12th ICCRTS

"Adapting C2 to the 21st Century"

Paper: **I-032**

Title: **Executable Architecture of Net Enabled Operations: State Machine of Federated Nodes**

Topics: Modeling and Simulation
C2 Metrics and Assessment
Network-Centric Experimentation and Applications

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

Ronald Funk
Joint Staff Operational Research Team
Centre for Operational Research and Analysis
Ottawa, Ontario, Canada

Richard Sorensen
Principal Systems Engineer
Vitech Corporation
Vienna, Virginia, USA

Point of Contact:

Mark Ball
Centre for Operational Research and Analysis
National Defence Headquarters
101 Colonel By Drive
K1A 0K2
Office: (613) 992-4539
Fax: (613) 992-3342
Email: ball.mg@forces.gc.ca

Executable Architecture of Net Enabled Operations: State Machine of Federated Nodes

By

Mark Ball, Ronald Funk and Richard Sorensen

The Defence Research and Development Canada (DRDC) Centre for Operational Research and Analysis (CORA) is developing capability-engineering analysis tools to support the building, demonstration, and analysis of executable architectures. Our paper to 11th ICCTS [1] described how to model workflows within an Operations Centre (OPCEN) employing a Net-Centric architecture. It used a State Machine (SM) model to simulate how multiple jobs can proceed in parallel when operators use Task, Post, Process, Use (TPPU) cycle to organize their work.

This paper extends the OPCEN SM model to track the interaction of work between OPCENs. The State Machine of Federated Nodes (SMOFN) model is organized around networked nodes that produce and consume products held in a virtual Repository. The data-driven simulation uses files to build customized job workflows and configure any combination of nodes without affecting the business logic. SMOFN also accounts for the following overhead activities:

(1) Tracking consumer perception of product utility as it accrues and decays;
(2) Consolidation of products into higher-level aggregated products; and
(3) Triggering new jobs where needed whenever relevant products become available.

Customization of SMOFN is underway to account for the data and product flows between OPCENs in new Canadian Forces Command structure.

- [1] Funk, R.W., M.G. Ball, and R. Sorensen, "Building Executable Architectures of Net Enabled Operations Using State Machines to Simulate Concurrent Activities", presented to 11th ICCRTS, Cambridge, UK, September 28, 2006.

Outline for I-032

Introduction

Background

Previous work on executable architectures includes single OPCEN State Machine. That model has been extended to account for interactions between several OPCENs.

Brief primer on TPED vs TPPU

Need for State Machine comes from TPPU logic. TPED handles jobs in serial fashion; can be simulated in flowchart diagrams (ie. CORE) by putting process in a loop and repeating for each job. TPPU allows multiple jobs to occur simultaneously by interrupting low priorities in favour of high priorities. State Machine records state of each job so logic to assign operators to high priorities can be executed for a moment in time.

Conceptual Basis for SMOFN

OV-1 diagram illustrating relationships between Producers, Consumers, Discoverers, External Sources, and Repository. Used as the basis for the logic controlling interactions between nodes in the SMOFN.

Producer-Repository-Consumer model logic

Simple diagram to illustrate how producers can post to repository and consumers pull from repository.

Operational Decision Making Logic

Extensions beyond OPCEN SM capabilities

SMOFN allows user to specify job threads by identifying any number of steps within a job by name (OPCEN SM assumed 10 steps, could handle fewer by assigning zero time to some steps)

Producer logic

a. Was the focus of the OPCEN SM.
b. Each time step checks for new jobs.
c. Logic to break ties between equal priorities
d. Operators with multiple skills
e. Logical interruptions
f. Though Utility is in the eye of the Consumer, Producer has own idea of utility and jobs decaying too fast will be abandoned.
g. Goes through jobs in order of priority, assigning available operators to tasks.
h. Utility is updated whenever a post is made.
i. Jobs abandoned if slack time (difference between deadline and estimated completion time) becomes negative.

Consumer logic

For each product received, there is a chance of generating a question which will trigger a new discovery thread

Discovery logic

a. Answers questions for the consumer

b. Questions are answered by jobs which are handled similar to producer threads (most steps are actually handled by the same logic scripts).

c. Main differences from Producer: Does not consider utility, no chance of Nothing Significant to Report, possibility of tasking External Sources to collect more information

External Sources logic

Receive Requests for Information from Discovery threads. The amounts of information that can be found, and how long it takes to find, vary. New raw data triggers analysis jobs by producer.

Repository logic

Main job is to transfer

a. Newly arrived data from External Sources to Producers

b. Products from Producers to Consumers

c. Questions from Consumers to Discoverers

d. RFIs from Discoverers to External Sources

Implementation in COREsim

SMOFN top level

Top level flowchart diagram controlling SMOFN execution. Colour-coded to highlight elements controlling logic behind each of the nodes illustrated in OV-1. Unlike most flowchart diagrams in CORE, time is not left to right. Time is actually stopped and left to right process is decision making process at any instant in time. Time is then incremented and process is repeated.

Overview of the logic which executes for each time step, tracking decisions made at each node.

Input Data Files

OPCEN Input data

a. Switchboard to personalize business rules
b. List of events for each OPCEN, based on operating procedures
c. General details of each job thread (percent change of Nothing Significant to Report, nominal amount of slack time, previous products used as input for aggregation)
d. Step-specific details of each job thread (duration, skill required, number of posts of information, interruptibility, utility added, and file size of product).
e. List of operators in each OPCEN; including their trained skills, speed, and quality of work.
f. Decay of each type of product over time, as determined by each OPCEN.

Repository Input Data

a. Schedule of data arrival – data sent to specified OPCEN and used as trigger and input for new job.
b. Schedule of arrival of new products created by OPCENs outside the model (new products are used by modelled consumer OPCENs).
c. Delivery matrix specifying probability of each job type produced by each OPCEN being sent to any other given OPCEN.
d. Bandwidth between each OPCEN and Repository

Way Ahead

Work in Progress

Goal is to accurately model data flow between OPCENs in the Canadian Forces. Need to specify operators, job threads, lines of communication.