DEFENCE R&D DÉFENSE

DEFENCE R&D DÉFENSE

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

Mr. Ron Funk (DRDC CORA) - Presenter

Mr. Mark Ball (DRDC CORA)

Mr. Rick Sorensen (Vitech Corp.)

11th ICCRTS at Cambridge, UK

28 September 2006

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

Mr. Ron Funk (DRDC CORA) - Presenter

Mr. Mark Ball (DRDC CORA)

Mr. Rick Sorensen (Vitech Corp.)

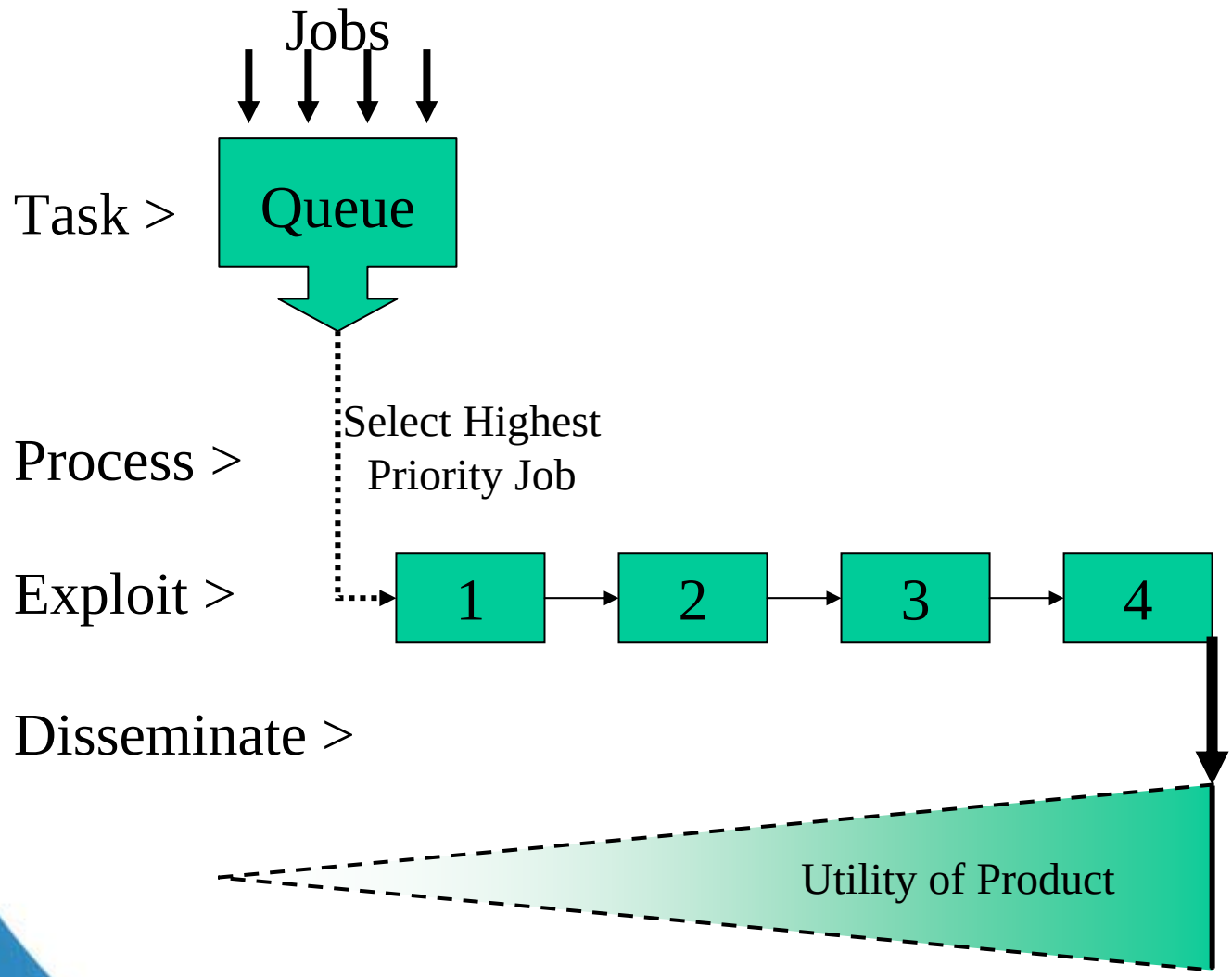
11th ICCRTS at Cambridge, UK

28 September 2006

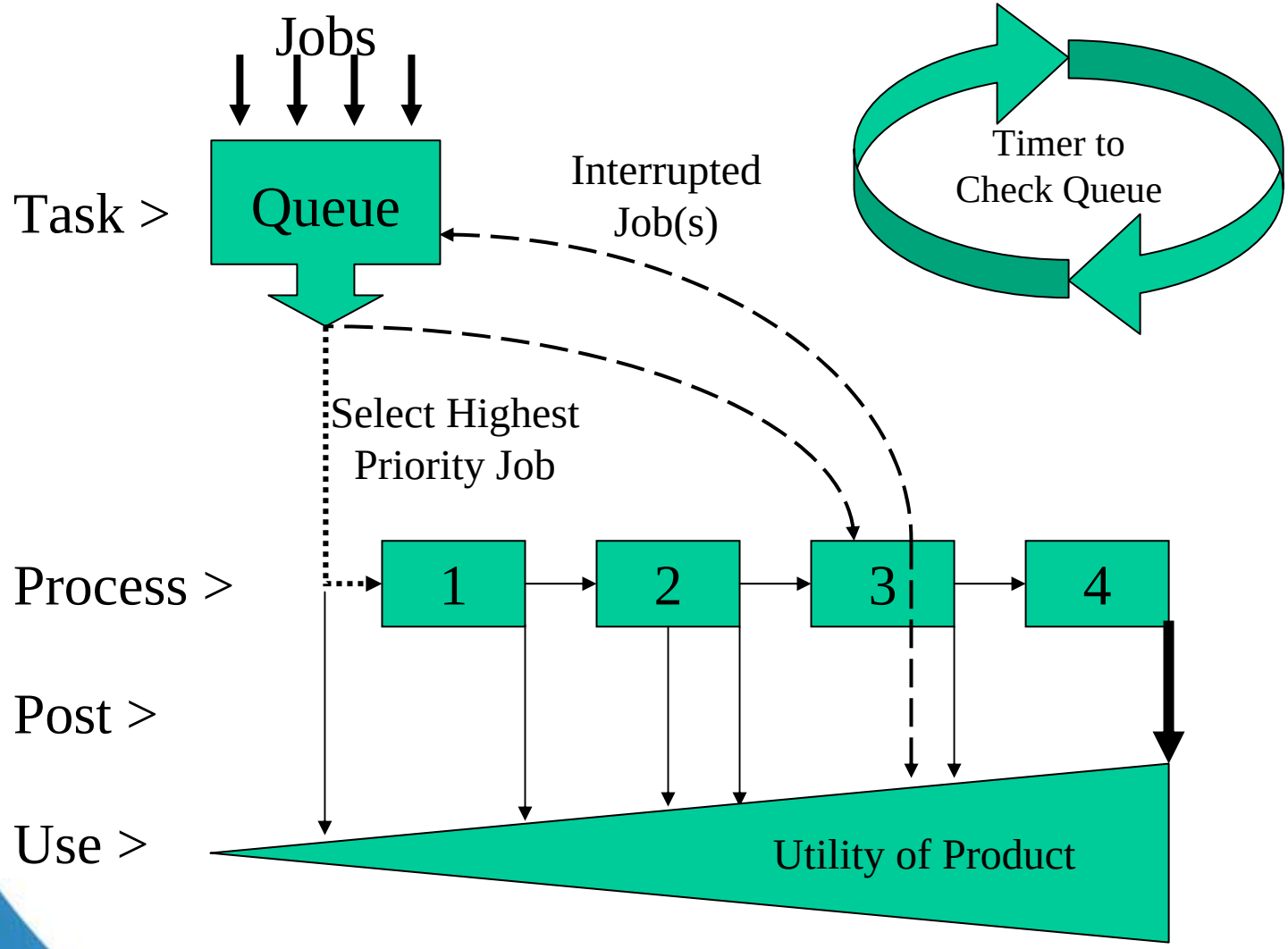
Outline of Presentation

- Background:
  - TPED vs TPPU Activity Modeling
- Process:
  - Modelling Steps & State Machine Concepts
  - Business Rules modelled
- Example Analysis (based on fictitious OPCEN)
  - Static Metrics
  - Timeline Metrics
- Way Ahead
  - Extending model to handle interaction of Federated Nodes
- Conclusions

How Task, Process, Exploit, Disseminate (TPED) Handles Jobs



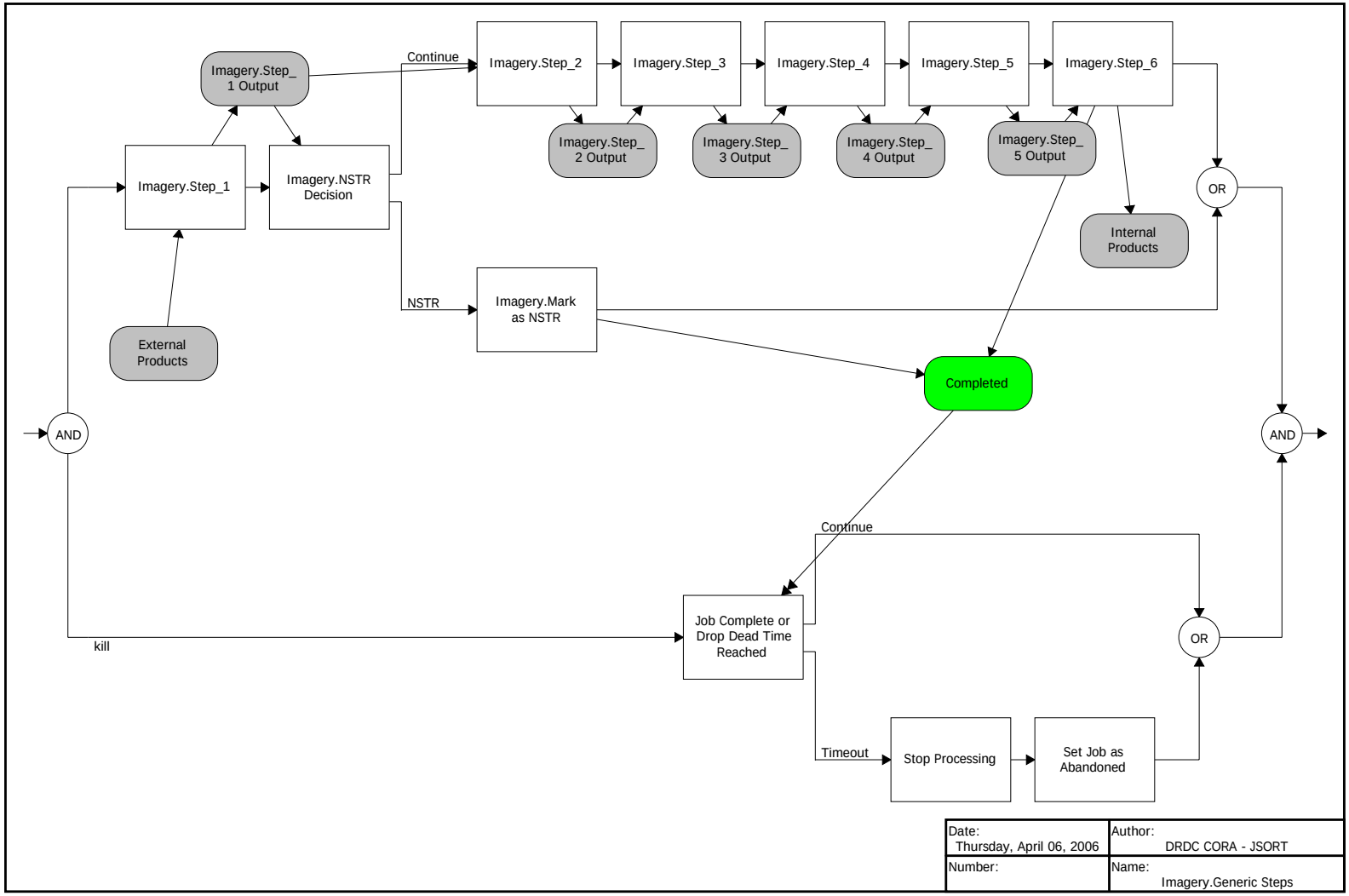
How Task, Post, Process, Use (TPPU) Handles Jobs



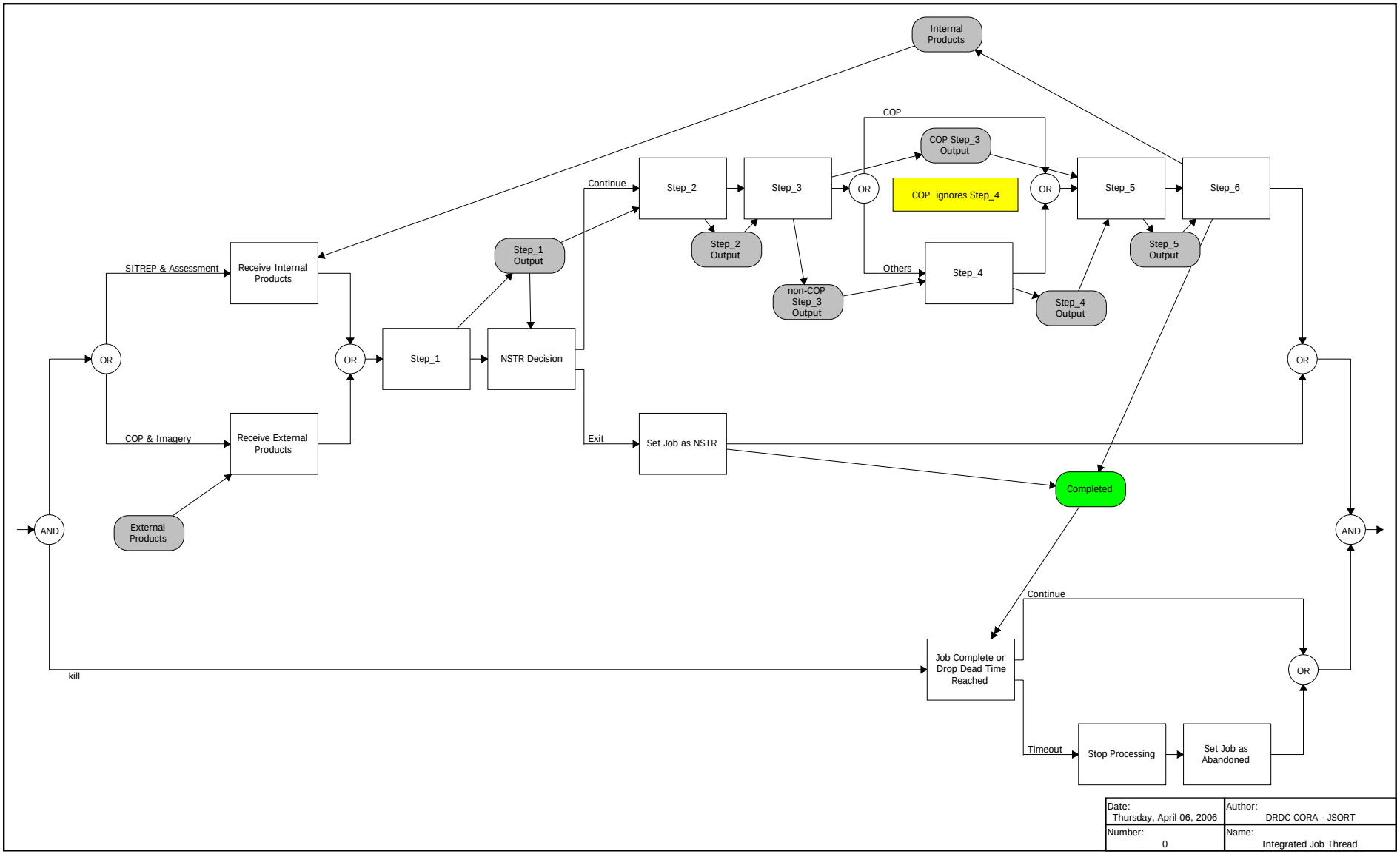
Process to Efficiently Build Behaviour Models

1. **Model Threads**: Work process of each job by an operator
   - Articulates specific job activities for each operator
2. **Integrated Model**: Combine common themes of threads
   - Calculates minimum resource demand
3. **Allocated Model**: Differentiation by operator skills
   - Added cost of specialization
   - Determine any offsets when generic work is done during idle time
   - Shifts focus from job activity to operator utilization

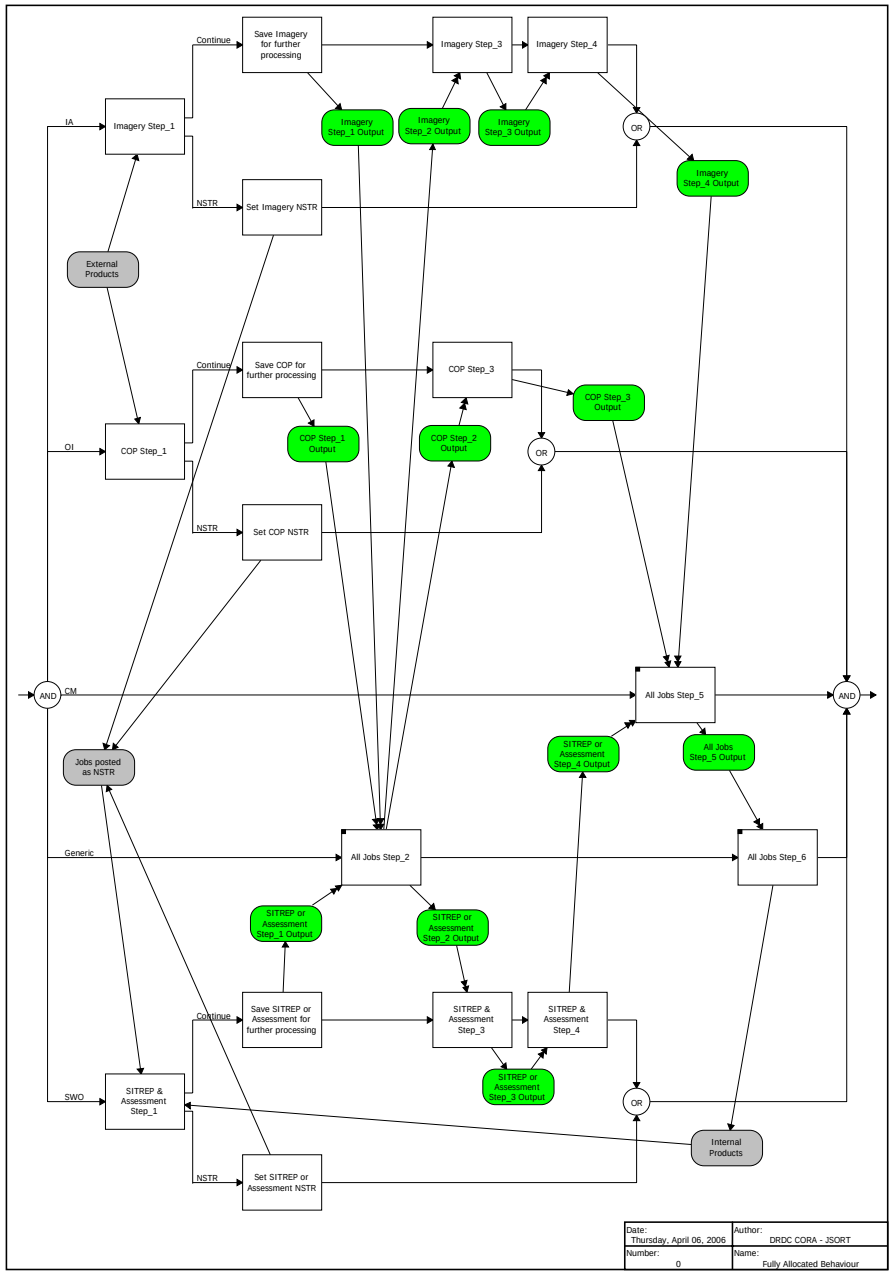
Example Job Thread



Example of Integrated Model



Fully Allocated Behaviour Model

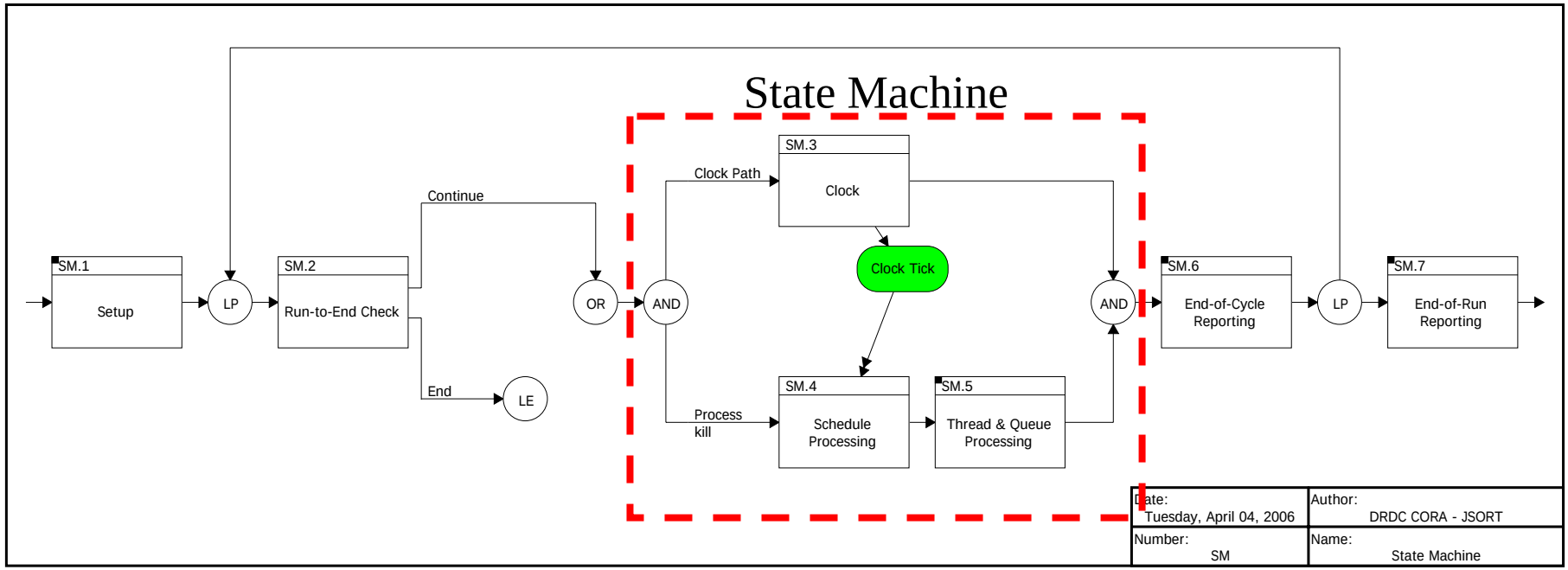


State Machine Versus Classical Views

- Classical behavioural diagrams display single-threaded, single-instance temporal domains (i.e. along timeline)
- State machine is multi-threaded, multi-instance of temporal domain (i.e. **orthogonal** to timeline)
- SM is really just an 'engine' for processing scenarios
  - Logic for workflow, resources, and queues captured as rule sets within tasking process
  - Model instantiated at runtime
  - Jobs allocated to operators and status is tracked
  - Job and operator states are recorded at end of each time step
  - Previous state used as start basis for the next time step
- Caution: SM logic is not as visible to users as the classical behaviour diagrams, such as EFFBD*

*EFFBD: Enhanced Functional Flow Block Diagram

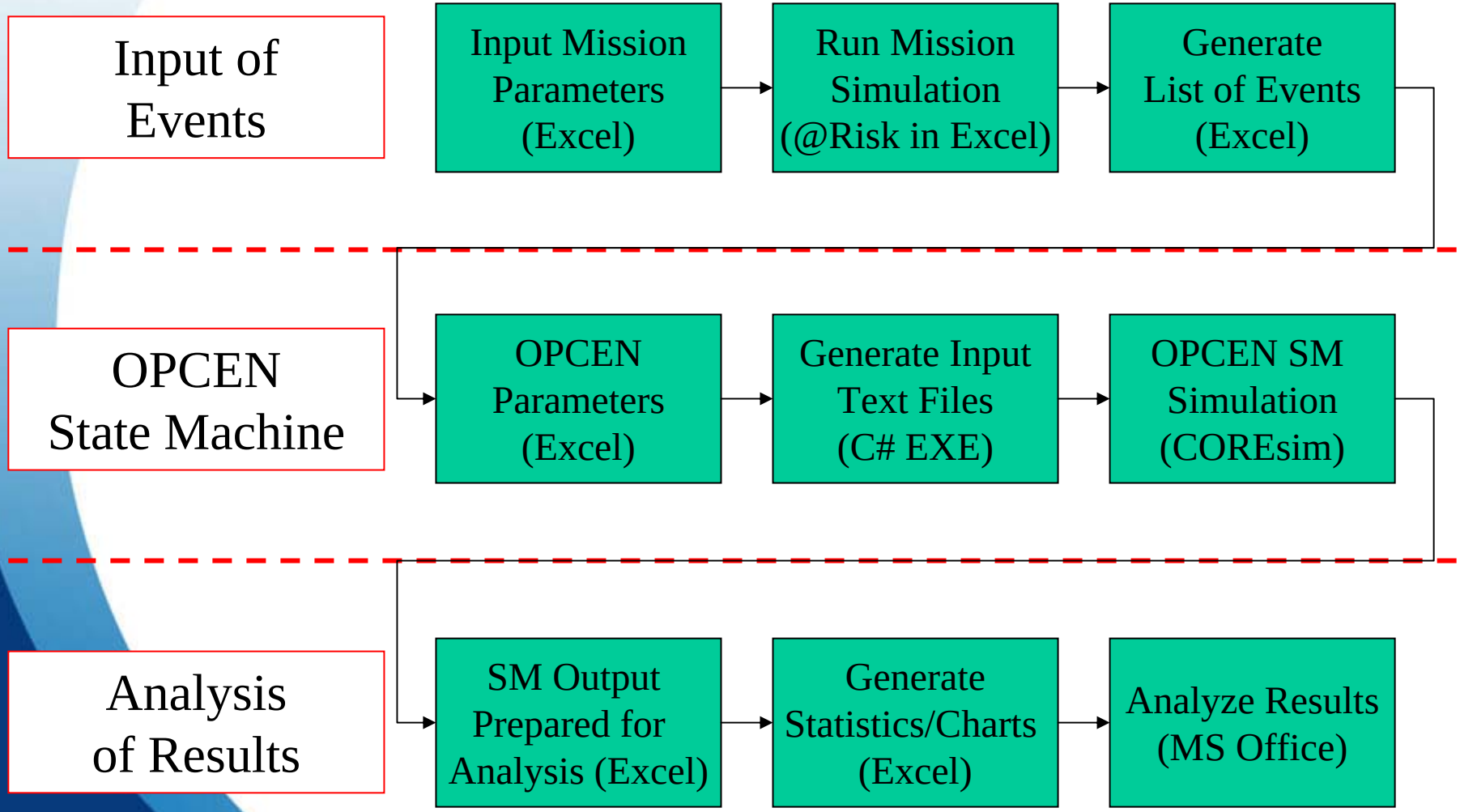
State Machine Top-Level Model



Thread Processing Characteristics

- Threads used to inspire the the State Machine thread processing element and are encoded within it
  - Process starts by rebuilding queue to deal with each job in order of priority
  - Interrupt mechanism determines what step is being worked upon
  - If job state is between steps, determine whether operator is available to begin next step
  - Repeat for next job in queue

Diagram of SM Simulation Process

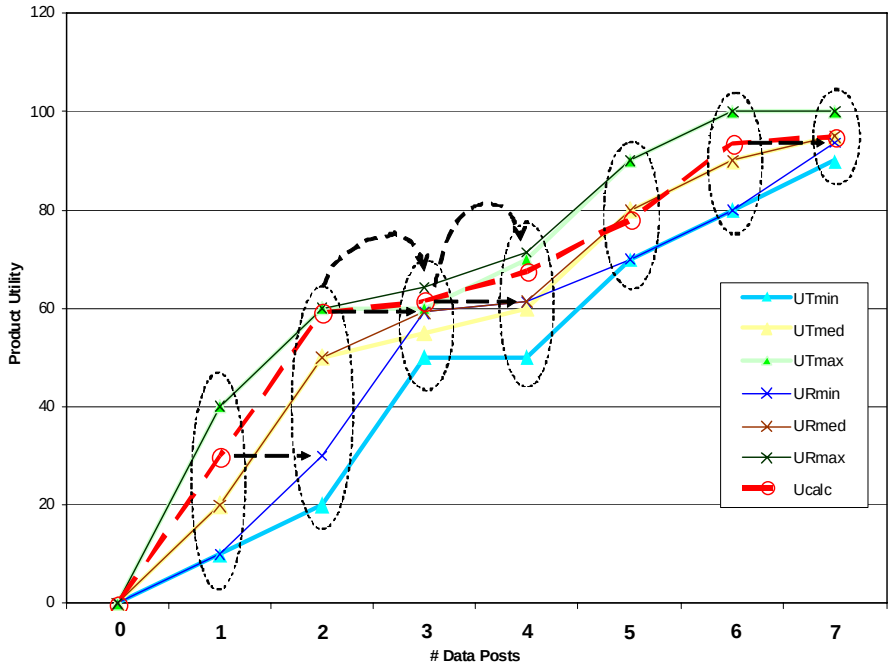


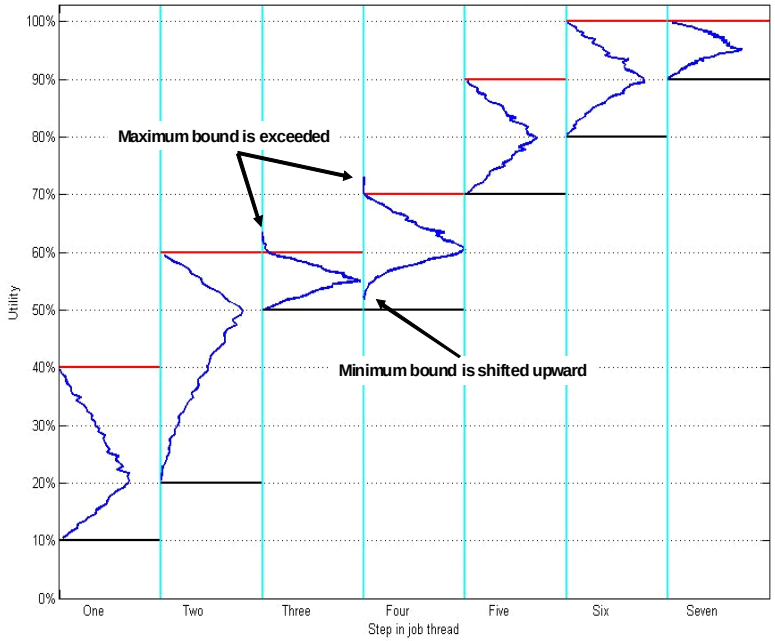
Business Rules Implemented (1 of 2)

- Ties between equal priorities broken by:
  - Least slack time
  - Most utility / time
- Operators allowed to have multiple skills:
  - Jobs allocated to assigned skill, then others
  - Generic jobs to idle operators in mandated order
- Interruptions
  - Operator can set aside Job A to work on Job B
- Available operator with highest (quality of work) X (speed) is preferentially assigned to jobs
- Event list triggers jobs that can be standard types or customized
- Switchboard to turn on and off business rules/options

Business Rules Implemented (2 of 2)

- Utility function
  - Operators define thread steps as uncertainty bounds for three-point estimates (i.e. minimum, most likely, maximum)
  - Value from previous step is used to revise distribution bounds

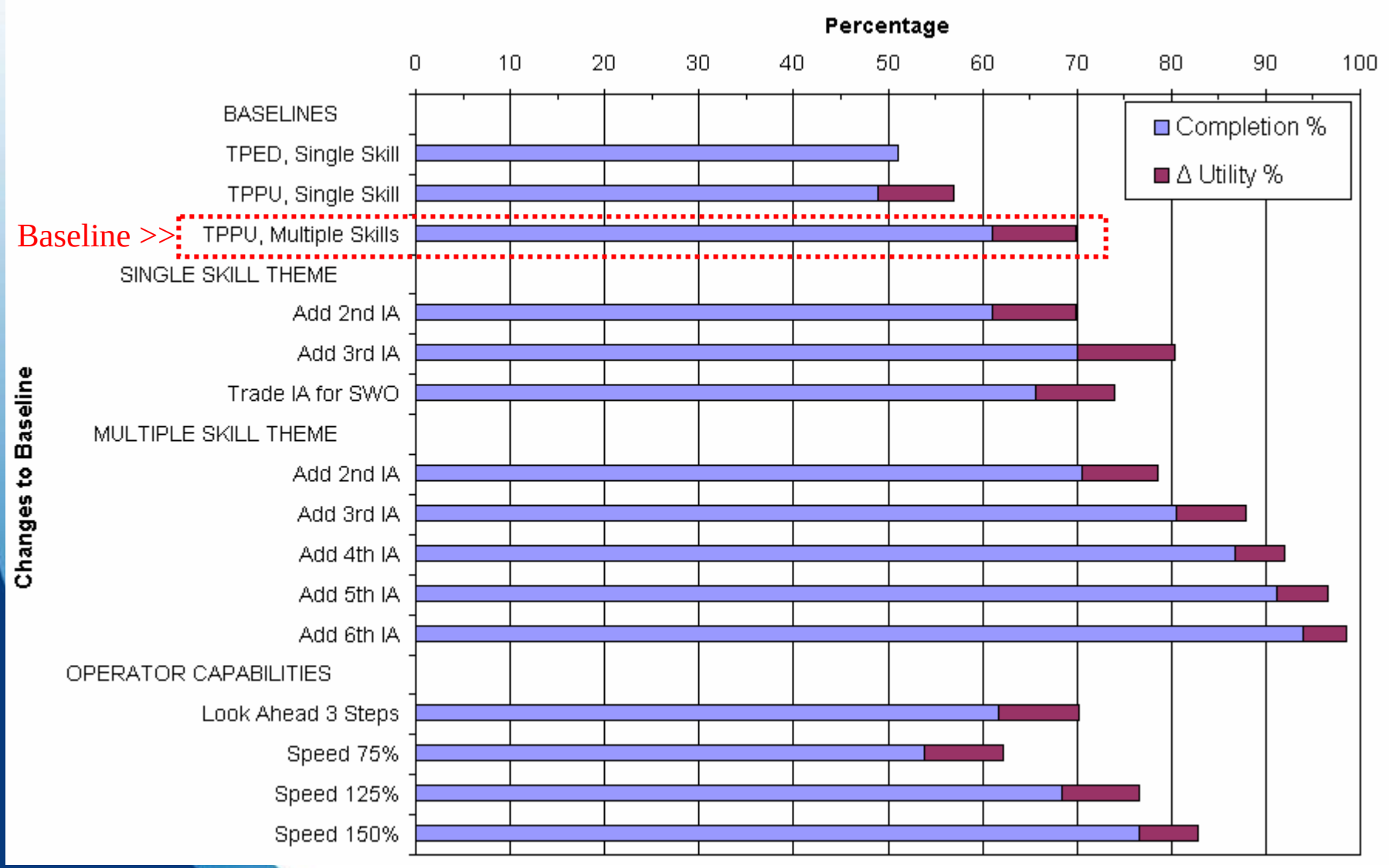




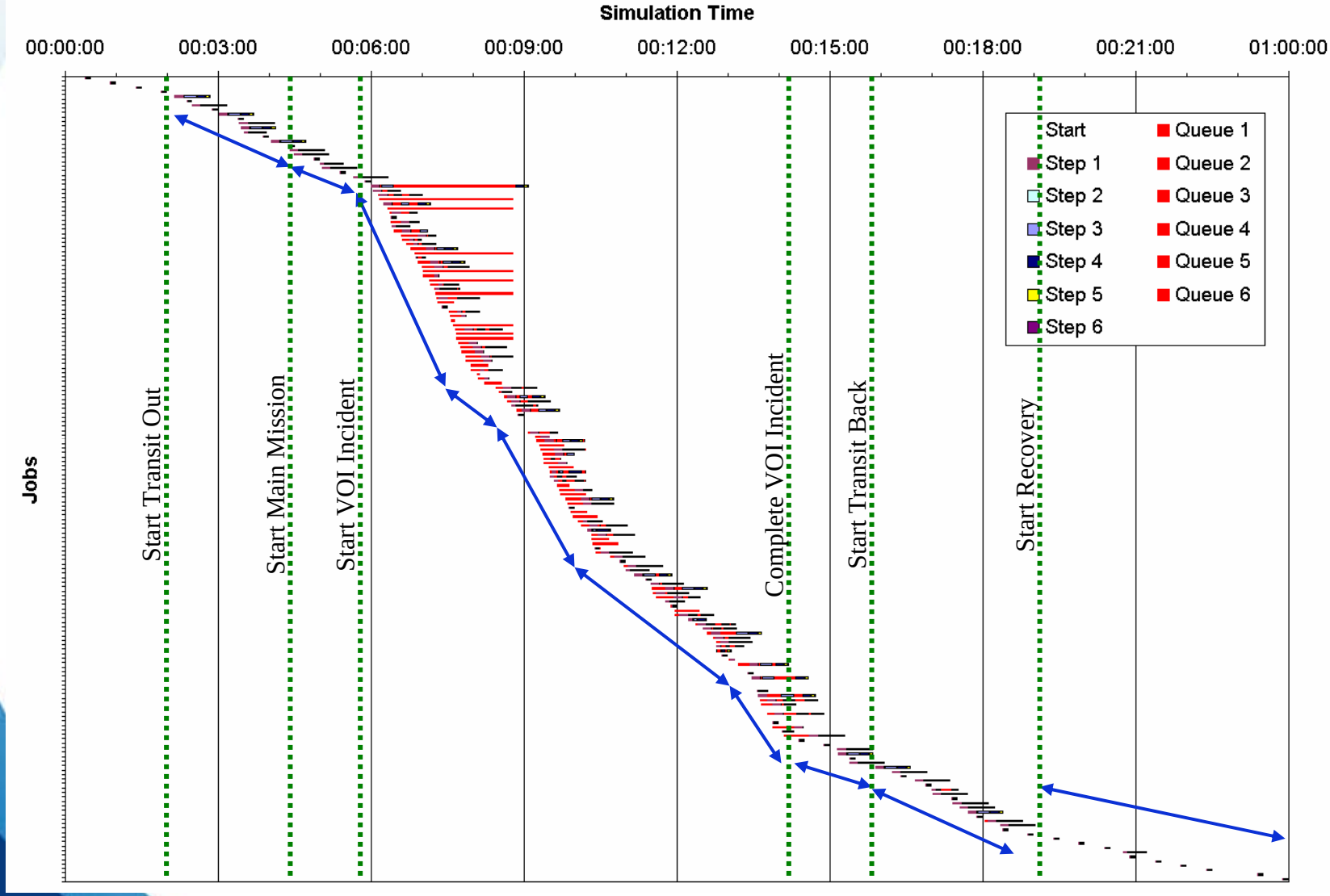
Simulation Analysis Metrics

- Metrics quantify activity but they also help to highlight anomalies
- Basic statistics
  - Job processing characteristics
    - Utility & Completion
    - Dwell (time spent idle in queues)
    - Churn (operator continuity between steps)
    - Volatility (disruption when steps interrupted)
- Activity along job timeline (using chart)
  - Duration of jobs (Rates of arrival and completion)
  - Operator utilization (i.e. SA needs idle time between jobs)
- Cumulative Utility
  - Consolidate products in SITREPs & assessments
- Entropy

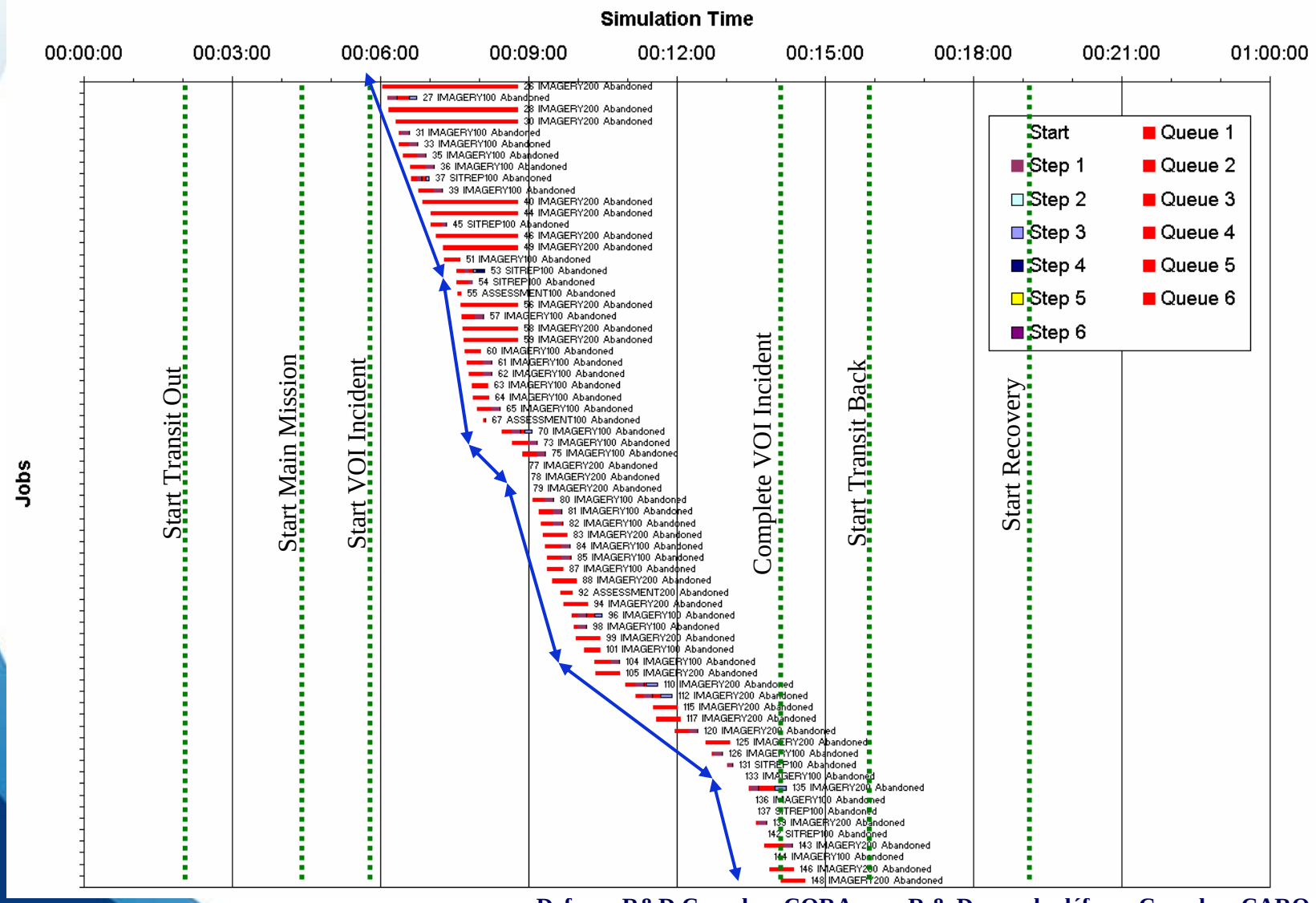
Comparison of State Machine Planning Assumptions



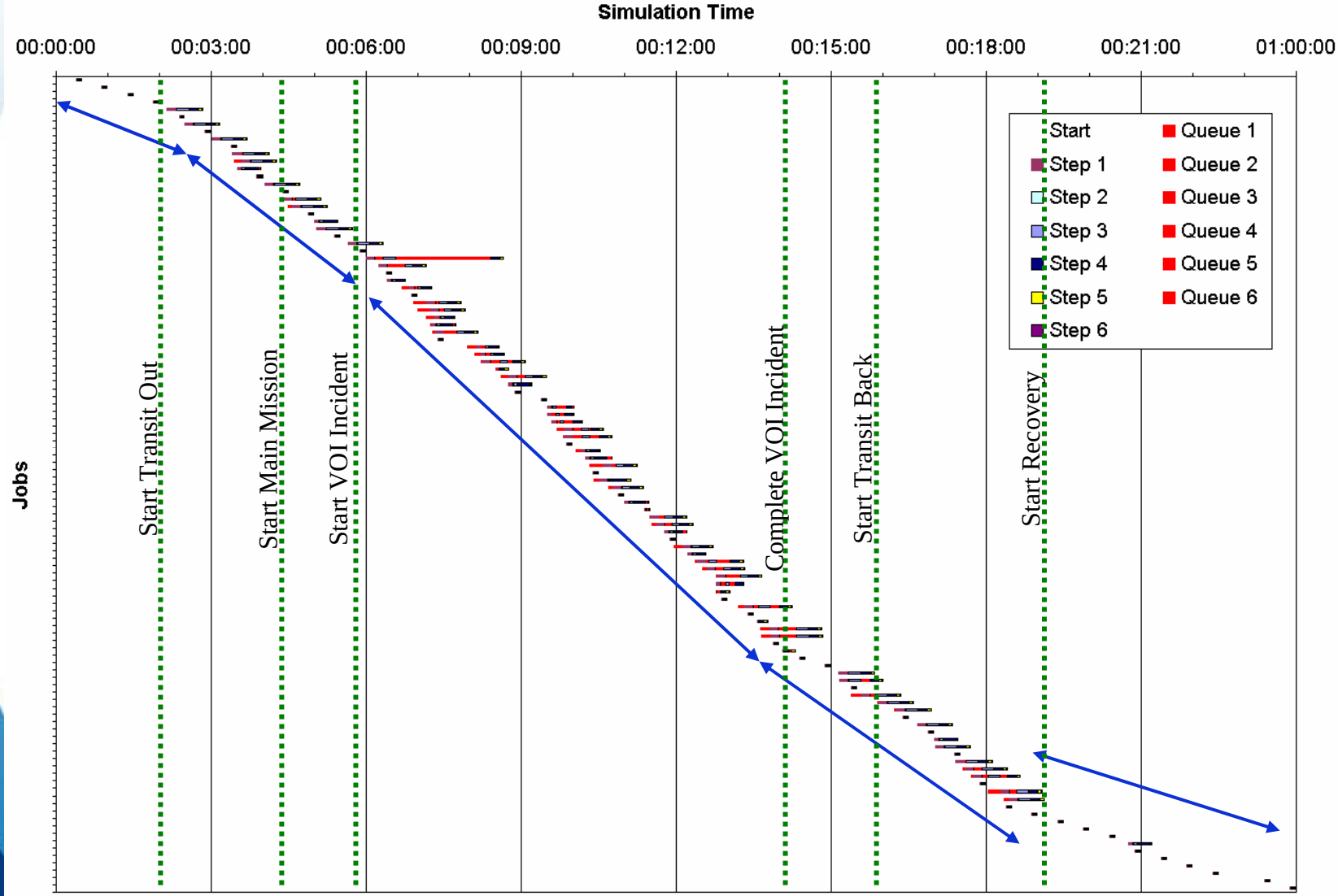
Timeline of All Jobs (Multiple Skills)



Timeline of Abandoned Jobs (Multiple Skills)

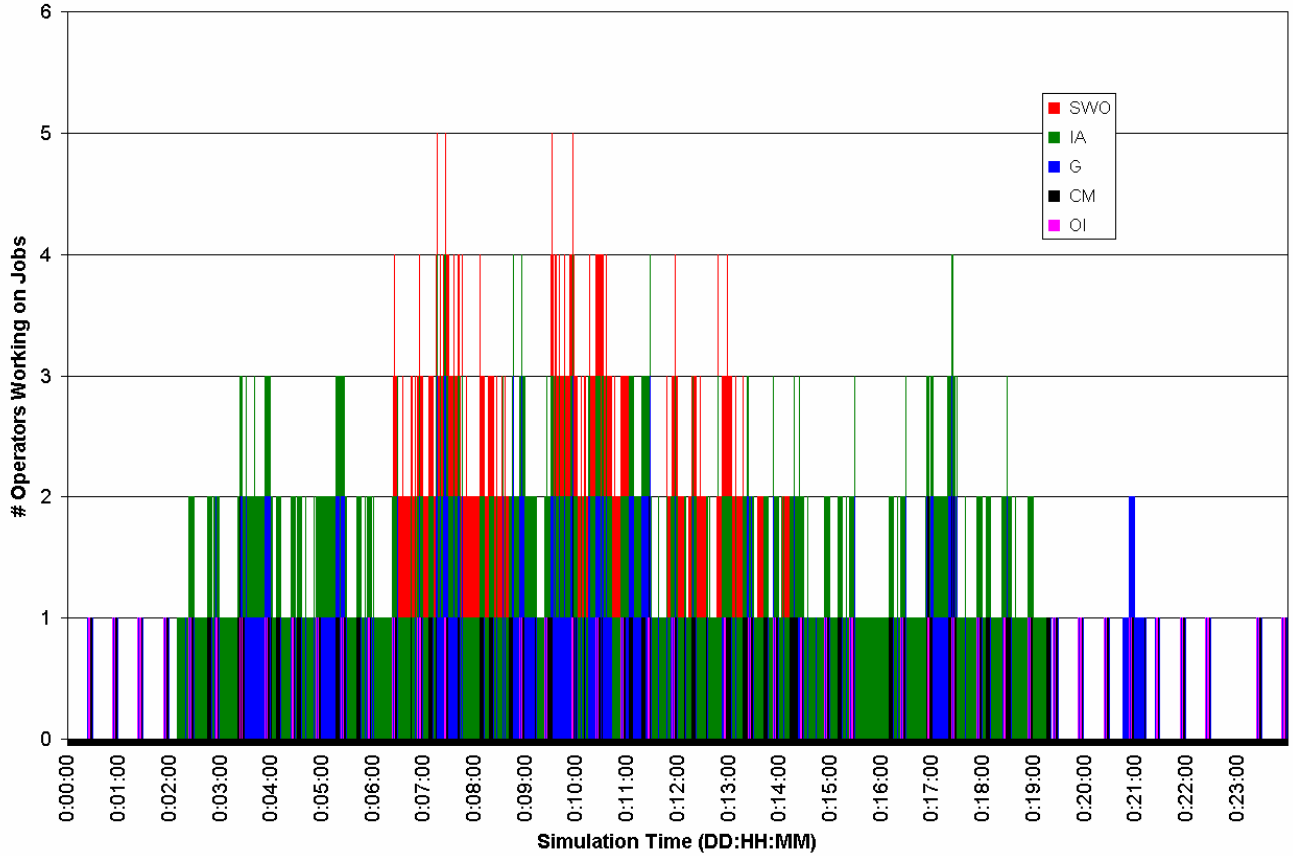


Timeline of Completed Jobs (Multiple Skills)



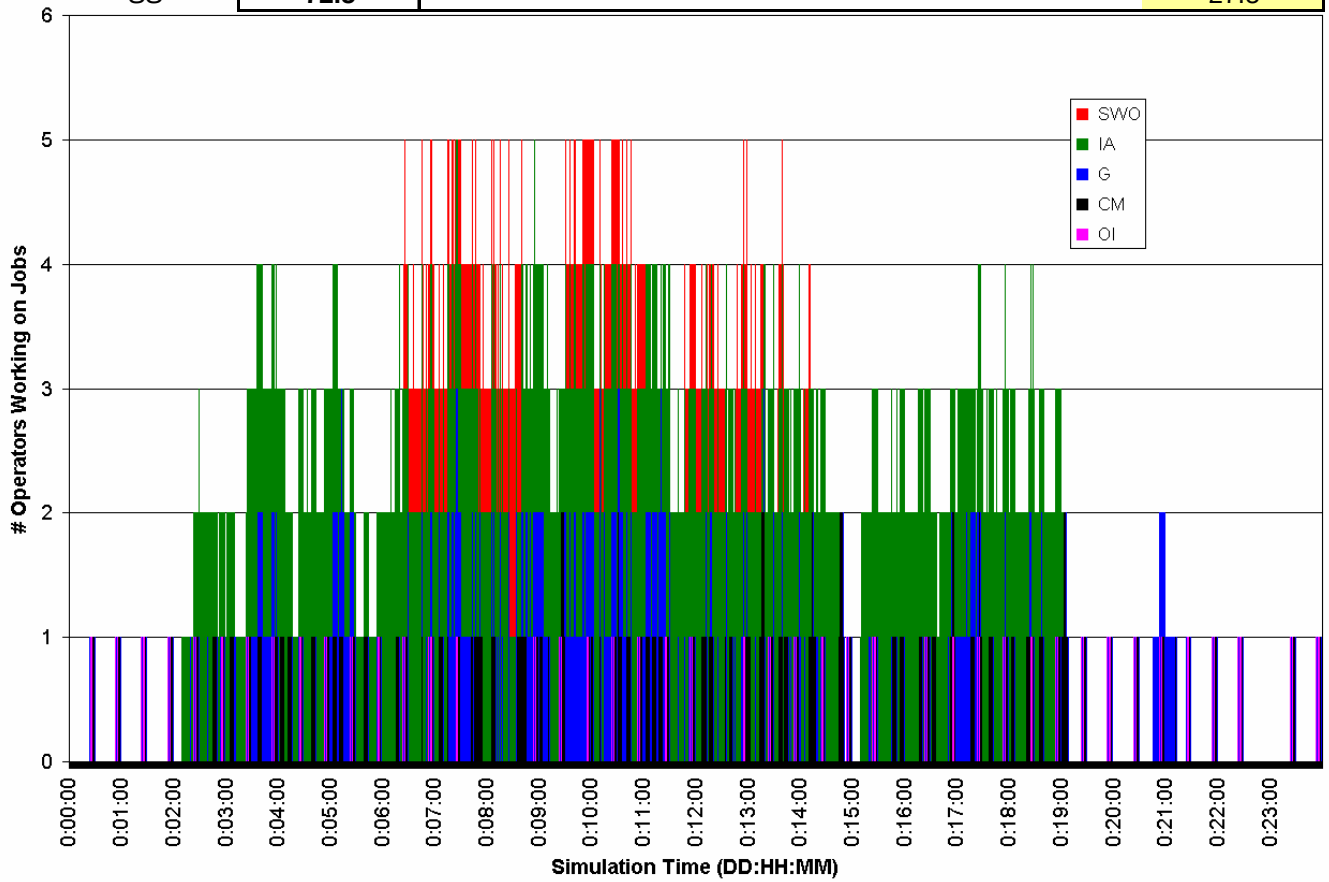
OPCEN Personnel Utilization (5 Operators With Single Skill)

Skill Used >>	*SA*	*SWO*	*IA*	*OI*	*CM*	*G*
Average	**70.3**	**4.3**	**14.2**	**1.8**	**3.3**	**6.1**
SWO	**78.4**	21.6				
IA	**29.2**		70.8			
OI	**83.3**			9.0		4.1
CM	**74.0**				16.3	0.4
SS	**86.9**					26.0

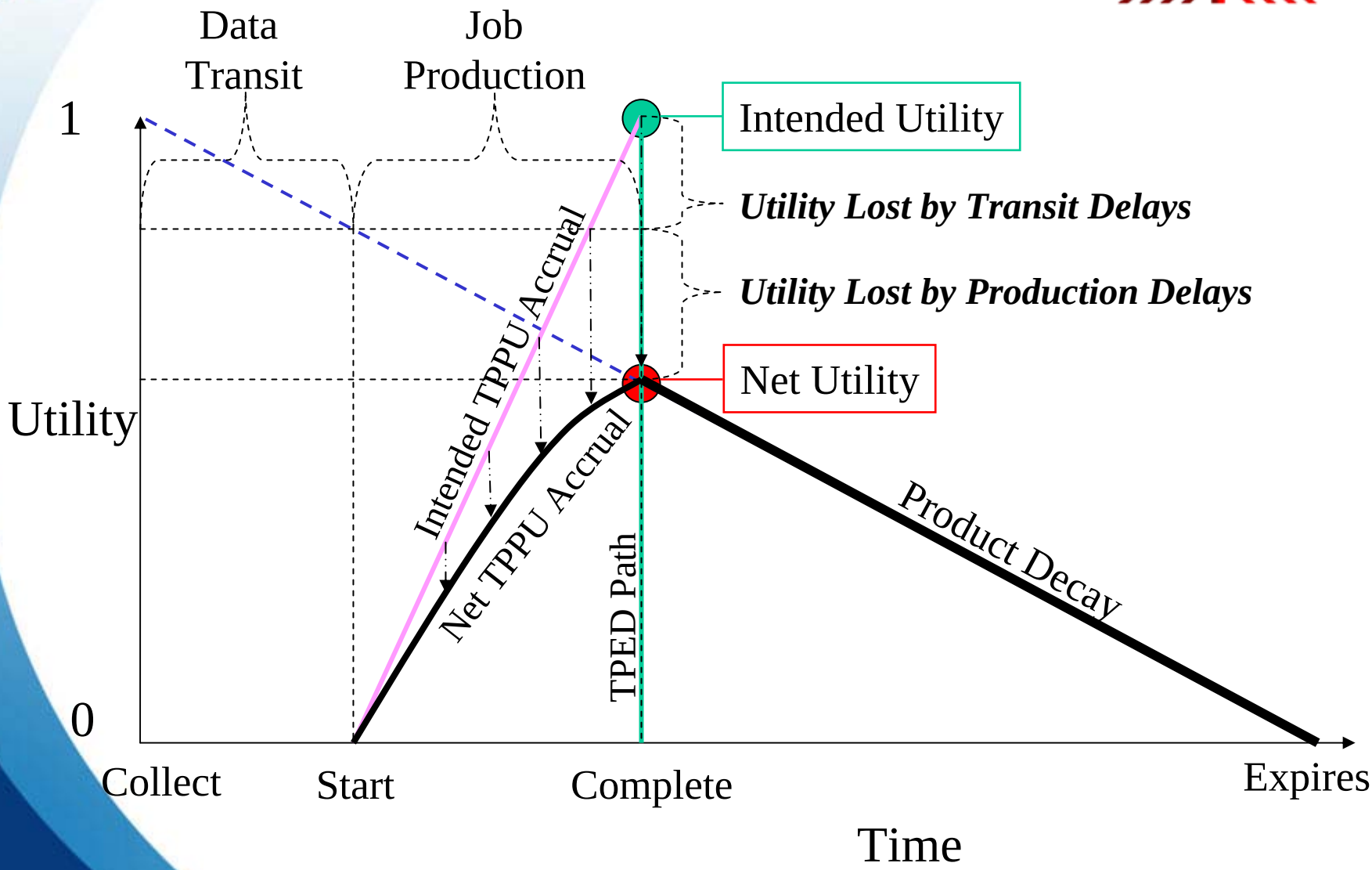


OPCEN Personnel Utilization (5 Operators With Multiple Skills)

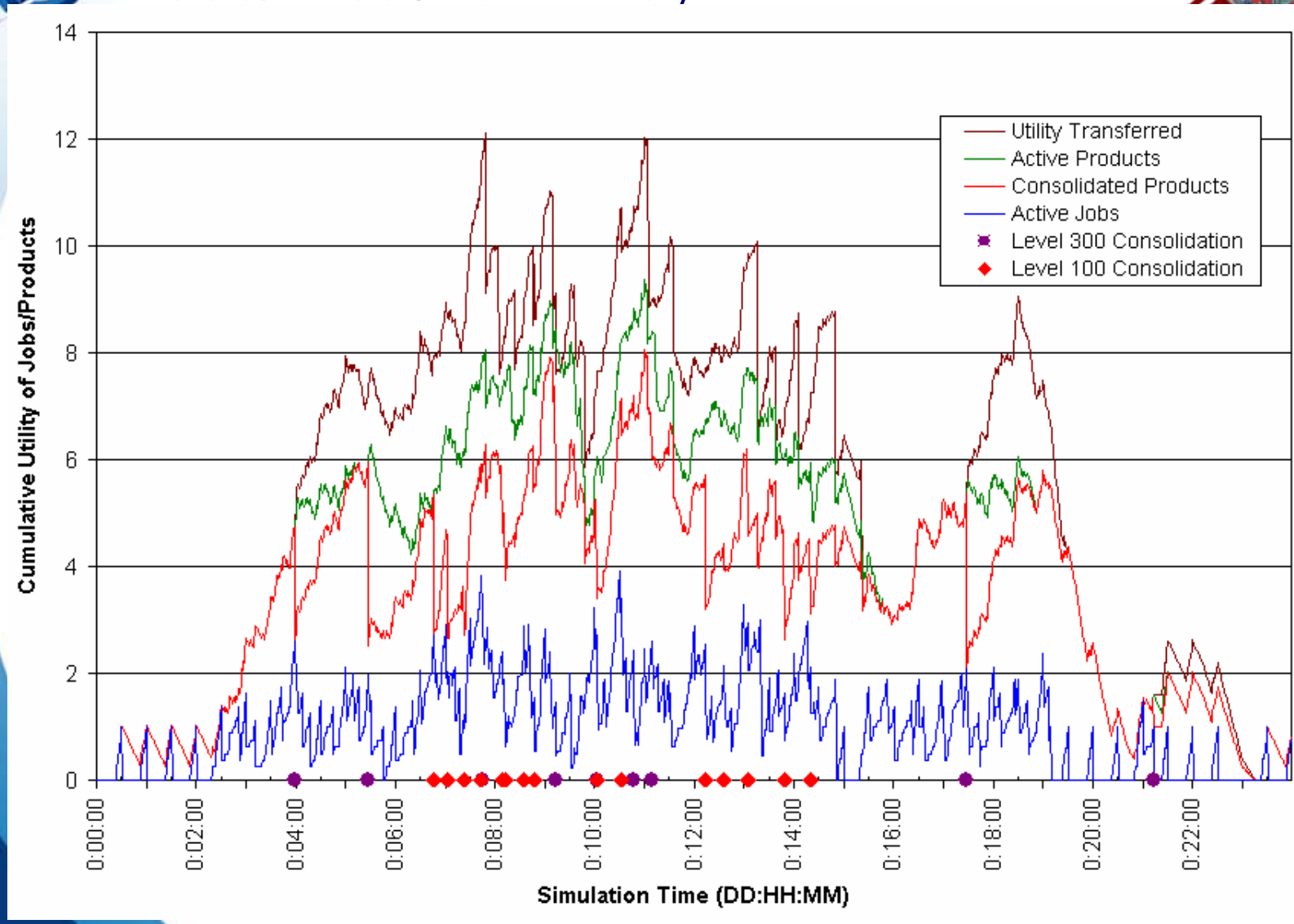
Skill Used >>	*SA*	*SWO*	*IA*	*OI*	*CM*	*G*
Average	**58.0**	**4.5**	**24.2**	**1.8**	**4.7**	**6.7**
SWO	**77.8**	21.5		0.6		0.1
IA	**33.7**	1.0	65.3			
OI	**70.8**			8.4	15.0	5.8
CM	**35.4**		55.9		8.7	
SS	**72.5**					27.5



Example of Utility Life Cycle (Using Linear Curves)



Example of How Utility Behaves in Job/Product Life Cycle



State Machine of Federated Nodes (SMOFN) Model

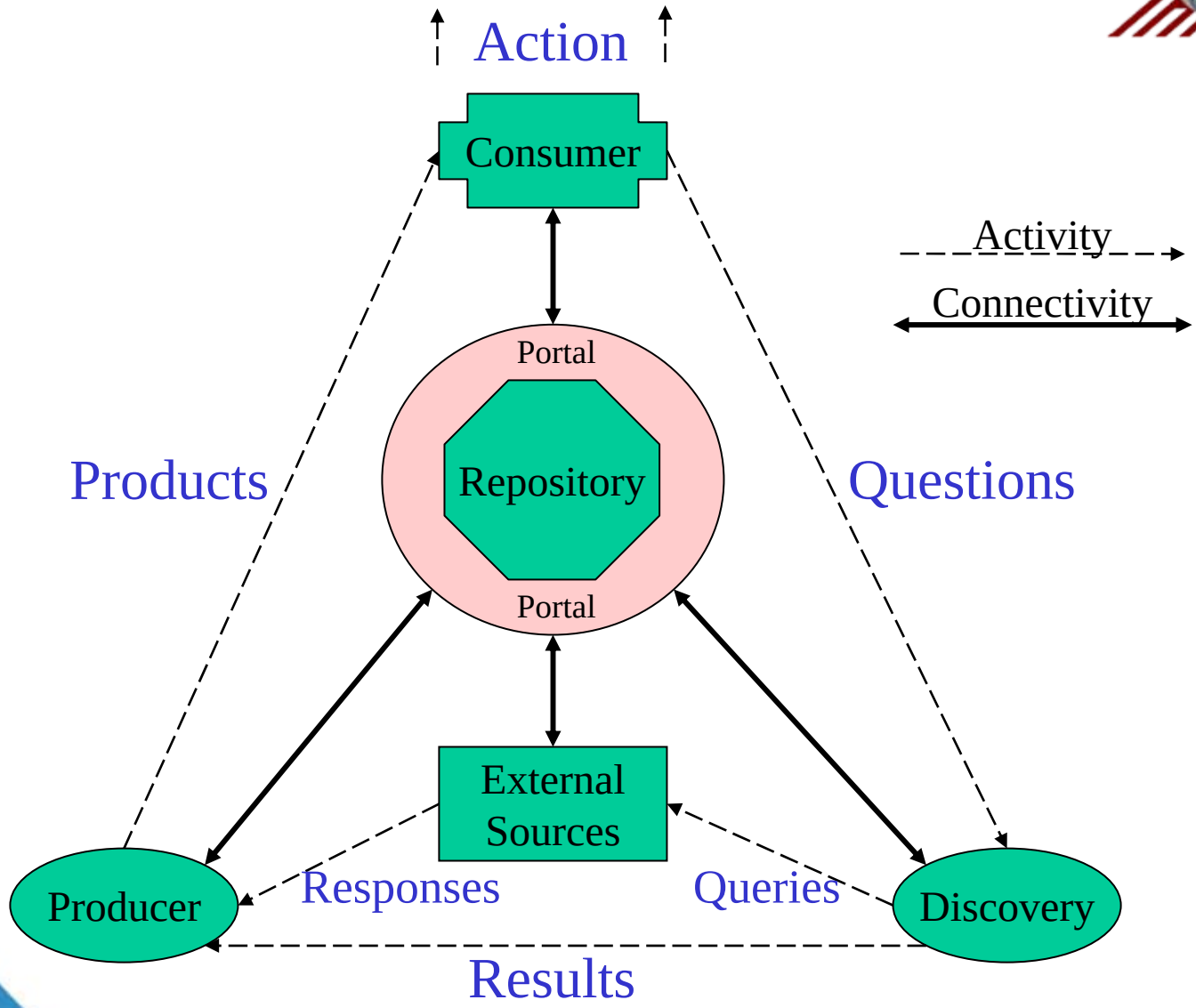
- CORA funded to develop capability engineering tools
- Extends SM model from single OPCEN to multiple locations
  - Completes implementation of utility life-cycle
  - Focus is on tracking flow of products between nodes
  - Repository node manages and shares products
- Project covers five phase (Jan 06 to Mar 07)

1. Explored issues with simple example using nominal data (Completed - Demonstrated that concept is feasible)

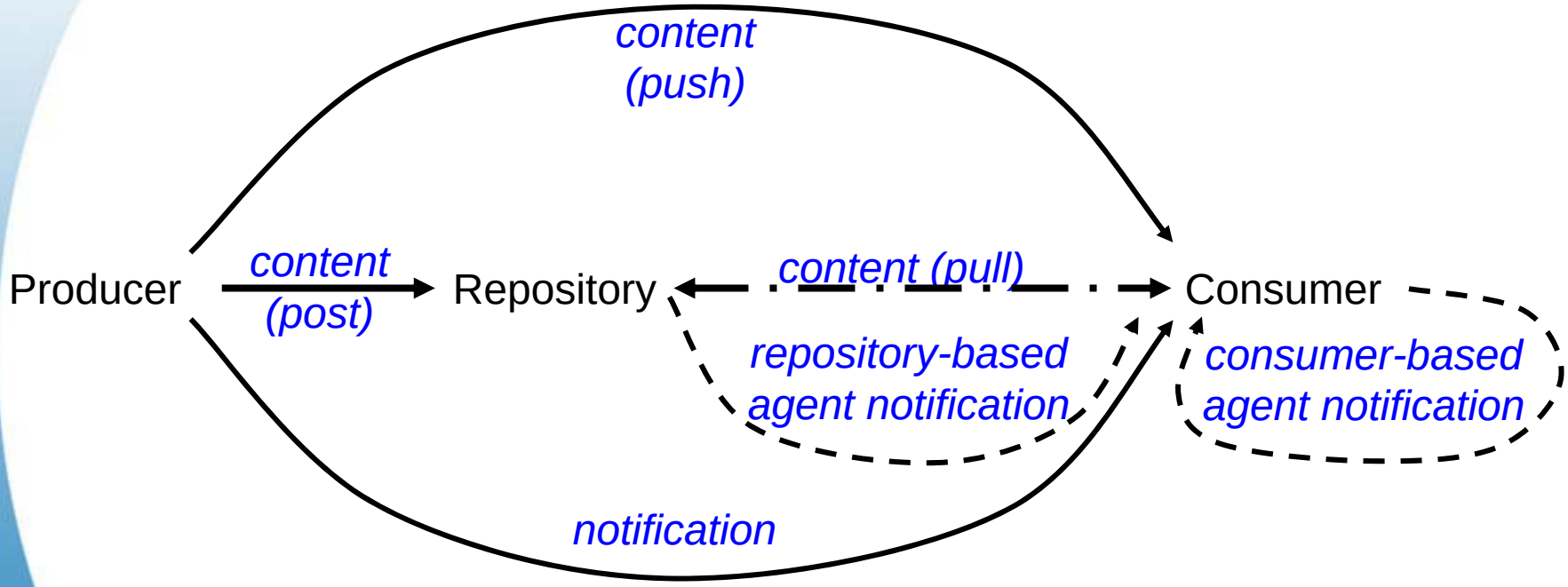
2&3. Develop and adapt use-cases as realistic version (Underway - Modelling the new CF Command Structure)

4&5. Refine analysis tools & support further use-cases (Funded - TBD)

Scale Free OV-1 for SMOFN



Producer-Repository-Consumer (PRC) Model



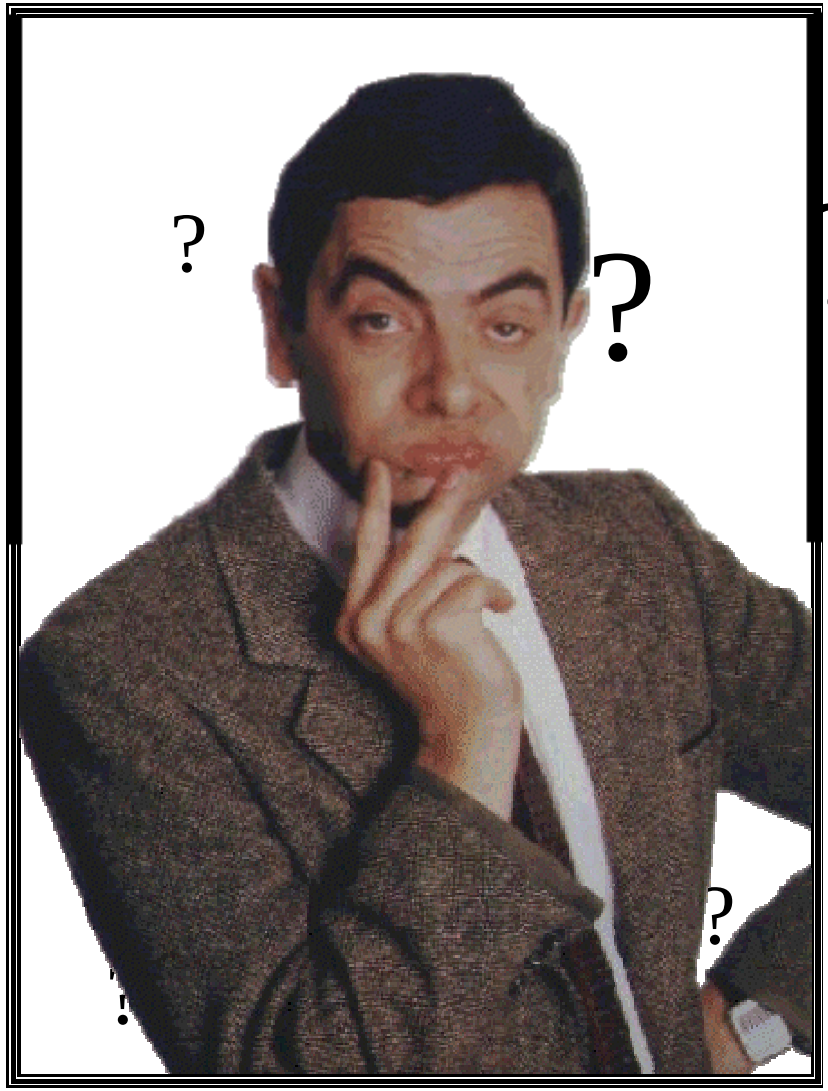
State Machine Model - Design & Limits

- The SM model is designed to:
  - Calculate how much work can be done on a set of jobs
  - Provide a basis to compare a range of business rules
- The SM model can account for job status:
  - Who is assigned
  - Where they reside as they flow through the thread
  - How business rules interact to direct the flow
- The SM model is NOT intended to explain:
  - Why a particular workflow pattern was chosen
  - How an operator actually accomplishes each step

Lessons Learned – So Far

- Talk is cheap and plentiful, very few work on real architectures
- DoDAF shows perspectives <u>into</u> problem
  - mostly based on PowerPoint “bubbleology”
- Executable architectures require MBSE approach
- Architectures are foundation datasets (like mapping data)
  - Requires long term investment
  - Becomes the enabler to evaluate project options
- Modelling apps tend to focus on collating and linking data
- Analysis tools are limited or non-existent
- Everybody talks about metrics but very few test their viability
- Classical behaviour models illustrate progression of jobs
  - Need SM to mimic net-centric concurrent activity

Questions



Contacts

Mr Ronald W. Funk & Mr. Mark Ball

Joint Staff Operational Research Team

DRDC Centre for Operational Research and Analysis

National Defence Headquarters (6CBS)

MGen George R. Pearkes Building

Ottawa Ontario Canada K1A 0K2

Ron.Funk@drdc-rddc.gc.ca (613) 995-6887

Mark.Ball@drdc-rddc.gc.ca (613) 992-4539

Mr Richard L. Sorensen

Senior Systems Engineering Consultant

Vitech Corporation

2070 Chain Bridge Road, Suite 100

Vienna, VA 22182

(801) 776-5794

rsorensen@vitechcorp.com

DEFENCE R&D DÉFENSE