11th International Command and Control Research and Technology Symposium

"Coalition Command and Control in the Networked Era"

Paper#: **I-014**

Title: **Building Executable Architectures of Net Enabled Operations Using State Machines to Simulate Concurrent Activities**

Topics: C2 Modeling and Simulation
C2 Architecture
C2 Analysis

Authors: Ronald Funk
Joint Staff Operational Research Team – Team Leader
Centre for Operational Research and Analysis
Ottawa, Ontario, Canada

Mark Ball
Joint Staff Operational Research Team
Centre for Operational Research and Analysis
Ottawa, Ontario, Canada

Richard Sorensen
Senior Systems Engineering Consultant
Vitech Corporation
Vienna, Virginia, USA

Point of Contact:

Ronald Funk
Joint Staff Operational Research Team – Team Leader
Centre for Operational Research and Analysis
National Defence Headquarters
101 Colonel By Drive
K1A 0K2

Office: (613) 995-6887
Fax: (613) 992-3342
Ron.Funk@drdc-rddc.gc.ca

Building Executable Architectures of Net Enabled Operations Using State Machines to Simulate Concurrent Activities

By
Ronald Funk, Mark Ball and Richard Sorensen

Abstract

The Defence Research and Development Canada (DRDC) Centre for Operational Research and Analysis (CORA) is developing capability engineering analysis tools to help build and assess Net-Centric architectures. This paper describes the research into using a State-Machine (SM) model to simulate job workflow that can either be represented as a serial Task, Process, Exploit and Disseminate (TPED) cycle or as a concurrent Task, Post, Process, Use (TPPU) cycle.

The research supports the Canadian Forces (CF) Command, Control, Computers, Communications Intelligence, Surveillance and Reconnaissance (C4ISR) Campaign Plan [1] thrust to define requirements for future capabilities and make measurable investment decisions. It does so by developing realistic "As-Is" and "Target" Command and Control (C2) architectures that capability engineers can then use to analyze detailed options.

Classical behavioural models cannot simulate TPPU beyond simple cases due to the possible permutations in the job flow. SM models can overcome this limitation by storing the status of activities and then use them in the next time slice as inputs to change the status through an action or an output.

This paper provides a detailed description of the SM model and business logic implemented to date. Sample results and analysis tools for the model are also presented.

Background

In 2003, the DRDC Collaborative Capability Definition, Engineering and Management (CapDEM) Technology Demonstration (TD) project started development of system engineering processes to support the CF. The development effort was organized around and leveraged by the United States Department of Defense Architecture Framework (DoDAF) [2]. The DRDC CORA Joint Staff Operational Research Team (JSORT) participation focussed on developing an analytical framework that capability engineering staff could use to build and assess executable Net-Centric architectures. The software application used in the research of how to build executable architectures was the Vitech Corporation CORE/COREsim® software application [3].

It is relatively simple to learn about the principles of DoDAF but it is much more difficult to make them work in practice. JSORT started its modeling efforts with the conceptual and Operations Centre (OPCEN) TPED processes to learn what executable architectures need. The emphasis was on learning how to do Model-Based System Engineering (MBSE). The major results of this initial effort were a much better understanding of the modelling issues along with a belief that DoDAF and MBSE key enablers to building executable architectures.

JSORT also used this approach to model an Integrated ISR Architecture (IISRA) of the Atlantic Littoral Intelligence, Surveillance and Reconnaissance (ISR) Experiment (ALIX) [4]. The Modelling and Simulation (M&S) of ALIX IISRA covered two distinct activity areas relevant to functions that take place in any OPCEN. The first aspect was a temporal behaviour model used to track changes C2 in relationship during the flight of an Uninhabited Air Vehicle (UAV). The second aspect investigated how the TPPU cycles interact at the operator level to enable self-synchronizing concurrent workflows.

A major conclusion of ALIX IISRA was that a temporal chain of events is relatively easy to implement using classical behaviour models. Unfortunately, these models do not adequately mimic concurrent activity in a net-centric architecture.

During a feedback session about ALIX IISRA, some prior work came to light involving conversion of an architecture behavioural model into a SM [5]. The participants agreed that the OPCEN workflow logic for operators processing jobs appears to behave like a SM.

The OPCEN SM model described here leverages the ALIX IISRA analysis and personal observations of the OPCEN activity. The data and results displayed here are reasonable approximations of the observed behaviour but the values used are fictitious in order to avoid any security classification issues. This paper is derived from work presented at [6] and documented in [7].

Description of Two Workflow Options

Modelling Generic Task, Process, Exploit & Disseminate

TPED is the process cycle that intelligence staffs and many commercial businesses use to organize and conduct most activities. The main feature of TPED is that it clearly lays out an orderly set of serial events used to determine what the customer needs (Task), convert the resulting data to a form that analysts can work with (Process), do the actual work to produce a useful product (Exploit) and finally get it to the customer (Disseminate).

TPED can also be used to describe the workflow of jobs within an OPCEN at the level of the individual worker. Jobs are assigned by a central coordinator and arrive in the analyst queue (Task). Each job has an priority that is used to assign the order in which it is to be tackled (Process). The analyst selects the highest priority job and works on it until he completes his part of the job (Exploit). The final product is then approved and released to the customer (Disseminate). The TPED approach ensures the orderly delivery of products and maximizes the utilization of available resources.

Modeling Generic Task, Post, Process and Use

The TPPU cycle [8,9] achieves TPED effects while permitting concurrent use of partially analyzed information—with the goal of improving the Commander's Situational Awareness (SA). The method of achieving timely analysis and use of the information via TPPU is through decentralized work activity that becomes self-synchronizing in a network-centric environment.

The 'Task' portion of TPPU is embedded in the queue and business logic while the actual work being done by the Job Threads is actually the 'Process' and 'Post' portions of TPPU. 'Use' is a residual activity that takes advantage of the unallocated time between jobs to monitor for SA and deal with routine or administrative jobs.

Combined Effect of TPPU Within TPED Setting

Figure 1 illustrates the basic TPPU cycle at the level of an operator and its basic structure is essentially the same as for TPED. The queue is treated as a standing Task where jobs are accumulated and sorted by order of priority (Task). The highest priority job is selected from the queue and the rest of the network is notified that it is started along with any initial information (Post). The job is then worked on (Process) and the incremental products are updated at logical steps in the process until a final product is generated and posted for others (Use).

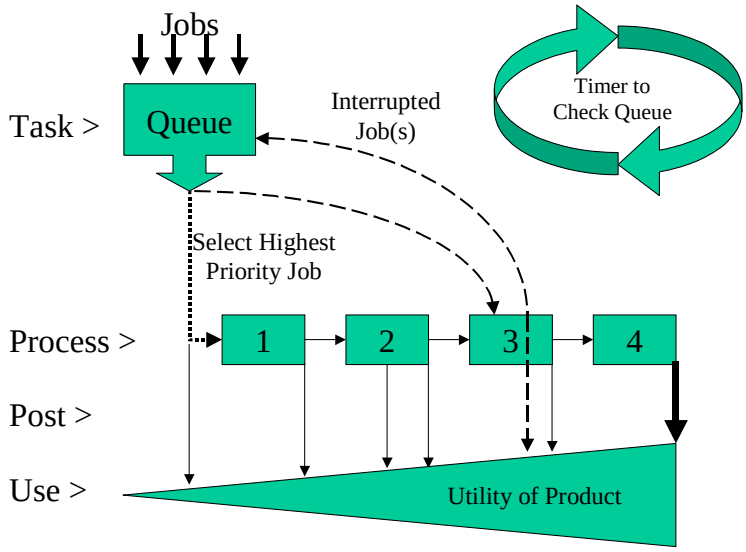

Figure 1. Workflow Through a Generic TPPU Cycle

The concept of TPPU is analogous to the way any office worker conducts his daily activity. TPED is a simplified view of the overall flow of work while TPPU recognizes that workers regularly check their email queue and reflexively treat phone calls as their top priority. They also make drafts of work and rarely work on only one job from start to completion.

The major characteristic of TPPU is that it explicitly recognizes the need for analysts to frequently check the job arrival queue for higher priority jobs. Any lower priority job is pre-empted and the highest priority job is worked on until it is in turn pre-empted or completed. TPPU tackles each job in priority order until the queue is empty.

Processing of the TPPU queue involves the operator posting their work periodically and then checking to see if a higher priority job has arrived. Each time the queue is checked it is dynamically updated and the highest priority job is compared to the one being worked upon. Any job pre-empted by higher priority jobs is returned to the queue and annotated to credit work completed so that later on it can be restarted from where it was interrupted.

When TPPU checks queues frequently the effect is the self-synchronization of decentralized and uncoordinated independent activities. The downside that limits the frequency of checking queues is that it takes time to perform this activity. Otherwise all the time set aside to gain and maintain SA can be lost by continually checking the queues.

Comparison of TPED and TPPU Utilization Metric

The inclusion of SA into the equation is expected to increase the overall effectiveness of TPPU because the distributed effort allows more eyes to notice something than TPED allows by limiting it to those assigned it as a job. A potential downside to TPPU is that the incomplete products could add to the complexity of integrating the information and thereby confuse the SA.

The reality is that both TPPU and TPED are actually working towards the same overall objective. The short repetitive TPPU cycles actually lead to a linear sequence of TPED events when unlimited resources and time are made available to complete each assigned job[1].

Utility Life Cycle

A key component to implementing a model to track TPED and TPPU activity is to calculate how utility accrues within jobs and decays in the resulting products. There are three major components that interact to account for the life cycle of utility. Utility life cycle is not a new concept but instead is the extension of well-established economic principles with respect to supply and demand that account for how C2 systems dynamically create and expend the utility of data.

The first segment of the utility life cycle involves the accrual of utility in the context of a job moving from start to finish. TPPU accrues partial utility for interim products created by steps in a job while TPED only accounts for job utility after the entire job is completed. The job accrual calculation used in the OPCEN SM model estimates the utility function with a triangular distribution using bounds that are updated to account for the results of the previous step [10].

The second segment of the utility life cycle discusses how the utility of data decays from the moment it is collected. It is described in the context of each consumer's perspective about how the product's utility decays over time. The calculation of product decay is a function of elapsed time instead of number of job steps but otherwise the implementation is the same as for utility accrual.

The separate utility functions for job accrual and product decay are worthwhile calculations in their own right but they have tremendous potential if they can be merged onto a net utility using a common timeline. The combined utility life cycle tracks the data as it moves from original observation, through the processing of a job into a product that is used until it is no longer valid.

Figure 2 illustrates a simplified example of an intended linear accrual function and linear product decay. The result is a net utility curve of how the two processes interact while the job is underway. In this example there is also a noticeable gap between the data is collected and start of the job that reduces the maximum potential utility by a corresponding amount. The producer decides what jobs to pursue using their perception of the decay curve and consumers apply their perspective of utility decay when using the products.

[1] Capt Derek Spencer of the JIIFC Detachment first noted this key aspect of TPPU/TPED interrelationship.

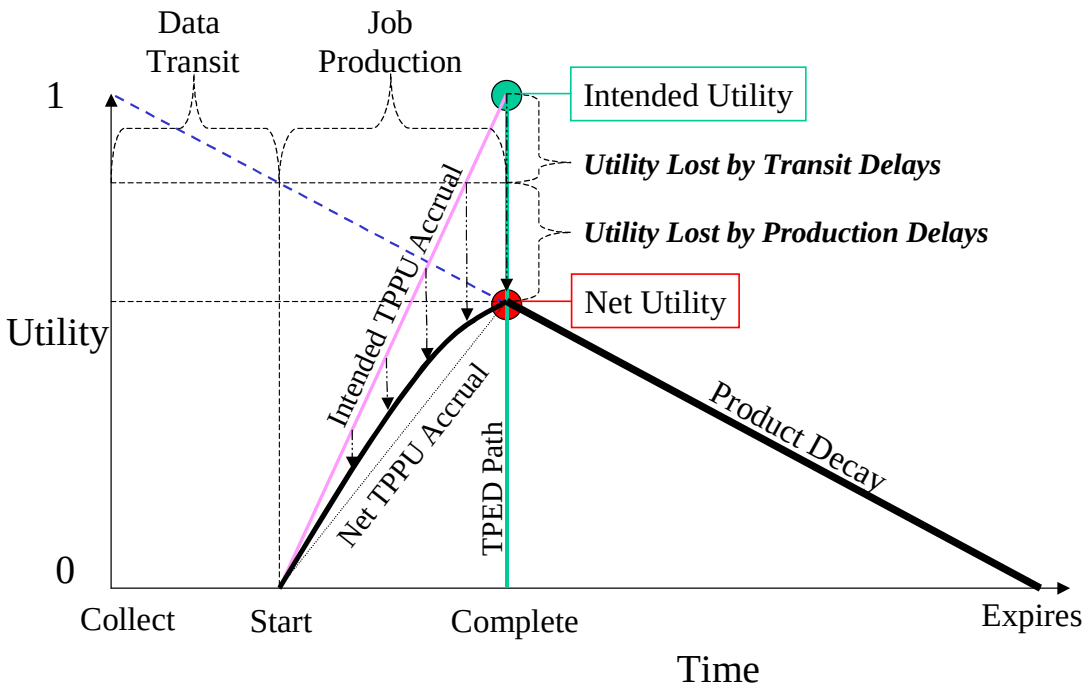

Figure 2. Example of Utility Life Cycle – Using Linear Curves

Modelling TPED and TPPU as a State-Machine

Process to Instantiate a State-Machine

The major shortcoming of the ALIX IISRA behaviour model [4] was that it illustrated the execution of a single path, such as the sequential steps of the UAV flight as a single temporal flow. However, The OPCEN TPPU model involves multiple jobs that can proceed concurrently down parallel paths. The classical model can be made to reflect all possible paths but every activity would require unique names and coding of the logic. Simple examples of TPPU are feasible but the model is not scalable because the number of elements grows exponentially with the number of operator interactions.

Behaviour Modelling Prior to Creating SM Activity Model

The process used to construct a SM activity model actually starts with the same generic process as any standard architecture model of activity along the timeline. These initial models then undergo extensive operator validation to identify the aspects that must be handled by the SM so that it captures the nuances of all the process models correctly. The initial behavioural modelling process described below involves:

a. Threads - Work process of each job by an operator. It articulates specific job activities for each operator as if they are independent of all other processes;
b. Integrated - Common themes between jobs & operators. It is used to calculate the minimum resource demand if operators are fully cross-trained and interchangeable;
c. Allocated - Differentiation of operator skills determines the added cost of skill specialization. It may be feasible to allow for resource offsets when generic work can be done during a time that a specialist would otherwise be idle.

Thread Behaviour

The models used to conduct the visualization need to start with simple behavioural models that describe an operator's workflow for each type of job. These job sequences are referred to as job threads. They are easily documented and visualized using an Enhanced Function Flow Block Diagram (EFFBD). The operator feedback about each job thread helps to uncover new steps and relationships. A very useful by-product of using EFFBD models in CORE is that they can be used to gather data and update the model logic during discussions with the operators.

Integrated Behaviour

Throughout the process, the analyst should be able to identify some common themes that occur in the job threads being modeled. This is expected since operators are trained to perform jobs that may make use of common or compatible skills. The common aspects of the various job threads are channelled into a single aggregated flow with exceptions being handled by logic constructs and separate branches. It assumes a single pool of operators that can perform any activity. This represents the lower bound on the number of required operators before activities are differentiated by specialized skills.

Allocated Behaviour

The reality is that it takes specific analysis skills to complete OPCEN jobs. The resulting allocated behaviour model describes specialized activities that need operators with differentiated skills. It also describes the overhead costs of specialization when the specialist skills are not used.

General Description of a State-Machine

A SM is defined in [11] to be "a modelling concept used by system engineering to describe any device that stores the status of something at a given time. It can then use inputs to change the status and/or cause an action or output to take place for any given change." A more formal description is available in [12].

SM versus Classical Views

The classical behavioural model diagrams display DoDAF views very well as a serial process along a timeline but they are unable to handle concurrent activity. The SM overcomes this shortcoming by shifting the modelling perspective orthogonal to look down the time axis. The SM construct then simulates the decision-making process at a moment in time and uses the results to set conditions for the next time slice. This is like progressing through time by stepping a clock forward and running through the business logic for each time slice. The result is that the SM acts as an analytical 'engine' that is instantiated at runtime to process complex and interactive scenarios described by simpler behaviour models.

The advantage of a SM being orthogonal to the timeline is that it can support multiple threads and multi-instances that execute simultaneously anywhere. The SM structure also makes it easy to identify and articulate workflow rule sets, such as queue logic and resource allocation within tasking processes. This allows the analyst complete control over how jobs are tracked between steps and allocated to operators.

Instantiating a State-Machine in CORE

Instantiating SM simulations using CORE software is a fairly simple and straightforward process. The behaviour model is developed in the usual way and then the logic is incorporated into SM structural components. The Figure 3 'Thread Processing' element contains the applicable behaviour model details and SM logic to support its execution. What makes the SM unique to other CORE models is the imposition of an external 'Clock' to control the simulation's progress so that major activities can be scheduled and executed at any time during the simulation.

Schedule Processing Characteristics

The Schedule Processing element in the top level of the SM diagram is responsible for activating jobs whose event arrival time has been reached. At the beginning of each time step, this element goes through the entire list of jobs that are to be simulated and picks out those scheduled to be queued this time step. Any job already in progress or scheduled for later is ignored.

Thread Processing Characteristics

The original version of the SM was inspired by specific job threads which were abstracted to the point where they could be articulated using the same terminology. More to the point, the jobs could all be described as instantiations of a single common thread. The logic describing how operators would perform the standardized jobs was articulated as the basis for the SM Thread Processing element. The high-level steps within this logic description are illustrated as an EFFBD in Figure 4 and scripts are used to handle the business logic.

Job Queue Processing

When the highest priority job has been selected, the SM determines if a step within the job is, or should be, queued. The exit conditions put a job that is between steps into the next queue and start processing it. The job in a step can also be allowed to proceed uninterrupted or be marked as completed.

Some Advantages of State Machines

A major advantage of the SM approach is that it is a very scalable model. This is because the business logic is defined in detail once and then the resources and activities are driven by data files. Those files are based on n-dimensional dictionaries of global variables that hold attribute values of the element and schedule the major events. The model can be executed for different variations of the workflow process by simply changing the values in the data files.

Some Disadvantages of State Machines

The SM model comes with a price because it relies upon the COREscript to program much of the simulation's activity. The result is a much lower visibility of process steps than would otherwise be available from the behaviour model diagrams. The threads, integrated and allocated models provide the foundation to write the SM logic but the onus is on the analyst to ensure processes are correctly implemented. This can place a major burden on the analyst because COREscript has limited Smalltalk constructs that can be quite cryptic.

Analysts must also be aware that the SM clock in Figure 3 moves forward at a rate independent of the CORE simulator's internal clock. The analyst needs to avoid the temptation to use simulation time to generate jobs and insert them directly into the SM schedule. The preferred method is to generate the list of events by separate temporal simulations (i.e. UAV flight) and then round the time values to the same unit used by the SM.

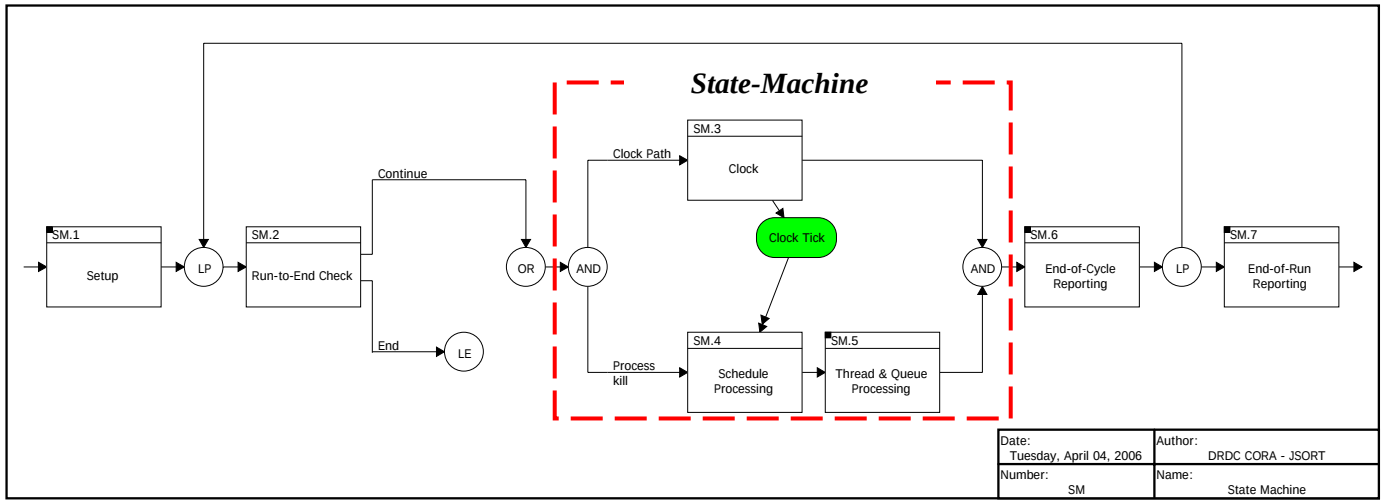

Figure 3. Structure to Support CORE State-Machine and Its Behaviour Model

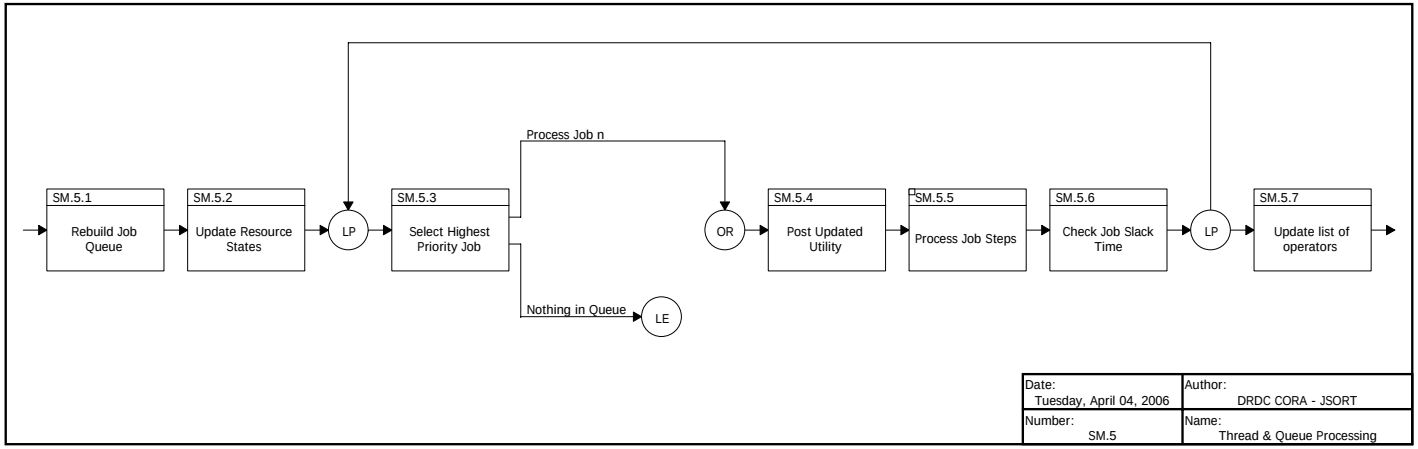

Figure 4. Thread Processing in the State Machine

OPCEN SM Model

The SM logic was tested by simulating the process of multiple operators in an OPCEN performing analysis as described by several job threads. The following sections describe the simulation of OPCEN activity during 24 hours when the focus is the analysis of imagery from a UAV mission.

OPCEN Workflow Characteristics

External Events

The OPCEN SM model is focused on internal OPCEN activity so the UAV mission activity is mimicked beforehand by simulating a temporal flow of external events that eventually activates jobs relevant for the OPCEN to assess. The external events list also contains associated OPCEN jobs related to the UAV mission.

OPCEN Configuration

The crux of the OPCEN activity is to generate products and combine them with other information into aggregated products. The basic composition of the OPCEN watch used in the following example is 5 operators with generic names[2] of Senior Watch Officer (SWO), Imagery Analyst (IA), Operational Information (OI), Content Manager (CM) and Summer Student (SS). Activities covered by the OPCEN watch team have been purged of any specific meaning and the descriptions below use the term "Step" to represent the stages of a job as it progresses from start to finish.

The connectivity assumed by the OPCEN SM model is that the mission delivers a list of events to a central repository that is assumed to exist but is not explicitly modeled here. The repository is where the operators gain access to the data to do their assigned jobs and it is where they store the products from the jobs.

Job Threads

The job threads directly relevant to the operational workload are the starting point for describing the OPCEN internal workflow. The process starts with the analyst documenting the basic characteristics of each type of job handled by each operator. This is the basis for the operator job threads and is described below using TPPU terminology.

Figure 5 shows a thread for the Imagery activity job flow using the EFFBD in CORE. It concisely represents the activity needed to complete the job for that job type. The imagery thread uses the AND construct with two branches; one handles the actual job process while the other monitors the clock time to determine if the available time has been exceeded.

The key feature of the job process is that steps 1 to 6 follow a fairly straightforward serial process. The only exception is that after step 1 the operator must decide if the job is Nothing Significant To Report (NSTR). Jobs designated as NSTR are stopped; otherwise the job is completed using steps 2 to 6. Completed and NSTR jobs both trigger the decision that the job is finished so that the AND branch can finish. If the drop-dead time is exceeded, the "Job Complete or Drop-Dead Time Reached" activity times out, leading to the exit branch where the AND structure is killed.

[2] The remainder of the report uses the short form of the operator assigned position to describe both the operator position and the skill. This is possible because the operator positions are defined by the skills they employ.

The precise work undertaken in each job thread activity is unique and must eventually be assessed. However, at this early stage of describing the OPCEN workflow the analysis only needs to identify the duration of each activity for a particular job. The data is in a table that lists the duration of each job thread activity plus the percentage that will be NSTR.

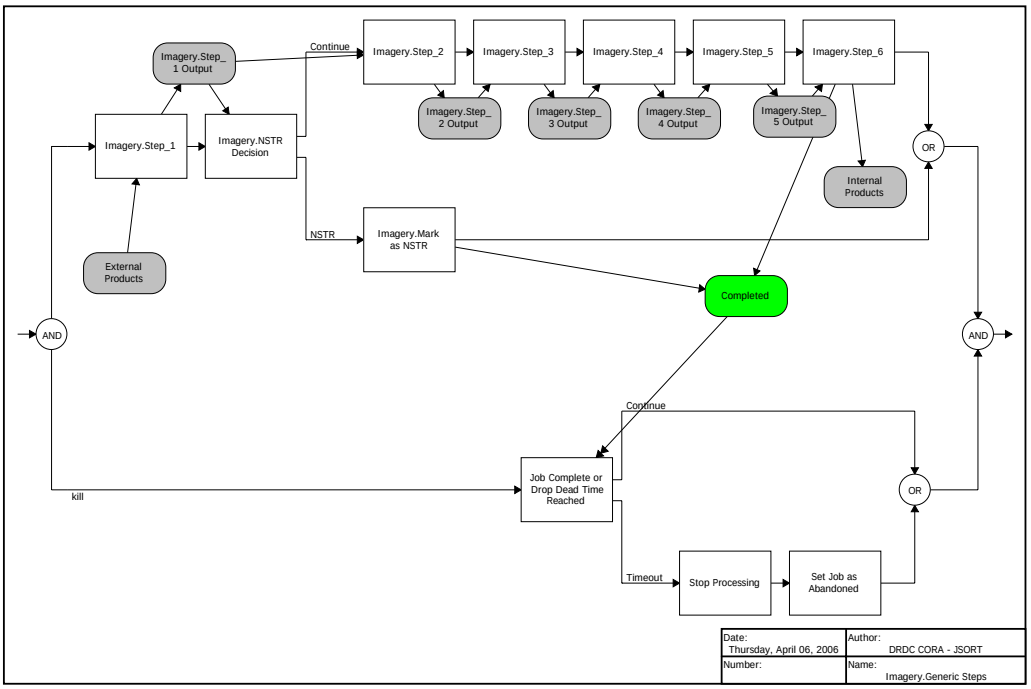

Figure 5. OPCEN Imagery Example of Modelled Thread

Integrated Model

The cumulative effect of combining the four main types of threads (i.e. Imagery, Snapshot of the Common Operating Picture, Situation Report (SITREP) and Assessment) leads to a single integrated model. The aggregation of threads combines common aspects of each thread into a generic “super thread” and then adds in unique aspects of particular threads where appropriate.

In the OPCEN SM model the four major thread types are very similar in structure and process. There is only one significant internal thread exception where the Snapshot thread bypasses step 4. There is another key exception in that the Imagery and Snapshot products are used as input into the SITREP and Assessment threads. External and internal input sources are also differentiated by an OR construct at the start of the job processing branch.

Allocated Model

The final step in describing the OPCEN workflow example dataset is to allocate the behaviour to job skills. Figure 6 shows the result of allocating the thread activity to specific job skills based on the job skills matrix. It is more complex than the integrated model because it must use triggers to coordinate the transfer of activity between skills at the appropriate time. The effect is that the modeling perspective changes from a job centric view to a skill centric view of how jobs move between operators with that skill.

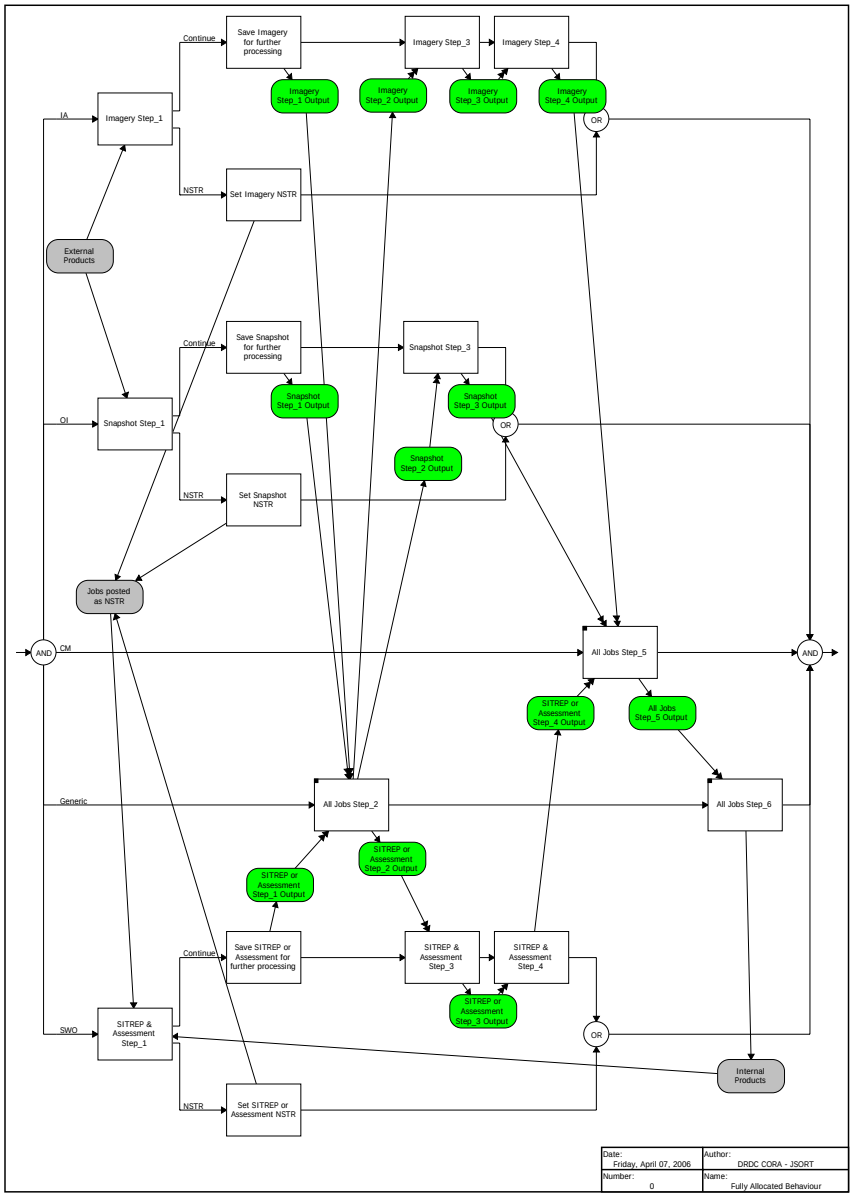

Figure 6. OPCEN Allocated Behaviour Model

OPCEN SM Model Business Rules

<u>Major Model Characteristics</u>

The OPCEN SM model is scalable because it uses coding that can be easily amended and data files that drive the business rules to the maximum extent possible. The OPCEN SM model also uses detailed TPPU business rules create concurrent independent TPPU cycles that are automatically and continuously repeated. The rules are used to assign the jobs to operators and then continuously track their progress in case the jobs are interrupted due to higher priority or reallocation of resources. The jobs are held in a single centralized queue that is continuously monitored and automatically reprioritized each time a job enters or leaves the queue. The utility of the product depends on the stage the job has reached when it is posted.

Business Rules Currently Modelled

Prioritization

Each job has a specified priority so available operators can be assigned to high priority jobs before low priority jobs. Many aspects of a job are specified for a particular job type so a particular job with its unique priority value is treated as a job type with the closest standard priority.

When more than one queued job has the same priority, the SM must decide which should be treated as the next in line. The first tiebreaker rule is that the job with the least slack time remaining is treated as the highest priority. This rule is only applied if jobs have remaining slack time less than an analyst-specified percentage of the remaining time needed to complete the job. The second tiebreaker rule is that the job with the highest utility-to-time ratio takes precedence. The utility-to-time ratio depends on the number of steps ahead that are considered, and this number is specified separately for each job type.

In the case of work that requires a particular skill, the SM will look at each available operator's primary skill, then to his or her next listed skill and so on until it finds someone with the skill necessary. If more than one operator has the required skill in the same order (i.e. primary, secondary, tertiary skill etc.), then the operator who has the highest product of speed and skill (both expressed as percentages of what is considered typical) will be assigned the job.

For unskilled, generic work, the SM looks to each operator in an order specified by the analyst to see if that operator is available to take on the job. Typically, the analyst will specify that the operator with skills in least demand be the first one assigned to generic work.

Job Delivery Deadlines

All jobs have an acceptable delivery time, after which they are no longer relevant and should be abandoned. Prior to the delivery deadline, jobs are either worked on until finished or they remain in a queue until delivery is no longer feasible.

The model employs a mechanism to keep track of the time remaining so the job can stop as soon as there is no possibility of being able to deliver the final product on time. The planned time for new jobs is first calculated and then compared with the time to the delivery deadline to determine the initial slack time. When the jobs execute a step they check the time used with the planned time and then adjust the slack time available for the job. The slack time decreases while a job remains in the queue and jobs are abandoned when slack time becomes negative.

The drop-dead time is the clock time after which the job is no longer relevant. By default, the drop-dead time for a job is calculated as the sum of the slack time, the total time necessary to complete the job, and the time at which the job is queued. The analyst can also specify a drop-dead time for each job, reflecting a situation where a job has a specific deadline that takes precedence over the slack time-derived drop-dead time.

Interrupted Jobs

Each job step can be specified as being interruptible, or not. If a step can be interrupted, the operator working on a job will abandon it under certain conditions. The OPCEN SM model currently requires the operator to abandon Job A in favour of Job B if (1) the current step within Job A is defined as interruptible, (2) every operator capable of picking up the next step within Job B is busy, and (3) either Job B is a higher priority than Job A or another operator who is not busy is capable of picking up Job A where the current operator leaves off.

Product Utility Curves

The job utility accrual through interim products has some uncertainty and is estimated using fixed bounds of a triangular distribution of minimum, most likely and maximum values. A full description of the algorithm is documented in [10].

Switchboard to turn on and off business rules/options

Business rules which can apply to some OPCENs but not others, or which the analyst may wish to bypass for analysis purposes. A switchboard input file is available for the analyst to specify what business rules apply and to specify values that are not specific to job threads.

Interruption Penalties

The SM assumes that if a job is interrupted mid-step (after a post) and a new operator is assigned, they need extra time to familiarize themselves with the job. If no interim post was made, then the new operator has to redo all the work and so there is no time penalty added. If the same operator who had abandoned the job picks it up again it must be done within a certain grace period or the familiarization penalty still applies. The analyst specifies the time limit before an operator must become re-familiarized with his own work and the percentage of time between posts needed to become familiar with previously done work. The utility penalty for interruptions has not yet been articulated but it will likely be a ratio of the utility achieved in that step.

Operators Allowed Partial Availability for Generic Jobs

Operators who are not currently processing any jobs are not necessarily available to do generic work. This is particularly true of operators, such as the SWO, whose jobs rely heavily on maintaining SA. The SM assigns generic work to operators using a data file that specifies the order that work is assigned and the percentage of time each operator is available. The SM uses it to check for idle operators until one is found or the job is left in the queue. The process is repeated at each time step until an operator becomes available.

Business Rules Still Under Development

SITREP and Assessment Consolidation

The first major product consolidation currently modeled is the SITREP. The operator is aware of all previous SITREP and Assessment products generated by their OPCEN because they either created the products or they have read them before going on watch. The basic analysis time contained in the job duration matrix covers the time needed to check through common files. The analysis time is then increased by the amount of time needed to transfer the utility from other products. The effect of products that are NSTR are also deemed to be factored into the aggregated product but this is done without incurring any additional analysis time.

SITREP NSTR is only invoked when there have been no new products created since the previous SITREP (i.e. new products, including NSTR, are those jobs where utility > 0). When SITREP NSTR occurs its utility is set to a level that the analyst believes corresponds to knowing there is no significant activity in the area of interest (i.e., utility between 0 and 1).

Assessments have the same basic characteristics as SITREPs with two additional features. Firstly, SITREPs generated since the previous Assessment are considered to be new products and take time to integrate in the analysis step. Secondly, products already processed as part of a previous SITREP are quickly reviewed at the cost of additional time.

OPCEN SM Model Implementation

OPCEN SM Simulation Process

The SM simulation is actually composed of a three-layer process. It was cobbled together to streamline the simulation process to the maximum extent possible without imposing a structure that would be awkward to modify in the future. The current process has the following steps:

a. Input of Events – Generates the list of activities that initiate the internal activity within the OPCEN. The example mission currently used by the OPCEN model is high-level outcome from a single UAV flight. Any other ongoing battle rhythm activity or specific mission would be equally valid;
b. OPCEN State Machine – Process of the actual simulation activity as described in [13];
c. Analysis of Results – Converts the OPCEN SM output into analyzed results.

OPCEN SM Model Analysis of Baseline Scenario

Simulation Analysis Metrics

The OPCEN SM model's ability to quickly generate detailed results for any attribute being tracked by the model so it can be used to test a variety of metrics candidates. Indeed, once the OPCEN SM produced its first viable results the modelling emphasis shifted to developing the basic analysis capability and testing the metrics. The analysis is actually a sample set of unclassified results.

Job activity metrics were investigated for several basic job characteristics and the following ones were assessed to be suitable static metrics for overall job activity:

a. Completed Jobs – Percentage of jobs that were completed (but not those that abandoned because they ran out of time or arrived after their delivery deadline);
b. Utility – Percentage of the total possible utility achieved from fully and partially completed jobs. This value equals the value of TPED completed jobs while incomplete jobs usually cause the TPPU value to exceed the completion ratio;
c. Slack Time – Average amount of time remaining once the job is stopped (i.e. either completed or abandoned);
d. Time Spent in Queues – Average amount of time each job spends in a queue waiting to be processed. A large dwell time in the queue for a job type indicates that the jobs had to wait for operators with the needed skills;
e. Continuity –Average number of operators needed to complete a job type. Job churn is the difference between the average and minimum number of operators to complete a job; and
f. Disruption – Number of interruptions to a job so operator could be shifted to another job (i.e., usually higher priority but lower priority allowed if suitable skill is idle). Job volatility caused by interruptions should be avoided because it impacts on both the time and utility of the interrupted job and the operator's ability to concentrate.

Caveat about Analysis Results

The analysis that follows lists results as absolute quantities but they only illustrate how the analysis works and what kind of results it can produce. The data was selected to produce representative results but they do not describe actual capabilities. Multiple runs with small adjustments allow for valid comparisons of business rules in qualitative terms but the quantitative differences are only indicative at this point in time.

Comparison of State Machine Planning Assumptions

The initial analysis of the example OPCEN workflow characteristics started with a review of a range of planning parameters using two related candidate measures of effectiveness, namely the percentages of completed jobs and utility. The preparation involved modifying the data for what is being changed, running the OPCEN SM model and then tabulating the high level statistics into a worksheet.

Figure 7 lists the results of four analysis themes covering 16 cases. The four themes were comparisons of (1) candidate baselines, (2) adding operators in the single skill baseline, (3) adding operators in the multiple skills baseline and (4) impact of operator ability to look ahead and their speed. The completion and utility quantities are shown for comparison purposes only.

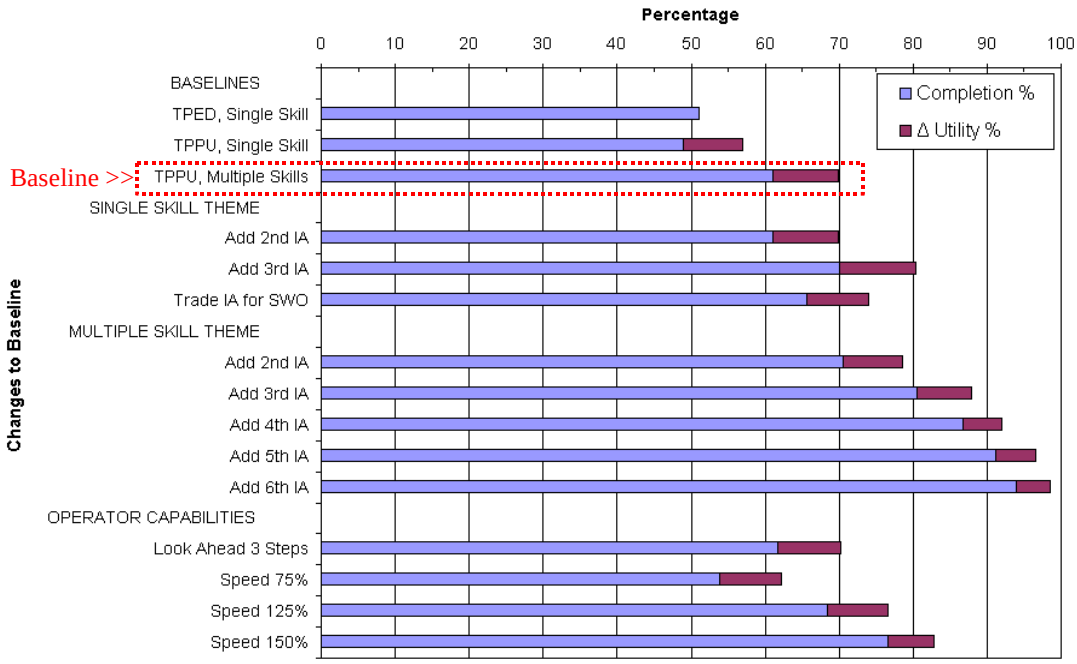

Figure 7. Comparison of State Machine Planning Assumptions

Three candidate baselines considered were TPED, TPPU with a single skill and TPPU with multiple skills. As expected, TPPU outperformed TPED in terms of utility even though there were fewer completed jobs. Multiple skills TPPU was a even a better option because operators it allows operators to use past training as well as their position's assigned skill. In this example the multiple skill TPPU allowed the SM to automatically allocate jobs to otherwise idle operators and this had a cumulative effect equivalent to adding a 2nd IA. The TPPU with multiple skills was selected as the baseline for the subsequent analysis and a later section further discusses the impact of multiple skills on operator utilization and SA.

The theme of adding operators in the single skill baseline was also assessed. The IA was obviously the choke point so a 2nd IA and 3rd IA were added and then the 3rd IA was traded for a SWO. The 2nd and 3rd IA each significantly increased completion and utility while the new SWO halved the gain compared to the 3rd IA level. This suggests the 3rd IA remains the chokepoint in the OPCEN process.

The theme of adding operators in the multiple skills baseline was done as five cases involving two to six IAs. Each additional IA increased the completion and utility rates. By the 6th IA the utility was approaching 100% where all jobs requiring an IA were either completed or impossible due to the drop-dead time being too close to the job arrival time. The marginal utility of each additional IA decreased steadily as expected.

The impact of operator capability was considered for multiple skills. The first case of looking ahead three steps to assess utility gain had a negligible effect. Modifying operator speed by decreasing it to 75%, then increasing it to 125% and 150% produced very large changes. This indicates that job durations must be well defined before the analysis can report valid absolute values. It also suggests that operator training must either be standardized or each operator's speed will have to be evaluated and treated as distinct planning factors.

Timeline Metrics

The metrics mentioned above all assume that activity is uniform at all points along the timeline. This static viewpoint is useful for noting high-level effects but it fails to account for variability in the rate of job activity along the timeline. In the example being assessed here the UAV mission has several different rates of activity during distinct periods along the simulated timeline. This time dependency implies that a far more useful diagnostic tool would be to show job activity along the timeline. The following graphs illustrate three distinct timeline perspectives:

a. Job Duration – Displays duration of each job showing when they were active;
b. Operator Utilization – Shows the baseline case of five operators with multiple skills. The top is a matrix of the proportion of time each skill is used and the remainder available for SA. The bottom is stacked bar chart of operator utilization over time, and
c. Cumulative Utility – Shows how cumulative utility of job accrual, product decay and consolidation over time.

Timeline of All Jobs

Figure 8 shows the duration of all 180 simulated jobs in the order and time when they are placed in the queue to be processed. The UAV mission is the dominant OPCEN activity and the vertical dashed green lines show the main transition points where the job activity shifted. The vertical slice corresponds to the number of concurrent jobs that are active at that time. This activity display is a very useful visualization of the active jobs at the OPCEN.

The simulation generated a subset of 70 jobs that were abandoned after they consumed all their available slack time. The dominant feature of these abandoned jobs was that they all occurred during the Vessel Of Interest (VOI) incident during high rates of job activity. The net effect of the 110 completed jobs produced a linear slope throughout the UAV mission with only a negligible increase in the rate of completed jobs during the VOI incident. This indicates that the OPCEN was operating at essentially full capacity and the queues were saturated without the increased VOI activity. The VOI workload caused lower priority jobs to be abandoned but some high priority jobs also needed to be sacrificed due to a lack of operators with the correct skills.

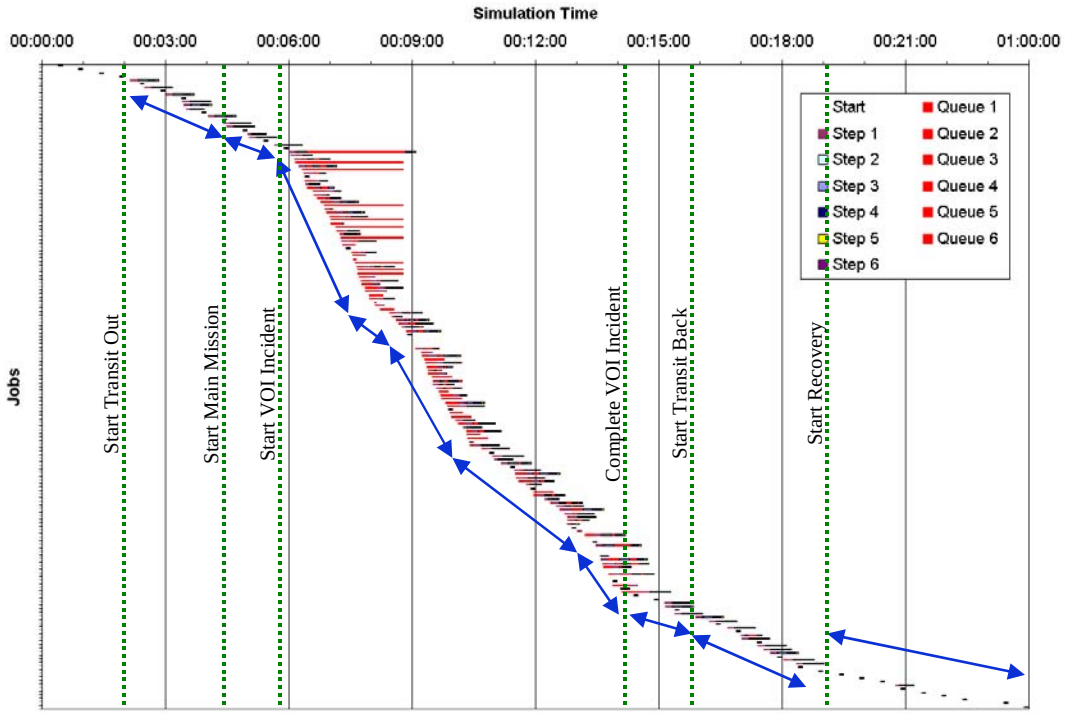

Figure 8. Timeline of All Jobs (Operators have Multiple Skills)

Personnel Utilization (5 Operators with Actual Skills)

Figure 9 shows the baseline case of five operators using all their skills without being limited to the position to which they are assigned. The overall operator utilization is 42% and is equivalent to adding a 2nd IA compared to the case where operators can only use their assigned skills. In this case the effect of a 2nd IA comes from the CM using his or her IA skills to do some of the Imagery workload with the assigned IA. Both operators are kept extremely busy with the IA position being utilized 65.3% of the period (92.2% during the flight) and the CM utilizing IA skills 55.9 % of the time (78.9% during the flight). The IA skill utilization remains very high despite the CM sharing the workload and this indicates there remains plenty of justification for additional IA positions.

The OI (who has CM as a secondary skill) covers for the CM position up to the point of doing almost twice as much CM work as the assigned CM. The model does preferentially assign CM work to the CM, but the CM is usually busy working on Imagery type jobs so the model selects the idle OI as the next viable alternative. The SS and SWO positions work at essentially a steady albeit periodic rate while the IA's SWO skill and SWO's OI skill are used only incidentally.

The SWO has 22.2% overall utilization but is used intensely during the VOI segment of the flight when the SWO skill is utilized 66% of time over 7.5 hours between 00:06:30 and 00:14:00. This heavy workload during the VOI incident compromises the SWO's ability to maintain adequate SA. This is an important consideration because concentrations of jobs can lead to loss of SA continuity. This is far more debilitating than if there had been a stable rate of utilization, even if the average of that steady rate had been somewhat higher. Moving some of SWO job steps to other skills could reduce the burden on the SWO and increase SA.

The interesting point about the above resource reallocation was that the OPCEN SM automatically rebalanced the workload as a direct consequence of the business rules. The only change the analyst made was to enter the data for each operator's applicable 2nd and 3rd skills.

Skill Used >>	SA	SWO	IA	OI	CM	G
Average	**58.0**	**4.5**	**24.2**	**1.8**	**4.7**	**6.7**
SWO	**77.8**	21.5		0.6		0.1
IA	**33.7**	1.0	65.3			
OI	**70.8**			8.4	15.0	5.8
CM	**35.4**		55.9		8.7	
SS	**72.5**					27.5

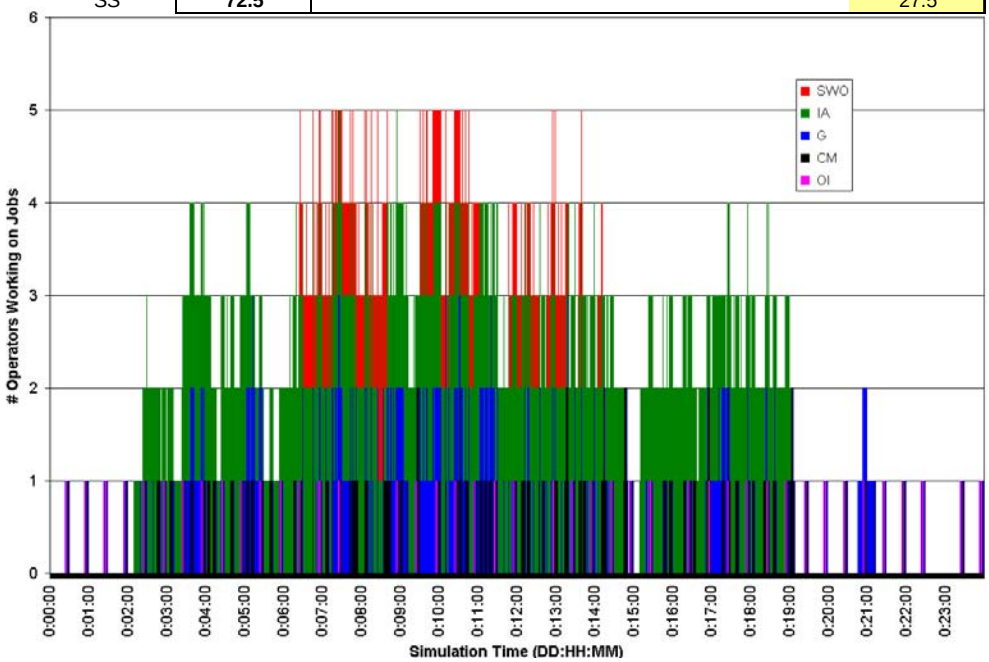

Figure 9. Personnel Utilization (5 Operators With Multiple Skills)

Cumulative Utility

Figure 10 compares the cumulative utility of the concurrent jobs, products and consolidated products over the timeline. The display clearly shows how job utility grows as the work progresses on "Active Jobs" and then decreases as the "Active Products" (i.e. sum of jobs and finished products) decay over time.

The "Consolidated Products" tracks the utility of the products remaining after individual products are factored into the SITREP and assessment products at the Level 300 and Level 100 consolidation points along the timeline. It represents a lower bound that assumes the utility of individual products disappear during the aggregation process.

The "Utility Transferred" line displays a nominal upper bound if the individual product utility is fully preserved by adding the individual utilities to the aggregated products. In this example the aggregated products tend to hold their initial value longer, and decay slower, than individual products. Consequently, the transfer accumulates utility faster, and keeps it longer, than when the products are kept separate.

The analysis also created figures to illustrate the effect of concurrent activity and level of utility entropy along the timeline but they are not presented here.

The above analysis of this example version of the OPCEN SM model illustrates the rich variety of metrics that can be used to help assess the impact of a wide range of planning parameters.

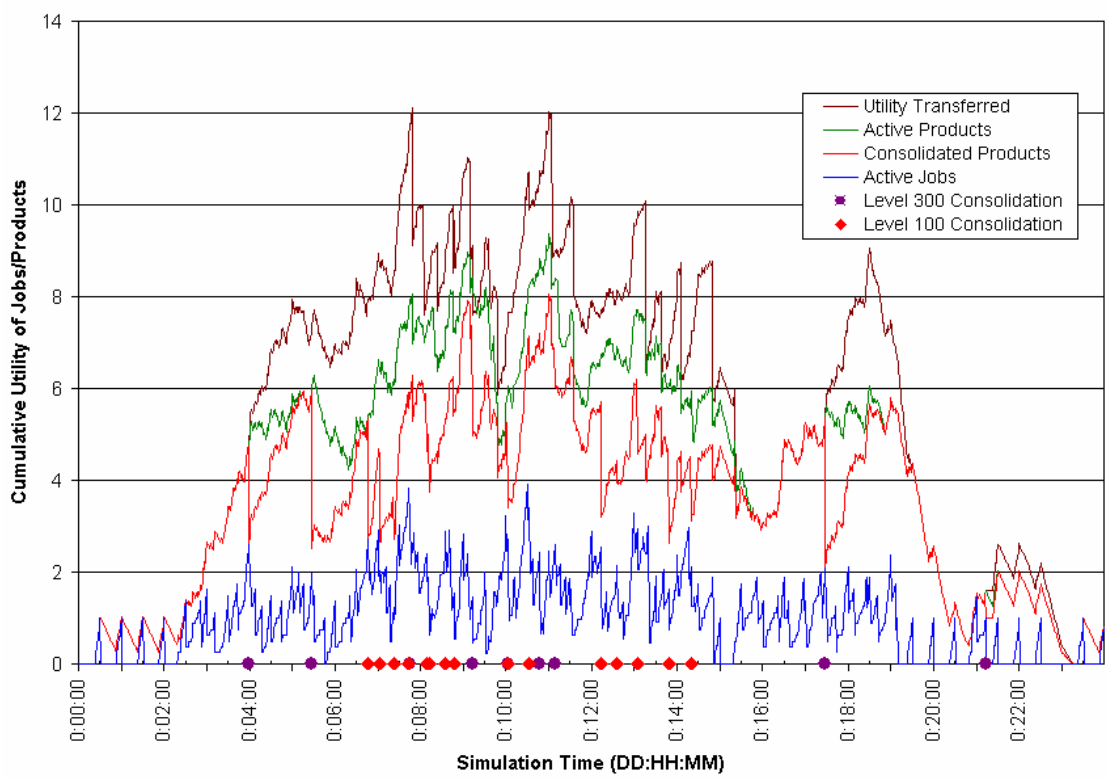

Figure 10. Cumulative Utility

State Machine of Federated Nodes (SMOFN) Model

SMOFN Intentions

The challenges of implementing SM simulations of architectures are examined in the context of the current effort to step beyond the detailed modeling of a single OPCEN. The State Machine of Federated Nodes (SMOFN), under development in DRDC CORA, will simulate workflows between OPCENS within a net-enabled C4ISR architecture. The SMOFN objective is to model the impact on SA due to the interaction of concurrent jobs that are shared between multiple OPCENs. It will also investigate methods to analyze and visualize the results of these simulated architectures. The goal is to develop an analysis tool to faithfully model the major aspects of a major integrated C4ISR architecture.

The first phase has already worked out how to extend the single OPCEN business rules into the SMOFN setting. The next phase will work out the detailed interaction between net-enabled federated nodes as well as the data repository. The process will consolidate the OPCEN SM model into a single standardized template with data files being used to instantiate each federated node. This approach will allow the SMOFN model to customize each OPCEN independently using the switchboard and data files.

Conclusions

OPCEN SM Model Design & Limits

The OPCEN SM model described here is designed to account for the work that can be done on a set of jobs. The focus is to account for job status and operator utilization. The specific portions that it tracks are where the jobs reside at any given moment as they flow through the OPCEN and the operators assigned to each job. The simulation provides a robust framework to compare how a wide range of business rules interact as the operators work on jobs to create products.

There are limits to what the SM model can explain. The most important is that it only reports on the consequences of a particular workflow pattern and does not attempt to explain why that pattern was considered. The other major limitation is that the SM only calculates the duration of the activity. It does not attempt to describe how the operator goes about completing the assigned work within each step nor the cognitive processes that are used.

The following conclusions are made about the viability of the OPCEN SM model:

a. Modelling of TPPU is viable using the state machine but a full range of business rules must be modelled in order to make its logic both comprehensive and cohesive;
b. It achieved its stated aim to successfully simulate concurrent activity;
c. The job activity is logical and becomes self-synchronizing when business rules allow intelligent interruptions to dynamically reprioritize the active jobs;
d. It is a sophisticated description of an executable architecture that faithfully simulates the concurrent operations that occur in net enabled operations;
e. The model achieved its design intentions to explicitly calculate the chain of job activity and operator utilization to a high degree of fidelity;
f. The utility life cycle concept for tracking the utility of consolidated products shows promise and should be implemented;
g. The structure of the model allows easy amending of the activity logic;
h. The use of data-defined job types and operator skills allows it to quickly adapt to new modeling situations and makes it very scalable; and
i. The existing metrics and tools are already capable of supporting a thorough analysis of the modelled characteristics.

Recommendations

The following recommendations are made about the OPCEN SM model:

a. Development of the SM models to simulate architectures should be continued;
b. Operational staffs should allocate personnel to populate the model with new data sets that will allow it to be validated using a range of operational scenarios;
c. The SMOFN model described above should be pursued as quickly as possible.

References

1. Canadian Forces C4ISR Command Guidance and Campaign Plan, 2 December 2003 http://dcds.mil.ca/dgjfd/djfc/C4ISR/documents/2150610547PM20031202%20Command%20Guidance%20&%20Campaign%20Plan%20-%20main.doc

2. OSD Memorandum by John P. Stenbit (Directing Use of DoDAF Within DOD), 9 Feb 04, http://www.defenselink.mil/nii/doc/DoDAF_v1_Memo.pdf, Accessed 1 Nov 04, (Composed of two volumes and a deskbook linked below):

   - DoDAF Version 1.0 - Volume I (.pdf), accessed 10 April 2006, http://www.defenselink.mil/nii/doc/DoDAF_v1_Volume_I.pdf
     Provides definitions, guidelines, background material.

   - DoDAF Version 1.0 - Volume II (.pdf), accessed 10 April 2006, http://www.defenselink.mil/nii/doc/DoDAF_v1_Volume_II.pdf
     Describes each architecture product.

   - DoDAF Version 1.0 - Deskbook (.pdf), accessed 10 April 2006, http://www.defenselink.mil/nii/doc/DoDAF_v1_Deskbook.pdf
     Provides supplementary "how to" information relating to building architectures.

3. Vitech Corporation Website, http://www.vtcorp.com, accessed 10 April 2006.

4. R.W. Funk, L. Arseneau and R. Youssef, *The Atlantic Littoral ISR Experiment: An Initial Model of the Integrated Intelligence, Surveillance and Reconnaissance Architecture*, ORD Technical Memorandum TM 2005-03, February 2005.

5. R.L. Sorensen and S.T. Booth, Gaining Insight into System Operational and Support Concepts through Modelling, Vitech Corporation presentation to National Defence Industry Association (NDIA), 2004 Systems Engineering Conference, Dallas, 25-28 October 2004.

6. R.W. Funk and R.L. Sorensen, *State Machine Modeling of TPED and TPPU*, Presented to the National Defense Industry Association 8th Systems Engineering Conference, San Diego, 27 October 2005, 63 slides.

7. R.W. Funk, M.G. Ball and R.L. Sorensen, *Building Executable Architectures to Simulate Concurrent Activities of Net Enabled Operations - Case of a Single Operations Centre*, DRDC Technical Report TR 2006-xx, Submitted for peer review 13 April 2006.

8. Global Information Grid Website, TPPU, the New Paradigm, http://ges.dod.mil/about/tppu.htm, accessed 14 December 2004.

9. SYS 501: Developing Executable Architectures Using the DoD Architecture Framework (DoDAF), Systems and Proposal Engineering Company, Course CD, 17-18 June 2004.

10. S.M. Gauthier and R.W. Funk, *Estimating Partial Utility of Interim Products*, DRDC CORA Technical Note TN 2006-xx, April 2006.

11. State Machine Definition, accessed 10 April 2006, http://searchsmallbizit.techtarget.com/sDefinition/0,,sid44_gci214244,00.html

12. B. Selic, G. Gullekson, P.T. Ward *Real-time Object-oriented Modelling*, John Wiley & Sons, 1994, ISBN: 0471599174.

13. R. Sorensen, Operations Center State Machine Model User's Manual, 31 March 2006.

Glossary

ALIX	Atlantic Littoral ISR Experiment (August 2004)
C2	Command and Control
C4ISR	Command, Control, Computers, Communications Intelligence, Surveillance and Reconnaissance
CapDEM	Collaborative Capability Demonstration and Engineering Management
CF	Canadian Forces
CM	Content Manager
CORA	Centre for Operational Research and Analysis
CORE	CORE Application from Vitech Corporation
COREscript	Smalltalk-based scripting language used in CORE
COREsim	CORE Application with Simulation Package
DoDAF	Department of Defense Architecture Framework
DRDC	Defence Research and Development Canada
EFFBD	Enhanced Function Flow Block Diagram
IA	Imagery Analyst
IISRA	Integrated Intelligence, Surveillance and Reconnaissance Architecture
ISR	Intelligence, Surveillance and Reconnaissance
JSORT	Joint Staff Operational Research Team
M&S	Modelling and Simulation
MBSE	Model Based System Engineering
NSTR	Nothing Significant To Report
OI	Operational Information Coordinator
OPCEN	Operations Centre
SA	Situational Awareness
SITREP	Situation Report
SM	State-Machine
SMOFN	State-Machine of Federated Nodes
SS	Summer Student
SWO	Senior Watch Officer
TD	Technology Demonstration
TPED	Task, Process, Exploit, Disseminate
TPPU	Task, Post, Process, Use
UAV	Uninhabited Air Vehicle
VOI	Vessel of Interest
NOTE	*Acronyms used in the diagrams are NOT in the glossary because specific model content is incidental to the issues discussed in this report.*