assumed that stressors in combination will act either synergistically or antagonistically. It is likely that complex interactions will occur and the implementation allows freedom for user defined functions and relationships to be inserted, based on standard mathematical and logical forms. Later, fuzzy rule sets may be seen to be more appropriate.

As a starting point, it is assumed that modifying effects are additive whenever they effect completion time (this will be the default condition and possibly reflects a 'worst case' situation). Hence, the original task completion time is multiplied by a factor which represents the sum of the percentage changes called for by all individual active stressors. For example, if

tpm(stressor 1) = 1.15 (+15%),
tpm(stressor 2) = 0.95 (-5%),
tpm(stressor 3) = 1.07 (+7%),
etc.,

then

$T_i' = 1.17T_i$ (15% - 5% +7%...).

Note that when discussing combined stressors, probabilities are multiplicative.

This page intentionally left blank.

Discussion

This document outlines the implementation of an information processing (IP/PCT) model for use in task network simulations. This implementation includes a representation of the operator's allocation of attention and human memory, together with a framework for tracking the load on the operator's information processing system. The framework for this implementation is provided by Hendy, *et al*.'s (1997) IP and Powers' (1973) PCT models. In positing that human information processing load is determined by the ratio of *time required* to *time available*, the IP/PCT model returns to an approach which has many precedents in the history of task network simulation. However, in recent years classical time-based predictions of operator load have largely given way to procedures that attempt to apply the tenets of resource theory to the problem. Current resource-based techniques owe much of their inspiration to the original work of Aldrich, Craddock and McCracken (1984) which in turn has its roots in Wickens' Multiple Resource Theory (Wickens, 1992, p. 375). These methods largely grew out of an attempt to address deficiencies seen to exist with the traditional time-based approach. These deficiencies include:

- the lack of a theoretical underpinning for the T_r/T_a ratio (where T_r is the time required to process the information and T_a is the time available);
- a limitation to serial processing, with the exception of Wingert's (1973) function interlace method — alternatively the inability to discriminate between single and multiple task performance;
- the insensitivity of time-based methods to the difficulty of continuous tasks; and
- the necessity to treat continuous and discrete tasks separately.

The IP/PCT model, as implemented in IPME, answers each of these criticisms. Firstly, the IP/PCT model provides the theoretical framework for claiming time pressure is the primary driver of operator workload, performance and errors. Secondly, the concept of interference in multiple task performance is implemented through the interference coefficients of Tables 1 to 5. Finally, by treating continuous tasks as repetitive discrete activities, the problems associated with the difficulty of continuous tasks and the combination of continuous and discrete tasks are addressed.

The introduction of an allocation of attention module allows the focus of the IP/PCT model implementation to shift from solely being that of *workload*, to having a greater emphasis on operator *performance* and *error*. In competitive systems (e.g., most military systems), the operator is likely to be always maximally (100%) loaded, as survival in a hostile environment may depend on constant information processing. The issue then becomes '...loaded with what?' For example, how much time does the operator have to scan the outside scene for targets, how much time is devoted to systems monitoring and navigational updates. etc? Hence, the issue for the designer is not so much to reduce the workload (utilisation of the timeline) imposed by the system, but to maximise the level of performance attainable at a given level of operator information processing capacity. The IP/PCT model implementation attempts to give that insight by tracking the tasks actually serviced in times of excessive task-loading.

Neither the IP/PCT model, nor the implementation of that model described in this report, can be considered to be all-encompassing. There will be many aspects of human behaviour that do not fall within the purview of the model and indeed these may be determined more by the nature and limitations of task network simulation than by the structure of the information processing model itself. While these gaps are lamentable, to the extent that they cannot be addressed by fine tuning the model, the IP/PCT model should be evaluated in terms of the following criteria.

Firstly	Is the IP/PCT model either better than alternative models that satisfy the constraints of the task network simulation environment, or is it at least as good and simpler to implement?
Secondly	Is the final product good enough to provide cost effective and useful predictions that might result in better systems design?

Arguably the IP/PCT model satisfies the first criteria. Its assumptions can be traced to a rational theoretical framework, although the final arbitration must wait until competing models can be fully tested in some type of common environment.

The advantages of a structured approach to front-end design, of which workload/performance prediction is a component, is generally accepted. Therefore, if the IP/PCT model passes on the first criteria it has some claim of at least face validity on the second. What is really needed is some indication of the percentage of variance that can be reliably accounted for with these methods. Unfortunately, definitive data are rare to non-existent, although impressive claims for correlations in the 0.8 to 0.9 range have been made for individual cases (e.g., Bateman and Thompson, 1986; Iavecchia, Linton, Bittner Jr., *et al*., 1989). Again the comparative assessment of models in a competitive environment should allow benchmarks to be established.

Conclusions

This document describes a time-based information processing model of the human operator, and derives various relationships and rules necessary for implementing these ideas in a task network simulation environment. The implementation of the IP/PCT model in IPME demonstrates that quite complex models can be embedded in task network environments within the limited degrees of freedom offered. This implementation covers both an allocation of attention module for scheduling tasks, and methods for predicting operator load and performance from the resulting task demand. The model that provides the framework for this report has been partially validated; however, within the context of this implementation further validation is necessary. Many of the parameters required by various elements of the implementation have been assigned arbitrarily and need to be verified.

The approach that follows from the theoretical framework offered by the IP/PCT model breaks from recent trends for workload prediction, which have been dominated by resource theory models, to focus once again on the time domain — specifically time pressure — as the prime driver of operator processing load, errors and performance. In returning to methods which appear at first glance to be similar to traditional *time occupied* models, the pitfalls inherent in these established procedures have been avoided. The framework for the implementation described in this document incorporates aspects of both serial and parallel processing, acknowledges task interference in multi-task situations, and handles both continuous and discrete tasks. Further, by returning to the time domain, model predictions are more readily testable.

This implementation of the IP/PCT model balances the more traditional focus on workload assessment with an emphasis on operator performance and error. This is achieved by tracking the tasks serviced by the allocation of attention module and logging and categorising the tasks shed, interrupted or postponed. This change of focus seems reasonable for competitive military systems, where the reduction of operator workload (in the sense of utilising the timeline) is probably not an achievable goal. For such systems, the aim of the designer should be to maximise overall system performance for a given level (generally maximal) of expended operator information processing capacity.

This page intentionally left blank.

References

Aldrich, T. B., Craddock, W., and McCracken, J. H. (1984). A computer analysis to predict crew workload during LHX scout-attack missions: Volume 1 (MDA903-81-C-0504/ASI479-054-I-84(B)). Fort Rucker, AL, USA: United States Army Research Institute Field Unit.

Bateman, R. P., and Thompson, M. W. (1986). Correlation of predicted workload with actual workload measured using the Subjective Workload Assessment Technique (SAE AeroTech). Warrendale, PA, USA: Society of Automotive Engineers.

Campbell, E. L. (1989). The application of queuing theory to the modelling of CP-140 aircraft communications (DCIEM No. 89-TR-23). North York, Ontario, Canada: Defence and Civil Institute of Environmental Medicine.

Card, S. K., Moran, T. P., and Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, New Jersey, USA: Lawrence Erlbaum Associates.

Clark, J. J. (1999). Spatial Attention and latencies of saccadic eye movements. *Vision Research*, 39(3), 583-600.

CMC (1992). Aurora Update evaluation: concluding report (DSS Contract No. W8477-9-AC33/01-QF for DND, Ottawa, Canada). Kanata, Ontario, Canada: Canadian Marconi Company.

Cumming, R. W., and Croft, P. G. (1973). Human information processing under varying task demand. *Ergonomics*, 16(5), 581-586.

Davis, M. (1971). Application of fast-time simulation techniques to the study of ATC systems. *Ergonomics*, 14(1), 661-668.

Detweiler, M., and Schneider, W. (1991). Modeling the acquisition of dual-task skill in a connectionist/control architecture. In, D. L. Damos (Ed.), *Multiple-task Performance*. London, United Kingdom: Taylor and Francis Ltd. 69-99.

East, K. P. (1993). *Operator performance following overload: Is 'Hysteresis' a factor in complex cognitive tasks*. Unpublished Honours Thesis. Waterloo, Ontario: Wilfrid Laurier University.

Glenn, F., Cohen, D., Wherry Jr., R., and Carmody, M. (1994). Development and validation of a workload assessment technique for cockpit function allocation. In, K. C. Hendy (Ed.), *Proceedings of the 'Workshop on Task Network Simulation for Human-Machine System Design'*. Washington, DC, USA: The Technical Cooperation Program, Subgroup U, Technical Panel 7. 59-88.

Hawkins, H. L., Olbrich-Rodriguez, E., Halloran, T. O., Ketcum, R. D., Bachmann, D. B., and Reicher, G. M. (1979). Preparation cost and dual task performance: further evidence against a general time sharing factor (Contract No. N14-77-C-643). Arlington, VA, USA: Personnel and Training Research Program, Office of Naval Research.

Hendy, K. C. (1994a). Implementation of a human information processing model for task network simulation (DCIEM No 94-40). North York, Ontario, Canada: Defence and Civil Institute of Environmental Medicine.

Hendy, K. C. (1994b). An information processing model for workload and performance prediction. In, K. C. Hendy (Ed.), *Proceedings of the 'Workshop on Task Network Simulation for Human-Machine System Design'*. Washington, DC, USA: The Technical Cooperation Program, Subgroup U, Technical Panel 7. 25-45.

Hendy, K. C. (1994c). Survey of national practices in task network simulation for human machine systems. Washington, DC, USA: The Technical Cooperation Program, Subgroup U, Technical Panel 7.

Hendy, K. C., Beevis, D., Lichacz, F., and Edwards, J. L. (2002). Analyzing the cognitive system from a perceptual control theory point of view. In, M. D. McNeese and M. A. Vidulich (Eds.), *Cognitive Systems Engineering in Military Aviation Environments: Avoiding Cogminutia Fragmentosa!* Wright Patterson AFB, Dayton, OH, USA: Human Systems Information Analysis Center. 201-250.

Hendy, K. C., East, K. P., and Farrell, P. S. E. (2001). An information processing model of operator stress and performance. In, P. A. Hancock and P. A. Desmond (Eds.), Stress, Workload and Fatigue: Theory, Research, and Practice. Mahwah, NJ, USA: Lawrence Erlbaum Associates, Publishers. 34-80.

Hendy, K. C., Kobierski, R. D., and Youngson, G. A. (1992). The integration of human factors engineering analyses with manpower and personnel studies. In, *Technical Proceedings AC/243(Panel 8)TP/5 — Workshop on Liveware Integration Needs*. Brussels, Belgium: NATO Defence Research Group, Research Study Group 21, Panel 8. 8-1 to 8-11.

Hendy, K. C., Liao, J., and Milgram, P. (1997). Combining time and intensity effects in assessing operator information processing load. *Human Factors*, 39(1), 30-47.

Hendy, K. C., and Lichacz, F. (1999). Controlling error in the cockpit. In, R. S. Jensen (Ed.) *Proceedings of the 10th International Symposium on Aviation Psychology. Columbus*, OH, USA: The Aviation Psychology Laboratory, The Ohio State University. 658-663.

Hicks, M. (1993). AWAS (Aircrew Workload Assessment System) analytical workload modelling tool — research and development programme. In, *Proceedings of the RAS Conference on 'Workload Assessment and Aviation Safety'*. London, United Kingdom: Royal Aeronautical Society. 4.1-4.8.

Hillier, F. S., and Lieberman, G. J. (1974). *Operations Research (Second Edition)*. San Francisco, CA, USA: Holden-Day, Inc.

Iavecchia, H. P., Linton, P. M., Bittner Jr., A. C., and Byers, J. C. (1989). Operator workload in the UH-60A Black Hawk: crew results vs. TAWL model predictions. In, *Proceedings of the Human Factors Society's 33rd Annual Meeting*. Santa Monica, CA, USA: Human Factors Society. 1481-1485.

Jensen, S. (1994). Scenario profiling — using task network simulations of task analysis and workload as the backbone for integrated human factors studies. In, K. C. Hendy (Ed.), *Proceedings of the 'Workshop on Task Network Simulation for Human-Machine System Design'*. Washington, DC, USA: The Technical Cooperation Program, Subgroup U, Technical Panel 7. 89-94.

Kinchla, R. A. (1980). The measurement of attention. In, R. S. Nickerson (Ed.), *Attention and Performance VIII*. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, Publishers. 213-238.

Linton, P. M., Jahns, D. W., and Chatelier, P. R. (1977). Operator workload assessment model: an evaluation of a VF/VA-V/STOL system. In, *Proceedings of the AGARD Conference on 'Methods to Assess Workload' (AGARD-CPP-216)*. Neuilly-sur-Seine, France: Advisory Group for Aerospace Research and Development. A12-1 to A12-11.

Linton, P. M., Plamondon, B. D., Dick, A. O., Bittner Jr., A. C., and Christ, R. E. (1989). Operator workload for military system acquisition. In, G. R. McMillan, D. Beevis, E.

Salas, M. H. Strub, R. Sutton and L. van Breda (Eds.), *Application of Human Performance Models to System Design*. New York, NY, USA: Plenum Press. 21-45.

Little, R., Dahl, S., Plott, B., Wickens, C., Powers, J., Tillman, B., Davilla, D., and Hutchins, C. (1993). Crew Reduction in Armoured Vehicles Ergonomic Study (CRAVES) (ARL-CR-80). Aberdeen Proving Ground, MD, USA: US Army Research Laboratory, Human Research and Engineering Directorate.

Mancini, G. (1988). Modelling humans and machines. In, L. P. Goodstein, H. B. Andersen and S. E. Olsen (Eds.), *Tasks, Errors and Mental Models*. London, United Kingdom: Taylor and Francis. 278-292.

Meister, D. (1985). *Behavioral Analysis and Measurement Methods*. New York, NY, USA: John Wiley and Sons.

Moray, N. (1986). Monitoring behavior and supervisory control. In, K. R. Boff, L. Kaufman and J. P. Thomas (Eds.), *Handbook of Perception and Performance: Volume II, Cognitive Processes and Performance*. New York, NY, USA: John Wiley and Sons. 40-1 to 40-51.

Powers, W. T. (1973). Feedback: beyond behaviorism. *Science*, 179(4071), 351-356.

Rasmussen, J. (1983). Skills, rules and knowledge; signals, signs and symbols and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-13(3), 257-266.

Seifert, R. (1980). Man-machine system performance as a function of the system demand/workload relationship. In, J. Moraal and K.-F. Kraiss (Eds.), *Pre-prints of the Proceedings of the Conference on Manned Systems Design*. NATO. 160-180.

Shaffer, M. T., Hendy, K. C., and White, L. R. (1988). An empirically validated task analysis (EVTA) of low level army helicopter operations. In, *Proceedings of the Human Factors Society's 32nd Annual Meeting*. Santa Monica, CA, USA: Human Factors Society. 178-182.

Smolensky, M. W. (1990). *The effect of work load history on operational errors in air traffic control simulation: the hysteresis effect — expectancy perseverance or short-term-memory overload?* Unpublished PhD Thesis. Texas, USA: Texas Technical University.

Sperandio, J.-C. (1978). The regulation of working methods as a function of workload among air traffic controllers. *Ergonomics*, 21(3), 195-202.

Wickens, C. D. (1992). *Engineering Psychology and Human Performance* (Second Edition). New York, NY, USA: Harper Collins Publishers Inc.

Wingert, J. W. (1973). Function interlace modifications to analytic workload prediction. In, K. Cross and J. McGrath (Eds.), *Crewstation Design: An Interagency Conference*. Santa Barbara, CA, USA: Anacapa Sciences. 373-378.

This page intentionally left blank.

Appendix 1: Derivation of task completion times under time-multiplexing

This Appendix describes the derivation of task completion times for interfering tasks. Assume that multiple interfering tasks are processed by a combination of time-multiplexed processing, together with processing unaffected by interference effects.

Let,

$t_i(\mathrm{s})$ be the start time of task *i*,

$t_i(\mathrm{e})$ be the ending time of task *i*, when performed alone,

$t_{ij}(\mathrm{e})$ be the ending time of task *i*, when performed with task *j*,

$\mathbf{c}_{ij}$ be the proportion of time that the overlapping tasks share a common processing structure (for simplification, this time is assumed to be evenly distributed throughout the overlap), and

p_i be the probability that in any given time interval, processing resources will be devoted to task *i*, rather than task *j* (note that $p_i = (1 - p_j)$).

Suppose,

$$\mathbf{c}_{ij} = \mathbf{c}_{ji},$$
$$T_i = t_i(\mathrm{e}) - t_i(\mathrm{s}),$$
$$T_{ij} = t_{ij}(\mathrm{e}) - t_i(\mathrm{s}), \text{ and}$$
$$t_j(\mathrm{s}) - t_i(\mathrm{s}) \geq 0.$$

Then two cases need to be considered (see Fig. A1.1):

CASE 1 — in which the processing of one of the tasks (assume it to be task *j*) is entirely embedded within the processing time of the other (task *i*); and

CASE 2 — in which the tasks partially overlap (the processing time of task *i*, remaining after task *j* starts, is entirely embedded within the processing time of task *j*).

CASE 1: Tasks Completely Overlap

In any small time interval δt, it is assumed that for $\mathbf{c}_{ij}\delta t$, due to interference effects, time-multiplexed processing takes place, while for time $(1 - \mathbf{c}_{ij})\delta t$ processing is unaffected by interference effects. Hence in the interval δt there is

$$(1 - p_i)\mathbf{c}_{ij}\delta t + (1 - \mathbf{c}_{ij})\delta t$$

processed of task *j*. Therefore, the number of intervals required to fully process task *j* is

$$\frac{T_j}{(1-p_i)\mathbf{c}_{ij}\delta t + (1-\mathbf{c}_{ij})\delta t},$$

for a total task completion time of

$$T_{ji} = \frac{T_j}{(1-p_i)\mathbf{c}_{ij}\delta t + (1-\mathbf{c}_{ij})\delta t}\delta t$$
$$= \frac{T_j}{1-p_i\mathbf{c}_{ij}}.$$

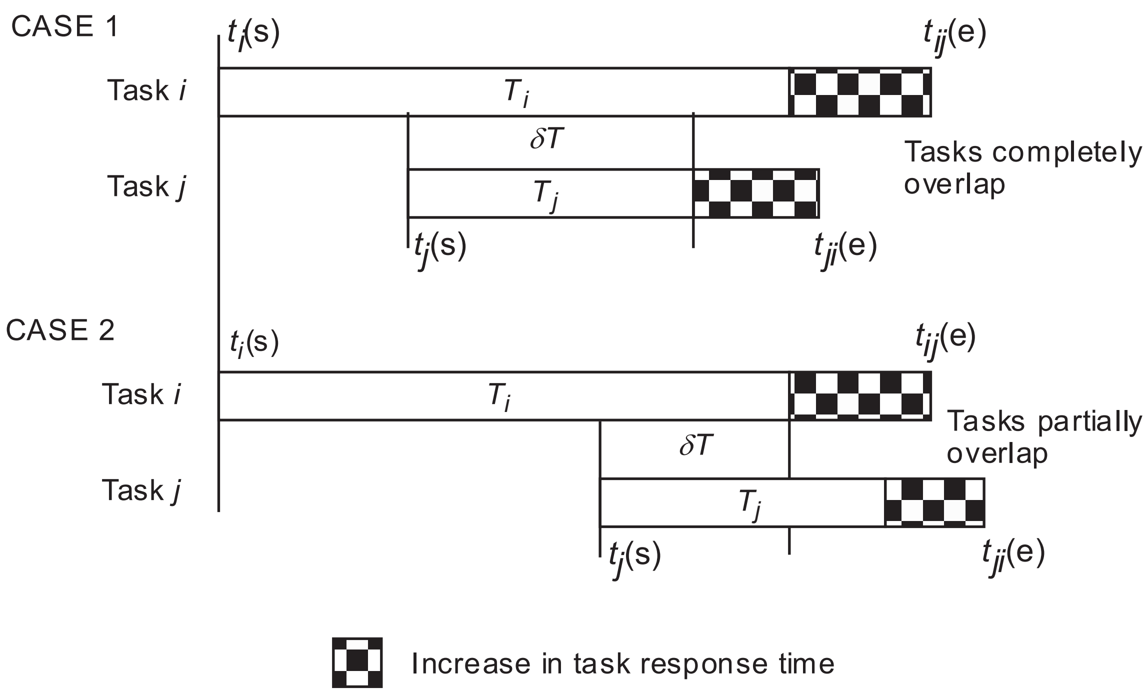

Figure A1.1. *Two cases of task overlap.*

During the period T_{ji}, $\mathbf{c}_{ij}T_{ji}$ involves time-multiplexed processing, while $(1-\mathbf{c}_{ij})T_{ji}$ involves processing unaffected by interference effects. Hence, during the time of overlap between tasks i and j,

$$p_i\mathbf{c}_{ij}T_{ji} + (1-\mathbf{c}_{ij})T_{ji}$$

is processed of task i. Therefore the time to fully process task i is

$$T_{ij} = T_{ji} + \left[T_i - \left\{p_i \mathbf{c}_{ij} T_{ji} + \left(1 - \mathbf{c}_{ij}\right) T_{ji}\right\}\right]$$
$$= T_i + \frac{\mathbf{c}_{ij}\left(1 - p_i\right)}{\left(1 - p_i \mathbf{c}_{ij}\right)} T_j .$$

Sufficient and necessary conditions for CASE 1 are defined by

$$t_j(\mathrm{s}) - t_i(\mathrm{s}) \geq 0, \text{ and}$$

$$t_{ij}(\mathrm{e}) - t_{ji}(\mathrm{e}) \geq 0.$$

Therefore,

$$\left\{t_i(\mathrm{s}) + T_{ij}\right\} - \left\{t_j(\mathrm{s}) + T_{ji}\right\} \geq 0, \text{ or}$$

$$T_i - T_j \left[\frac{1 - \mathbf{c}_{ij}\left(1 - p_i\right)}{1 - p_i \mathbf{c}_{ij}}\right] - \left\{t_j(\mathrm{s}) - t_i(\mathrm{s})\right\} \geq 0.$$

To allow for task interruptions it is necessary to keep a running total of the amount of actual processing time devoted to each task. At any time $t_j(\mathrm{s}) \leq t \leq t_{ji}(\mathrm{e})$, the amount of processing time devoted to task *i*, since task *j* commenced, is

$$\Delta T_{ij} = p_i \mathbf{c}_{ij} \left\{t - t_j(\mathrm{s})\right\} + \left(1 - \mathbf{c}_{ij}\right)\left\{t - t_j(\mathrm{s})\right\}$$

and to task *j*

$$\Delta T_{ji} = \left(1 - p_i\right)\mathbf{c}_{ij} \left\{t - t_j(\mathrm{s})\right\} + \left(1 - \mathbf{c}_{ij}\right)\left\{t - t_j(\mathrm{s})\right\}.$$

Therefore, at time *t* the amount of processing time remaining on each task is

$$T_i - \sum_j \Delta T_{ij} \text{ for task } i\text{, and}$$

$$T_j - \sum_i \Delta T_{ji} \text{ for task } j.$$

CASE 2: Tasks Partially Overlap

For CASE 2, not all of task *j* is processed in concert with task *i*. However, the processing time of task *i* $\left(t_i(\mathrm{e}) - t_j(\mathrm{s})\right)$, remaining after the start of task *j*, is embedded within the processing time of task *j*. Therefore, CASE 2 is similar to CASE 1, with the partial processing of task *i* equivalent to task *j* in CASE 1. Hence the processing time of task *i* is

$$
\begin{aligned}
T_{ij} &= \{t_j(\mathrm{s}) - t_i(\mathrm{s})\} + \frac{t_i(\mathrm{e}) - t_j(\mathrm{s})}{1 - p_j \mathbf{c}_{ij}} \\
&= t_j(\mathrm{s}) - t_i(\mathrm{s}) + \frac{t_i(\mathrm{s}) + T_i - t_j(\mathrm{s})}{1 - p_j \mathbf{c}_{ij}} \\
&= \frac{T_i - p_j \mathbf{c}_{ij} \{t_j(\mathrm{s}) - t_i(\mathrm{s})\}}{1 - p_j \mathbf{c}_{ij}}.
\end{aligned}
$$

Similarly, noting the symmetry with CASE 1, the processing time for task *j* is

$$
\begin{aligned}
T_{ji} &= \frac{\mathbf{c}_{ij}(1 - p_j)}{1 - p_j \mathbf{c}_{ij}} \{t_i(\mathrm{e}) - t_j(\mathrm{s})\} + T_j \\
&= \frac{(1 - p_j \mathbf{c}_{ij}) T_j + \mathbf{c}_{ij}(1 - p_j) T_i - \mathbf{c}_{ij}(1 - p_j)\{t_j(\mathrm{s}) - t_i(\mathrm{s})\}}{1 - p_j \mathbf{c}_{ij}},
\end{aligned}
$$

noting that $p_j = 1 - p_i$.

For the running total of actual time processed on each task, the equations for CASE 1 apply.

Abreviations and Acronyms

CDU	Control Display Unit
FMS	Flight management System
HGA	Hierarchical Goal Analysis
HOTAS	Hands-on-throttle-and-stick
IP	Information Processing
IPME	Integrated Performance Modelling Environment
ITP	Instantaneous Time Pressure
PCT	Perceptual Control Theory
POC	Performance Operating Characteristics
STM	Short Term Memory
TP	Time Pressure
TPM	Task Performance Modifiers

DOCUMENT CONTROL DATA

(Security classification of the title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document, Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's document, or tasking agency, are entered in section 8.) Publishing: DRDC Toronto, PO Box 2000, 1133 Sheppard Av West, Toronto, ON, CANADA M3M 3B9 Performing: DRDC Toronto, PO Box 2000, 1133 Sheppard Av West, Toronto, ON, CANADA M3M 3B9 Monitoring: Contracting:	2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.) UNCLASSIFIED (NON-CONTROLLED GOODS) DMC A REVIEW: GCEC JUNE 2010

3. TITLE (The complete document title as indicated on the title page. Its classification is indicated by the appropriate abbreviation (S, C, R, or U) in parenthesis at the end of the title)

An Introduction to the IP/PCT Model Implementation in IPME (U)
(U)

4. AUTHORS (First name, middle initial and last name. If military, show rank, e.g. Maj. John E. Doe.)

Keith Hendy

5. DATE OF PUBLICATION (Month and year of publication of document.)	6a NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)	6b. NO. OF REFS (Total cited in document.)
April 2010	63	62

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of document, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Report

8. SPONSORING ACTIVITY (The names of the department project office or laboratory sponsoring the research and development − include address.)

Sponsoring:

Tasking:

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant under which the document was written. Please specify whether project or grant.)	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)
44ga01	
10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document)	**10b. OTHER DOCUMENT NO(s).** (Any other numbers under which may be assigned this document either by the originator or by the sponsor.)
DRDC Toronto TR 2010−040	

11. DOCUMENT AVAILABILITY (Any limitations on the dissemination of the document, other than those imposed by security classification.)

Unlimited distribution

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11), However, when further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited announcement

UNCLASSIFIED

DOCUMENT CONTROL DATA

(Security classification of the title, body of abstract and indexing annotation must be entered when the overall document is classified)

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

(U) This document provides a description of the Information Processing (IP)/Perceptual Control Theory (PCT) model implemented in the Integrated Performance Modelling Environment software by Micro Analysis and Design (IPME). The current document is an edited and reduced version of an earlier report. In the current document, an attempt has been made to improve readability through a reorganisation of the material and the elimination of content that is not central to understanding the function of the IP/PCT model within IPME. The essence of the IP model is that all factors that impact on human cognitive workload can be reduced to their effects on the amount of information to be processed and the amount of time available before the decision has to be actioned. From this position, it can be shown that if humans are limited at the rate at which they process information then operator workload, performance, and error production are all functions of the time pressure. The IP Model is about time and the information to be processed.

The PCT Model argues that humans behave as multi–layered closed loop control systems. The set points for these control loops are our perceptual goals (or how we want to see, hear, feel, taste, or smell the state of the world). According to PCT, we sense the world state, forming a perception of that state which we then compare with our goal. If there is a difference between our perceived and desired states, we formulate an action. This action is implemented in order to operate on the world so as to drive the perceived state of the variables of interest towards the goal. The perceptual processes and the decisional processes draw on internal knowledge states that transform sensation to perception, and difference to action. Our attentional mechanism shifts our focus from loop to loop to loop. The PCT model is therefore about Goals, Attention, Knowledge and Feedback.

(U) Ce document fournit une description des modèles de traitement de l’information (TI) et de la théorie du contrôle perceptuel (TCP) mis en œuvre dans le logiciel Environnement intégré de modélisation des performances (EIMP) par Micro Analysis and Design. Ce document est une version réduite et modifiée d’un précédent rapport. Dans le présent document, nous avons tenté d’améliorer la lisibilité grâce à une réorganisation du matériel et à une élimination du contenu qui n’est pas essentiel à la compréhension de la fonction des modèles de TI et de la TCP au sein de l’EIMP. L’essence du modèle de TI est que tous les facteurs qui touchent la charge de travail cognitive de l’être humain peuvent être réduits à leur effet sur la quantité d’information à traiter et le temps requis pour prendre la décision. De ce point de vue, il est démontré que si les êtres humains sont restreints à la vitesse à la quelle ils traitent l’information, alors la charge de travail, la performance et la production d’erreurs de l’opérateur sont toutes fonction des contraintes de temps. Le modèle TI porte sur le temps et l’information à traiter.

Selon le modèle de la TCP, les êtres humains se comportent comme des systèmes de commande en boucle fermée (ou asservi) à de multiples couches. Les valeurs de réglage de ces boucles de contrôle sont nos objectifs perceptuels (ou comment nous voulons voir, entendre, toucher, goûter ou sentir l’état du monde). Selon la TCP, nous sentons l’état du monde, formons une perception de cet état, et nous le comparons ensuite à nos objectifs. En cas de différence entre l’état perçu et l’état désiré, nous formulons une action. La mise en œuvre de cette action vise à agir sur la réalité en vue de faire évoluer l’état perçu des variables d’intérêt vers l’objectif visé. Les processus perceptuels et les processus décisionnels se fondent sur le savoir en mémoire pour transformer une sensation en perception, et un écart en action. Nos mécanismes attentionnels dirigent notre attention

de boucle en boucle. Le modèle de la TCP est donc centré sur les objectifs, l'attention, le savoir et la rétroaction.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

(U) operator workload; human modelling; IPME;IP/PCT model

Defence R&D Canada

Canada's Leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE DÉFENSE

www.drdc-rddc.gc.ca