ZETA ASSOCIATES

MathWorks Symposium

Adopting Model-Based Design within Aerospace and Defense

May 11, 2010 • Baltimore, Maryland

Next Generation Electronic Warfare Modeling and Simulation

Dr. Randall Janka

(janka@ieee.org)

Support to Military Operations

Agenda

- Introductions & background
- I2WD's next gen EW technology & mission drivers
- Scheduling & control for optimizing concurrent ES & EA
- Scheduling algorithm design
- Simulation framework design
- Demo
- Some performance observations
- Conclusion

Introductions & Background

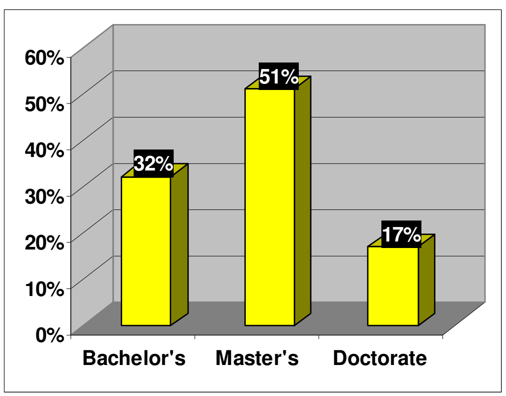
Who Is Zeta?

A Leading Intelligence Contractor

- Prime NRO contractor for:
  - SIGINT processing (Both SE and SI)
  - Enterprise Integration
  - COMINT system acquisition
  - COMINT solutions
- Highly valued provider for:
  - Military airborne signal processing
  - Special SIGINT programs
  - Collection products and survey tools to the IC

>300 Talented & Motivated Employees

Engineering Staff's Education
Highest Degree Earned



- 99% with clearances
- 99% retention rate
- 20 years average experience

Reputation for Reliability, Innovation, and Responsiveness

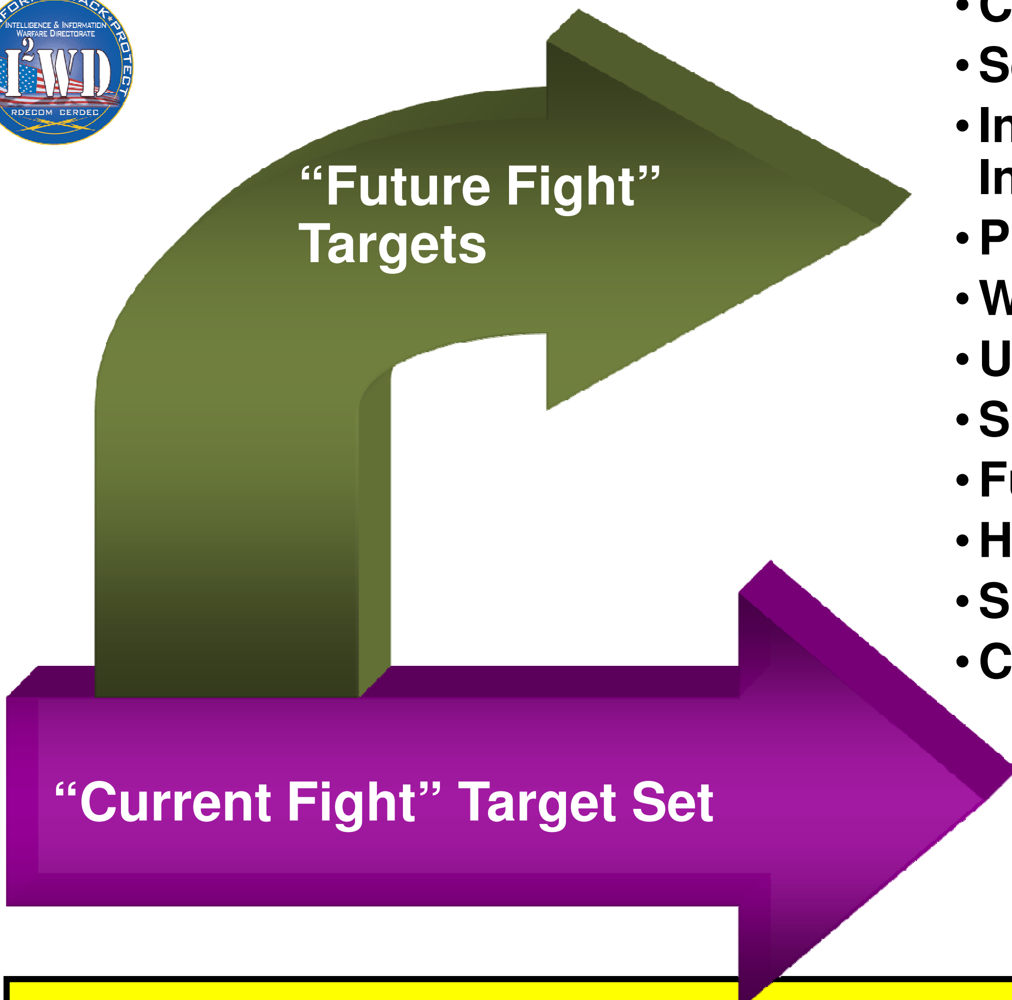
Strategically Positioned

- Enterprise Integrator
- Pioneering work on NRO Strategic Framework
- Agility of our software solutions
- Leveraging of solutions to DOD

Zeta's EW Mission Partner: USA's I2WD

- Zeta program support for USA/I2WD
  - SIGINT provider for USA's RC-12 Guardrail
  - Entering our third phase of next gen ES/EA scheduling & control applied R&D
    - Urban Sabre
      - M&S
    - IRON Symphony
      - Architecture development
      - M&S
      - Prototyping
- Recent collaborations
  - 54th JEWC (9/09)
  - 46th Annual AOC International Symposium & Convention Technical Poster Session (10/09)

I2WD's Vision of Army EW Into the Future



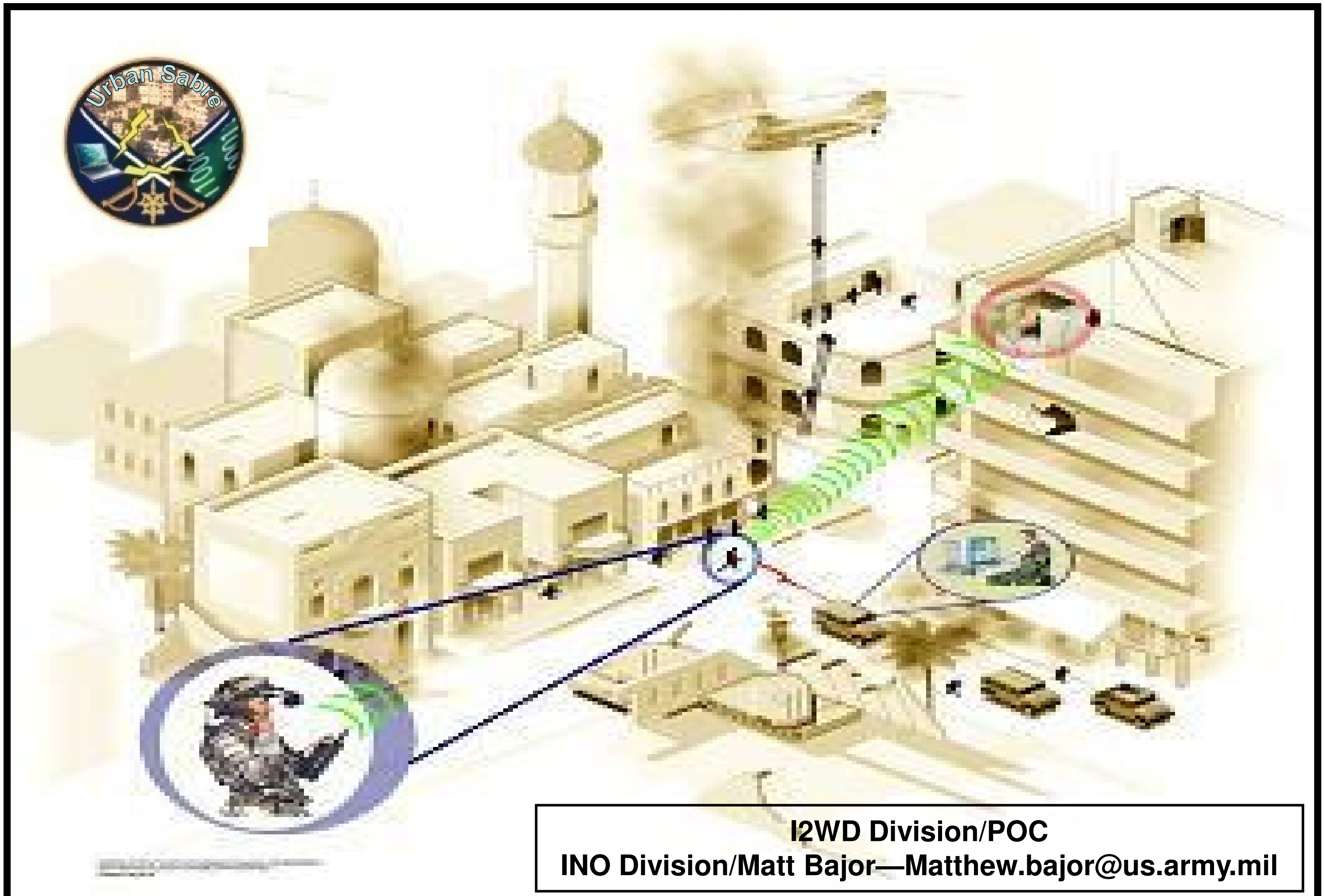
- Communications Systems
- Sensors
- Information Systems and Infrastructure (Comms)
- Positioning, Nav & Timing Systems
- Weapons of Mass Destruction
- Unmanned Aerial Vehicles
- Shoulder Launched Rockets
- Fused Projectiles
- Hard and Deeply Buried Targets
- Small, Highly Mobile Attack Teams
- Conventional Large Forces

Radio Controlled Improvised Explosive Devices (RCIEDs)

Army EW Must Address Broader Target Sets Than IEDs

I2WD's Next Gen EW Technology & Mission Drivers

Urban Sabre Vision



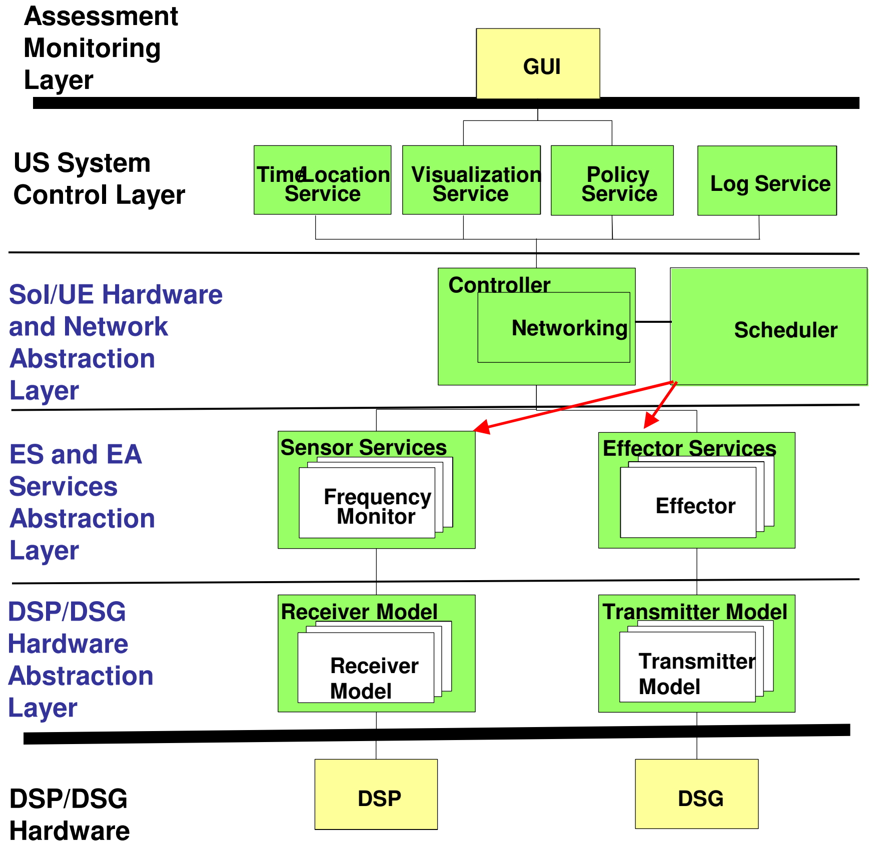
I2WD Division/POC
INO Division/Matt Bajor—Matthew.bajor@us.army.mil

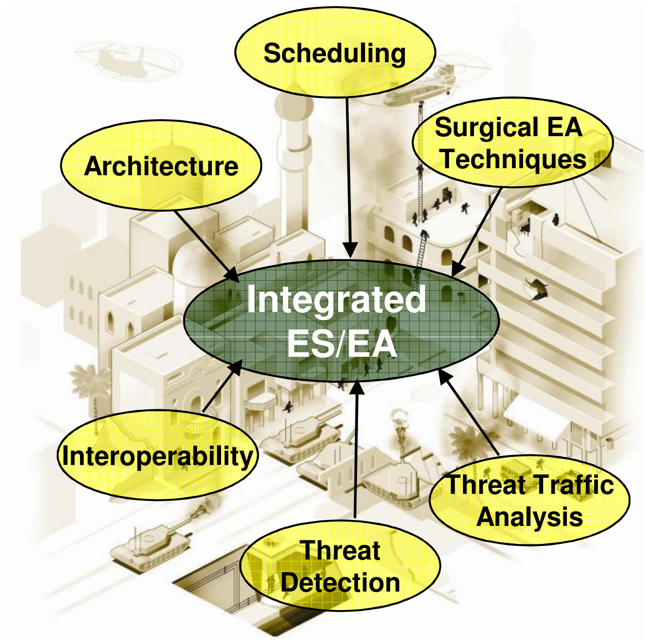
Provide an on-the-move (OTM) urban environment capability to—

Detect
Identify
Classify
Geolocate
Engage

— enemy C4ISR nodes in an urban environment

Urban Sabre System Architecture





Perform detect/ID/classification/geolocation/attack of a broad set of high priority wireless devices to regain & maintain control of the RF spectrum

Scheduling & Control for Optimizing Concurrent ES & EA

Problem & Solution Spaces

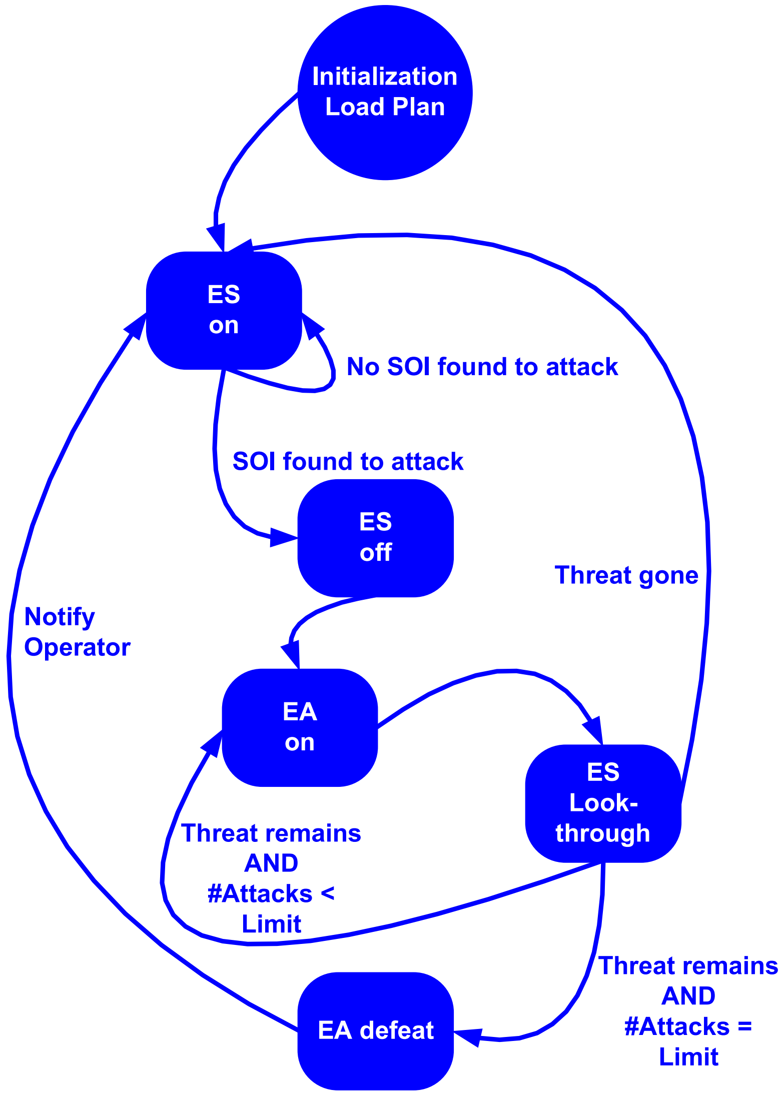
The Problem Space

- Need for concurrent execution of ES & EA missions
  - Important: ES & EA against diverse C^2
  - Critical: EA against RCIEDs
    - But defer this mission coverage to EP platforms
- Current inventory of legacy stovepiped ES & EA resources
  - Not easily adapted to rapidly changing EW space
  - Inhibit ability to schedule and coordinate ES & EA missions in real time
    - Lack sufficient processing resources in a given ES or EA resource
    - Do not have the necessary scheduling algorithms
  - Lack interoperability
- Cannot manage both resources in a unified fashion
  - Spectral fratricide

The Solution Space

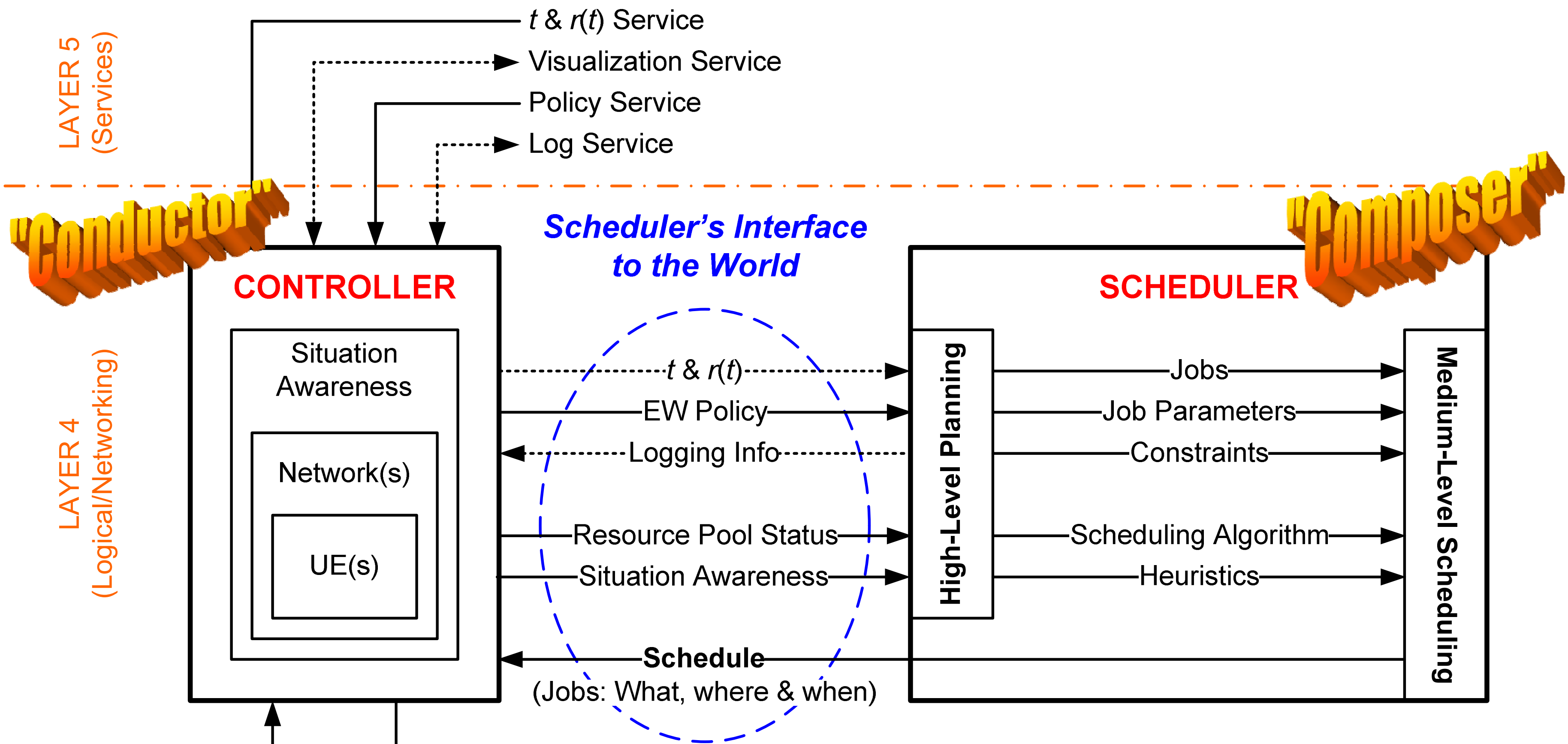
- Develop new real-time scheduling approach(es) to enable simultaneous execution of ES & EA missions
  - Concurrent control of ES & EA resources
    - Optimal utilization of HW/SW resources
    - Compliant with user-defined policies
  - Autonomous control in real-time
    - Optimize its target engagement schedule
    - Maximize effectiveness & efficiency
- Construct an open architecture
  - Dynamic management of HW/SW resources
    - Low-frequency configuration management
    - High-frequency application of resources to targets
  - Allow rapid integration of new EA/ES techniques against emerging targets
  - Based on industry best practices
- Allow for extension to the net-centric operations

High-Level State Diagram



- Basic template
  - E.g., could include HVT SOIs for immediate EA
- Pre-mission plan to be loaded:
  - SOI list for ES to prosecute
  - EA plan for SOIs
    - Techniques (M:N)
    - Policy
      - Priority order of techniques
      - Definition of resolution, i.e., when do you declare victory or defeat, then what?
  - Quiescent schedule
    - Default ES schedule policy when not processing reactive EA jobs
    - Policy for rescheduling
- For this CONOP, we need to model & optimize:
  - Planning
  - Scheduling

The Scheduler's View of the World



Controller handles low-level behavior management, assignment and execution

Scheduling Algorithm Design

Theoretic-based Scheduling Algorithms

- Theoretic = "AIOR" (AI + OR) or "AO"
  - AI = Artificial Intelligence
  - OR = Operations Research
- AI-based planning for creating EA tasks
  - Pragmatic AI approach
  - Use Partial Order Planning (POP) based on the Hierarchical Task Network (HTN) notion
  - Re EA task creation, this reduces to probabilistic-based ranking of techniques
- OR-based scheduling for ordering the EA tasks w.r.t. time and processor
  - Classical OR approach to optimize the use of resources, typically w.r.t. time
  - Applied flexible scheduling approaches
  - Leveraging the TORSCHE MATLAB scheduling toolbox from the Czech Technical University in Prague

AIOR Scheduling: Problem Definitions

- Problem descriptions reduce to three scheduling problems of the same form but different heuristics
  - Problem description uses standard form: $\alpha \mid \beta \mid \gamma$
  - α = "Machine environment"; i.e., the target platform
  - β = Processing characteristics & constraints; e.g., precedence, preemption, etc.
  - γ = Objective to be optimized
- $Pm \mid \mid C_{max}$
  - Using SPT, WSPT & LPT
- $Pm \mid \mid \Sigma C_j$
  - Using ECT & EST
- $Pm \mid \mid \Sigma w_j C_j$
  - Using ECT & EST

$Pm \mid \mid \gamma$

Pm = *m* processors of the same kind
| | = no constraints
γ = objective function to be optimized

Objective Functions (γ)

C_{max} = makespan; i.e., ~ completion time of last EA task (min Cmax implies good utilization)
ΣC_j = total completion time
$\Sigma w_j C_j$ = total weighted completion time

Heuristic Strategies

SPT	Shortest processing time first
WSPT	Weighted SPT first
LPT	Longest processing time first
ECT	Earliest completion time first
EST	Earliest start time first

Pragmatic Algorithms: Overview

- Pragmatic Algorithms
  - Best Effort (BE)
    - Discovered and adapted during study phase
  - Best Effort Optimized (BEO)
    - Improved BE developed during M&S
- Best Effort (BE)
  - Greedy scheduling algorithm
  - Pre-Simulation creates a look up table for techniques available to each SOI type and channel pair and orders them by their effectiveness level
  - Ranks each individual SOI type and channel pair according to priority allocated for each new SA report received
  - Guarantees the highest value targets will get scheduled first
  - Guarantees highest value targets will utilize most effective techniques for given SOI
- Best Effort Optimized (BEO)
  - Similar to BE
  - Starts to look at the group as a whole
  - Pre-Simulation creates a look up table for techniques available to each SOI type and channel pair and orders them by their effectiveness level
  - Ranks each individual SOI type and channel pair according to priority allocated for each new SA report received
  - Re-orders look up table created pre-simulation due to the number of SOI type and channel pairs contained in the SA report
  - Rest of algorithm matches BE

MOE & BDA

- MOE: Scheduler metric
  - Measures the number of SOIs successfully scheduled versus the number of SOIs reported in the SA reports.

$$MOE = \sum \frac{SOIs\_scheduled}{SOIs\_reported}\left(\|priority\_vec\|\right)$$

- BDA: System metric
  - Measures the number of SOIs successfully attacked and destroyed versus the number of SOIs reported in the SA report

$$BDA = \sum \frac{SOIs\_destroyed}{SOIs\_reported}\left(\|priority\_vec\|\right)$$

Simulation Framework Design

Modeling & Simulation (M&S) Initial Thoughts

- Initial analysis
  - Wanted to use MATLAB (M/L) for rapid evaluation whenever possible
    - Modeling and evaluation of algorithms
    - Associated data models
  - Expected to have to port non-M/L algos into M/L for analysis
    - C/C++ (algos?)
    - Java (UI?)
- Simulation framework
  - After we had modeled the simulation…
    - I/O
    - Signal environment
    - Simulation dynamics
  - …Then we would know what kind of DE-based framework to use
    - Roll-your-own M/L DE engine?
    - Simulink—possibly with Stateflow a/o SimEvents?
  - UI for front and back ends
    - User inputs: Scenario, EA lib & POP updates, etc.
    - Display: Data logging (TBD), MOE display, etc.
- View towards rapid prototyping
  - Use Embedded M/L (EML) if/when possible to make porting to prototype easier

Modeling Tools and Approach Evolution

- Tool considerations
  - Preferences
    - Right level of granularity & of fidelity and able to interface with M/L
  - M/L Toolbox candidates for modeling behaviors
    - From the MathWorks
      - Statistics, Optimization, Direct Search & Genetic Algorithm
    - From third parties
      - TORSCHE (Time Optimisation, Resources, SCHEduling)
        - » Recommended by the Godfather of Scheduling (Prof. Michael Pinedo)
      - Czech Technical University in Prague ("Czech Tech")
      - Educational freeware ⇒ didn't know its limitations
        - » You always get what you pay for with freeware!
  - M/L-friendly simulation tools if at all possible for ease of integration
    - Simulink? SimEvents a/o Stateflow?
- Use a model driven architecture approach that is rapid prototype friendly
  - Use Embedded M/L (EML) coding style
  - Possibly use Simulink with Stateflow a/o SimEvents and Real-Time Workshop (RTW) for rapid prototyping

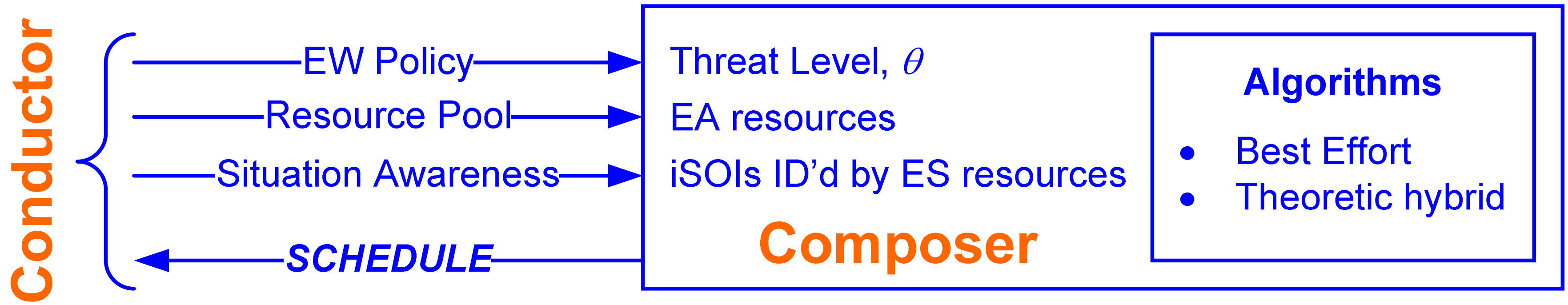
ZETA ASSOCIATES

The MathWorks MATLAB & SIMULINK

Simulation Tools and Approach Evolution

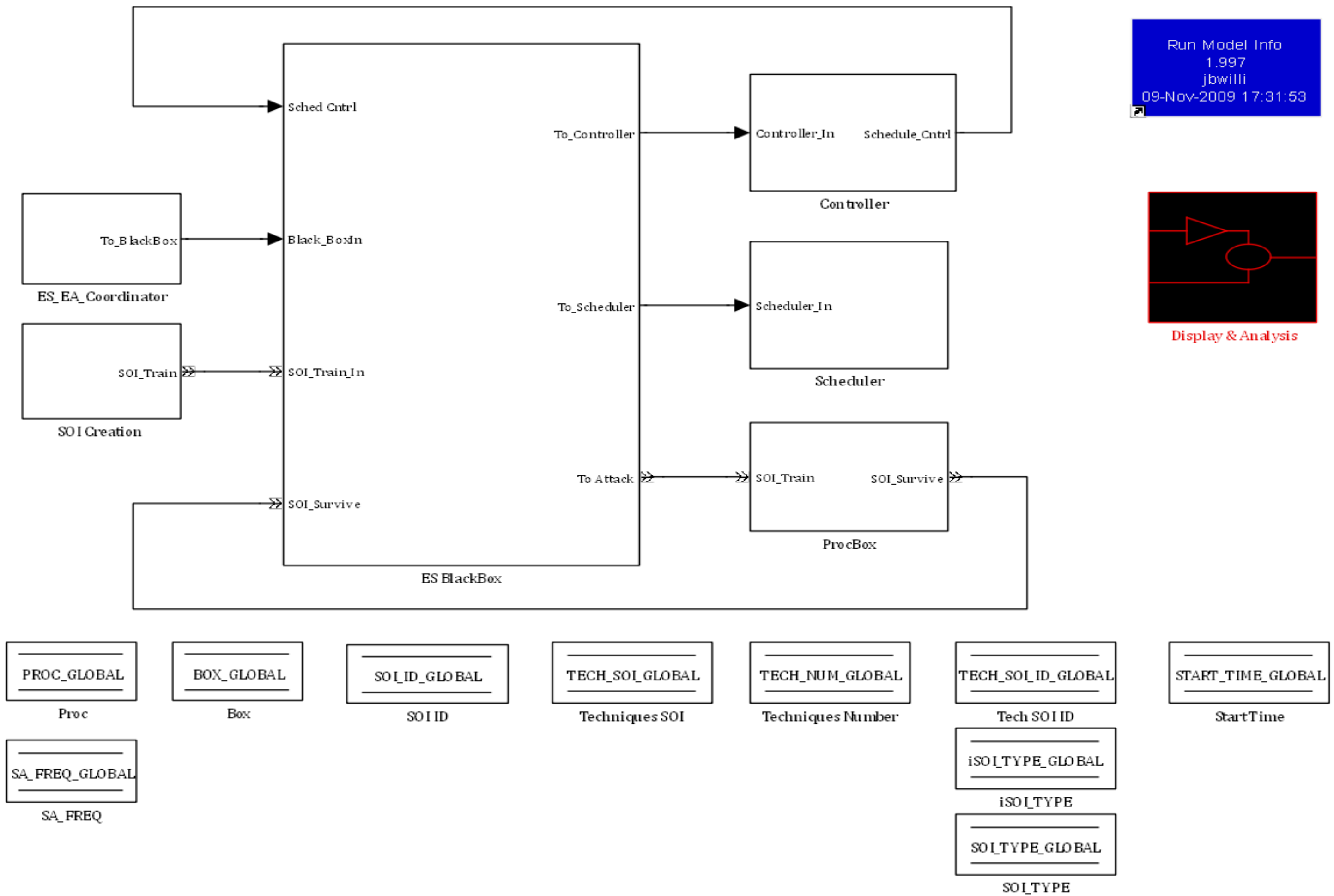
- Development platform
  - Mathworks' MATLAB/Simulink
- Embedded MATLAB (EML) code
  - Easily translated into either C or C++
  - All schedulers and deliverable code
- SimEvents
  - Discrete modeling environment
  - SOIs can be modeled as events that happen over time
    - Events are modeled as Entities
  - Flexible for expansion in later phases
- StateFlow
  - Model state based transitions in the environment
    - Determine when to create SA reports
    - Determine when to execute a produced schedule

Simplified Component View

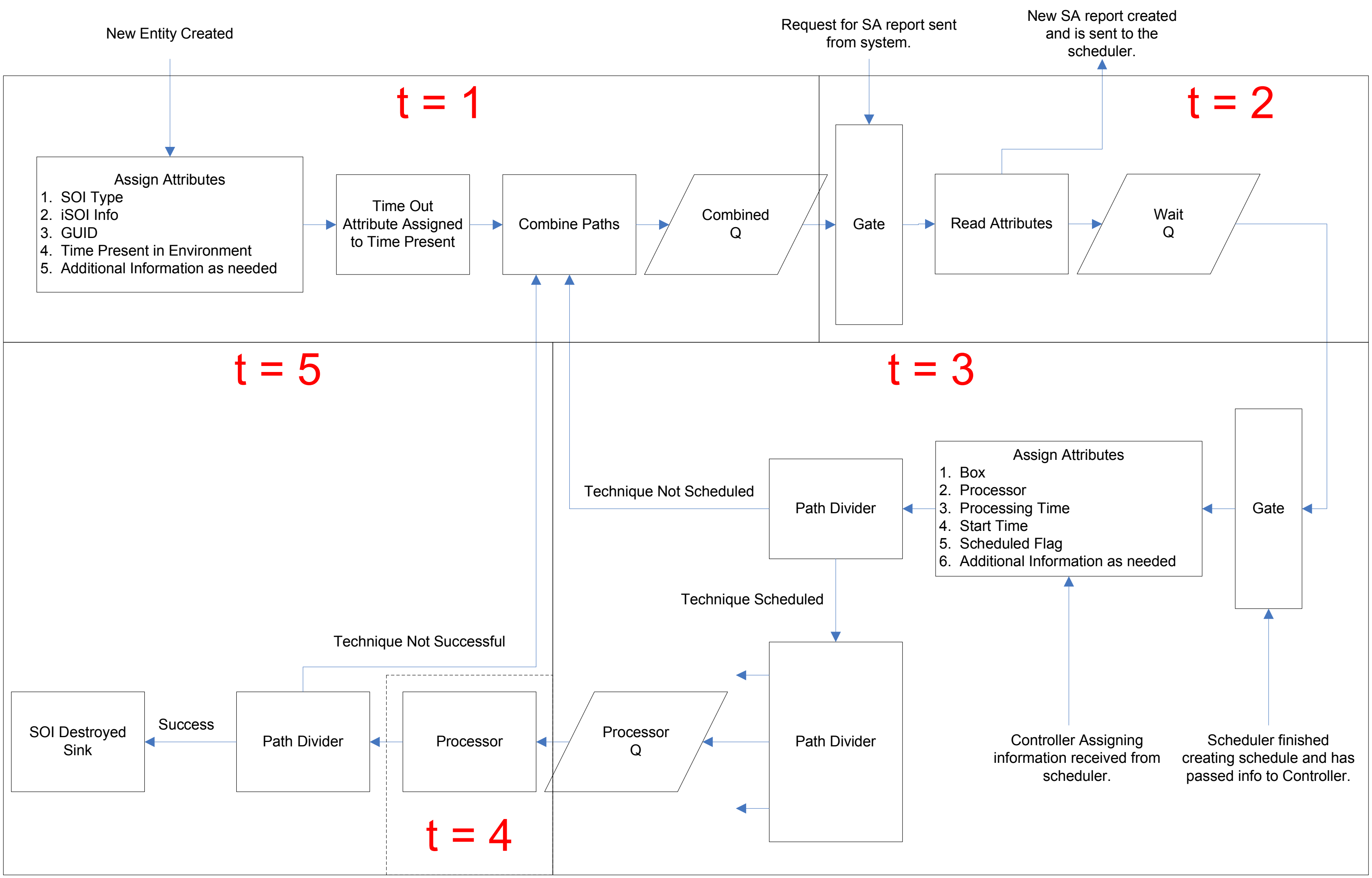


- Test harness
  - Environment sampled by ES resources
  - Generates SA reports
- Composer
  - Consumes SA reports and generates an optimal schedule based on user EW Policy and resources
  - Best Effort & AIOR
- Conductor
  - Executing schedule against signal targets
  - Simulated in the framework

Simulation Model Architecture



Simulation Flow Chart



Description of Flow Chart

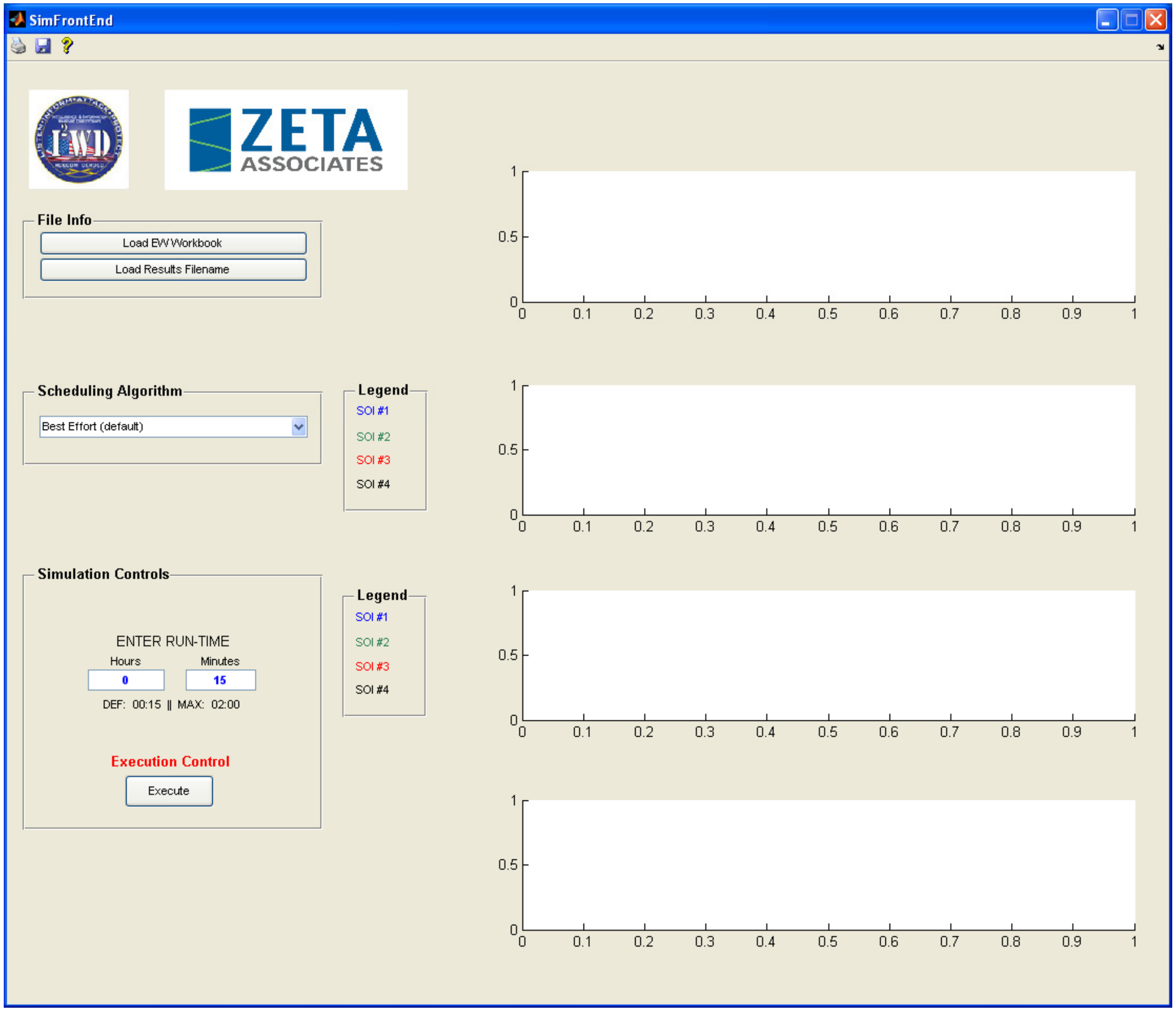
- t = 1 (SOI Creation)
  - SOIs created in the environment
  - Identification information added to SOIs in the form of attributes
  - SOIs added to the combined queue
    - Queue represents the SOIs currently in the environment
- t = 2 (SA Creation)
  - SOIs contained in the queue released
  - Information is read from the attributes
  - SA report is created
  - SOIs are placed in a wait queue
- t =3 (Schedule SOIs)
  - Schedule is created by the algorithm
  - Controller interprets schedule and attaches information about execution via attributes
  - SOIs scheduled are flagged and sent to processors
  - SOIs not scheduled are sent back to the combined environment queue
- t = 4 (Executing Techniques)
  - Techniques scheduled to process for the amount of time required
- t = 5 (BDA)
  - Techniques marked as successful are destroyed
  - Techniques not successful are returned to the combined environment queue

Demo

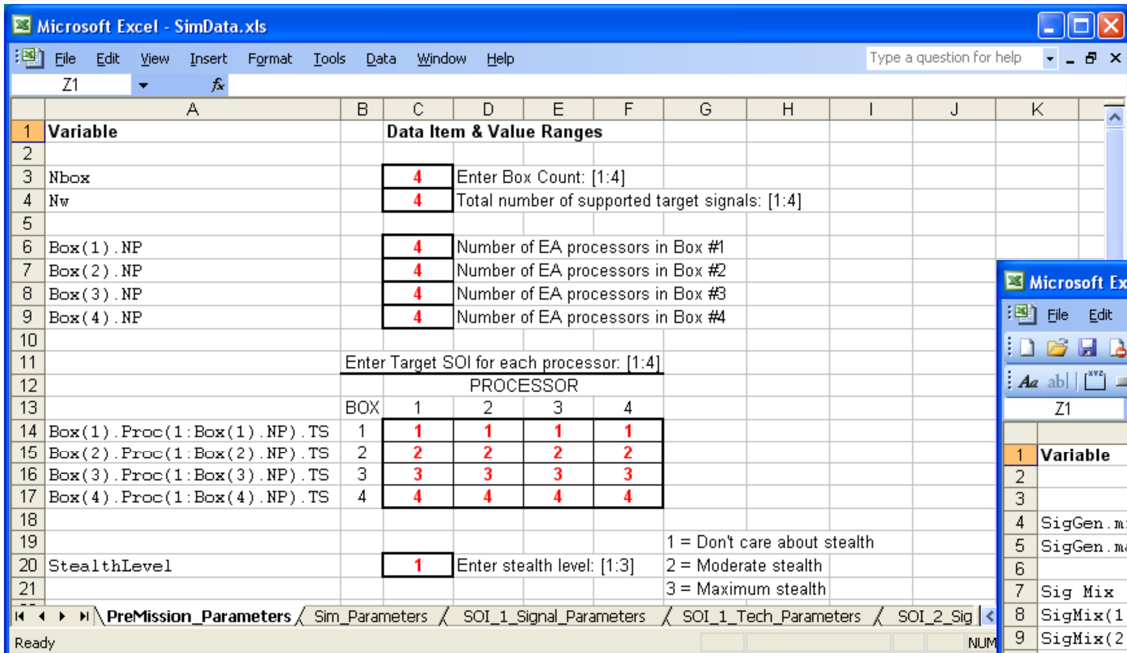
Datasets

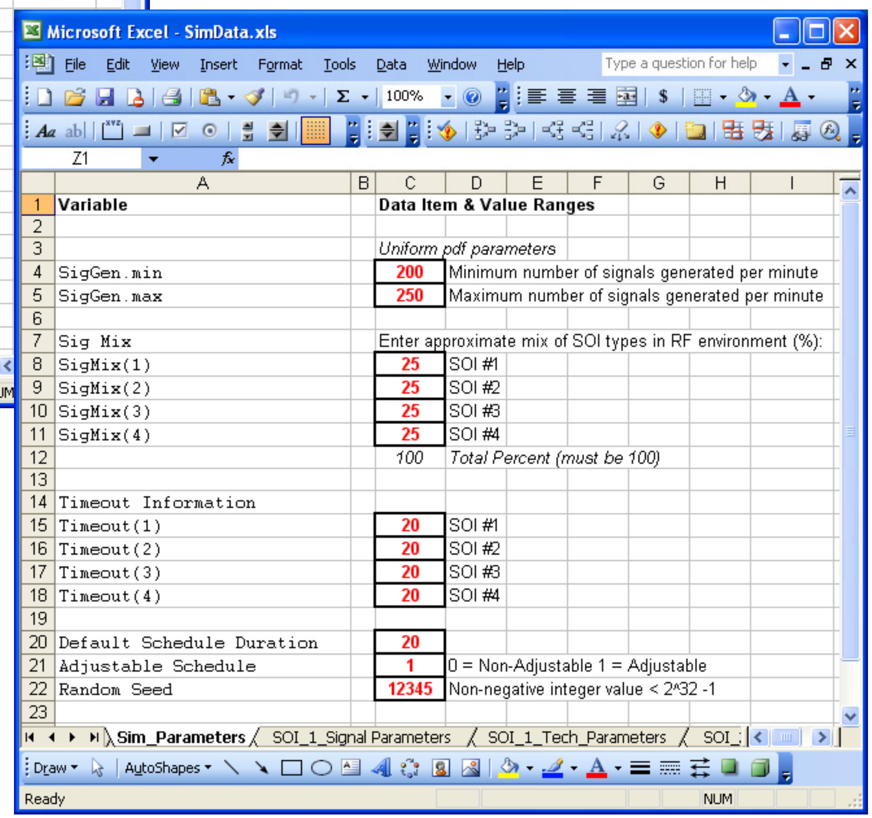
- Multiple datasets created to exercise scheduling algorithms
- Realistic data
  - SimData
- Synthetic data to stress the sim framework
  - NormCondsData
    - SimData with lower dynamic range of Sp
    - Techniques for every stealth level for every SOI
  - FastTimeout
    - SOIs up and down within schedule's dwell
    - Produces large number of SOIs
  - LowSuccessRate
    - Low P_k's
    - More signals survive and show up in subsequent SAs
  - OverloadedSOIs
    - High SOI count
    - Overwhelms the scheduler
  - UnderAllocated
    - Not all processors are allocated
    - Also stresses the scheduler

Control Panel Pre-Run



Parameter Data Entry





ZETA ASSOCIATES

The MathWorks MATLAB&SIMULINK

Signal & Technique Parameters Data Entry

Microsoft Excel - SimData.xls

	A	B	C	D	E
1	**Variable**		**Descriptions**	**Input Values**	
2					
3	TS(1).Nchans			14	Number
4	TS(1).Nusers			2	Number
5	NumHVChan			3	Number
6	HVChan Number			7	High Valu
7	HVChan Value			2	High Valu
8	HVChan Number			5	High Valu
9	HVChan Value			2	High Valu
10	HVChan Number			4	High Valu
11	HVChan Value			5	High Valu
12					

Sim_Parameters | **SOI_1_Signal Parameters** | SOI_1_Tech_Parameters

Ready

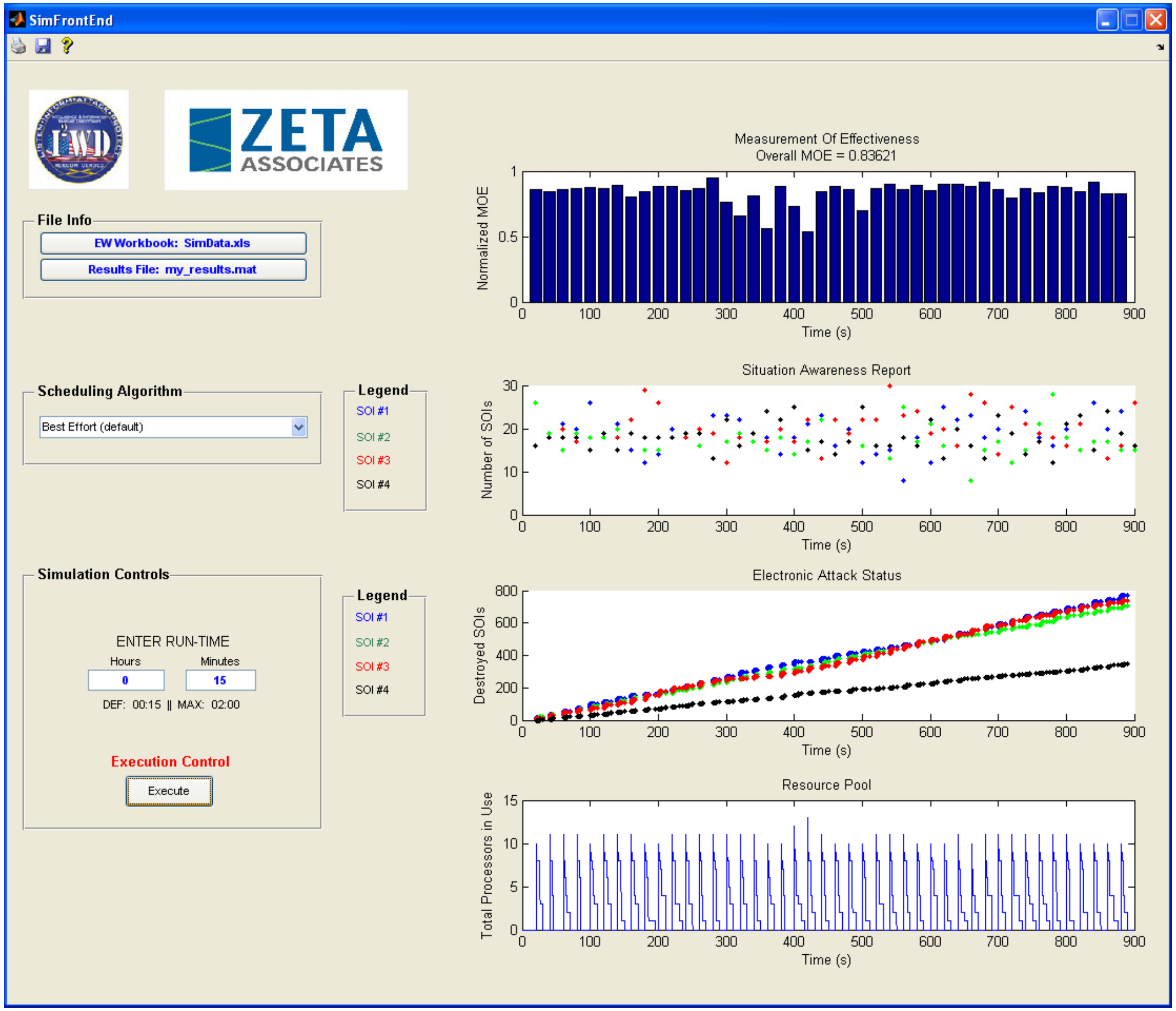
Microsoft Excel - SimData.xls

	A	B	C	D	E	F	G
1	**Variable**		**Descriptions**	**Input Values**	**Notes**		
2							
3	Nz(1)			4			
4							
5	TT(1,1).Sp			34.0E-6			
6	TT(1,1).PercentSuccess			0.90			
7	TT(1,1).StealthMin			2			
8	TT(1,1).MultiTarget			-3	1XN = -3, 1X1 = -2, 1XAll = -1		
9							
10	TT(1,2).Sp			3.0E+0			
11	TT(1,2).PercentSuccess			0.75			
12	TT(1,2).StealthMin			2			
13	TT(1,2).MultiTarget			-3	1XN = -3, 1X1 = -2, 1XAll = -1		
14							
15	TT(1,3).Sp			34.0E-6			
16	TT(1,3).PercentSuccess			0.70			
17	TT(1,3).StealthMin			2			
18	TT(1,3).MultiTarget			-3	1XN = -3, 1X1 = -2, 1XAll = -1		
19							
20	TT(1,4).Sp			34.0E-6			
21	TT(1,4).PercentSuccess			0.60			
22	TT(1,4).StealthMin			2			
23	TT(1,4).MultiTarget			-3	1XN = -3, 1X1 = -2, 1XAll = -1		
24							
25	TT(1,5).Sp						
26	TT(1,5).PercentSuccess						
27	TT(1,5).StealthMin						
28	TT(1,5).MultiTarget				1XN = -3, 1X1 = -2, 1XAll = -1		
29							

PreMission_Parameters | Sim_Parameters | SOI_1_Signal_Parameters | **SOI_1_Tech_Parameters** | SOI_2_Sig

Ready NUM

Control Panel Post-Run



Some Performance Observations

ZETA ASSOCIATES

The MathWorks MATLAB & SIMULINK

Some MOE Performance Data

Data Set	Scheduling Algorithms								
	BE	BEO	Pm| |C_{max}			Pm| |ΣC_j		Pm| |$\Sigma w_j C_j$	
			SPT	WSPT	LPT	ECT	EST	ECT	EST
Fast Timeout	97.8	99.9	98.9						
Low Success	95.6	92.5	64.4						
Norm Conditions	40.4	55.9	55.9						
Over-loaded	81.7	98.2	98.6						
Priorities	99.1	98.0	95.3				97.2	95.3	97.2
SimData	83.6	83.6	83.4						
Under-allocated	43.3	73.0	71.3						

Observations & Possible Solutions

- Some obvious observations present themselves
  - The system can suffer from dead space since the SOI with the longest suppression times will bound the schedule
    - Can EA processors handle more than one SOI to improve techniques coverage?
  - The AIOR is bound by the inability to interleave techniques' bursts
    - What agility can be expected from the Controller?
  - The AIOR variants all behave the same because AI planning always picks the highest rank technique in a memoryless system
    - Employ memory
- More can be found using the sim framework
  - Very multi-dimensional

Performance Bound for Single SOI/Proc

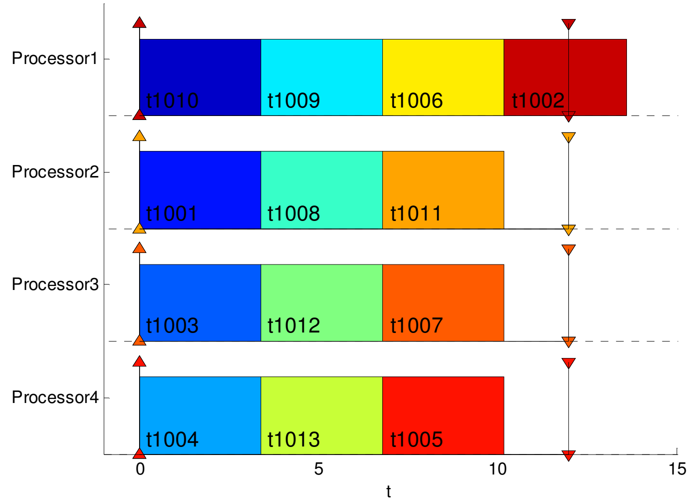
Sim Parameters

- SimData
- SchedDur = 12
- SigGen = [50:100]
- SigMix = 25%/SOI
- Box = 4 x 4
- Stealth Level = 1
- Early AIOR

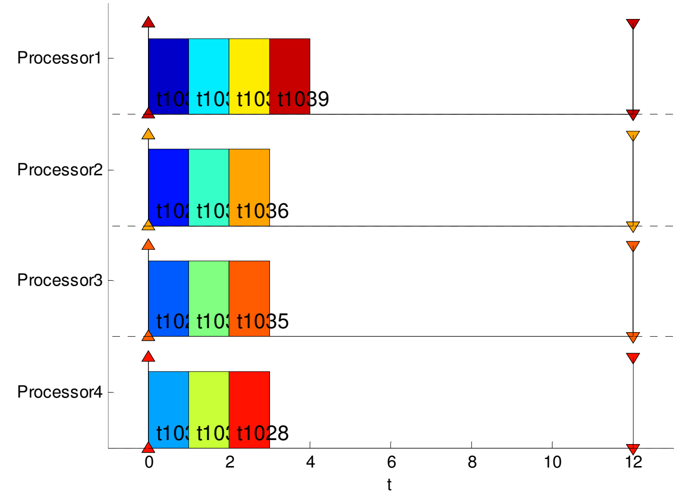
Observations

- Lost EA jobs
  - SOI#1; Proc1
  - SOI#4: Procs 1-3
- Idle Procs
  - SOI#2: Procs 1-4
  - SOI#3: Procs 2-4

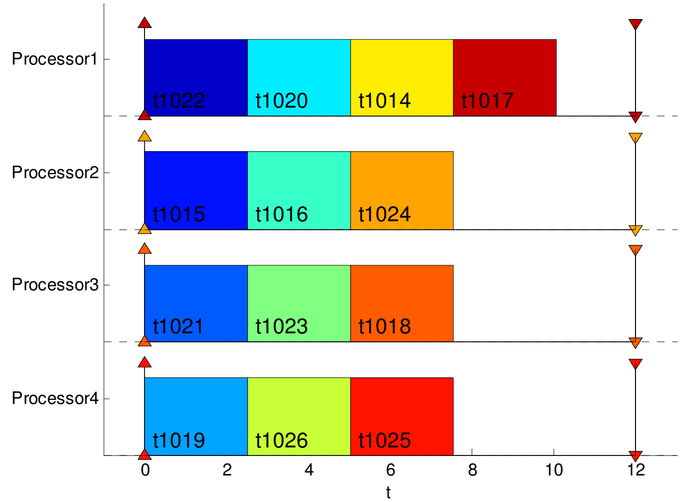
SOI#1



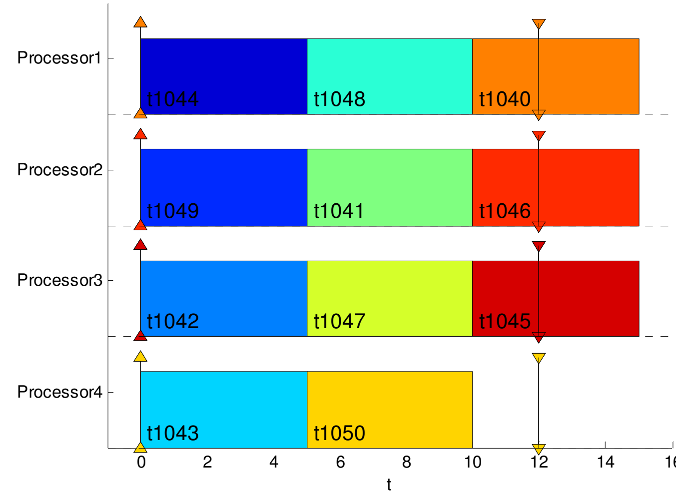
SOI#2



SOI#3



SOI#4



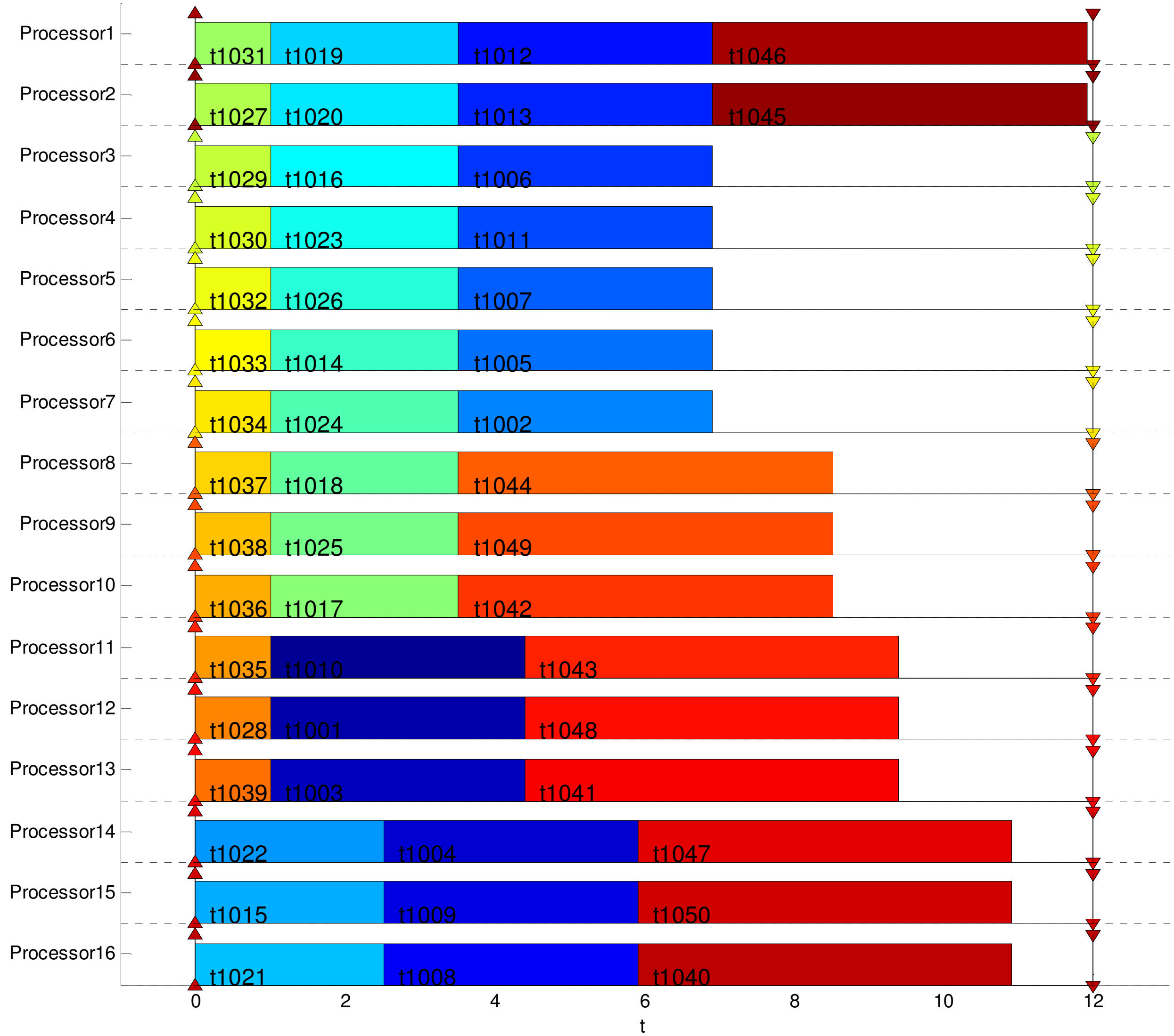
Performance Improvement for Flexible Processor

Sim Parameters

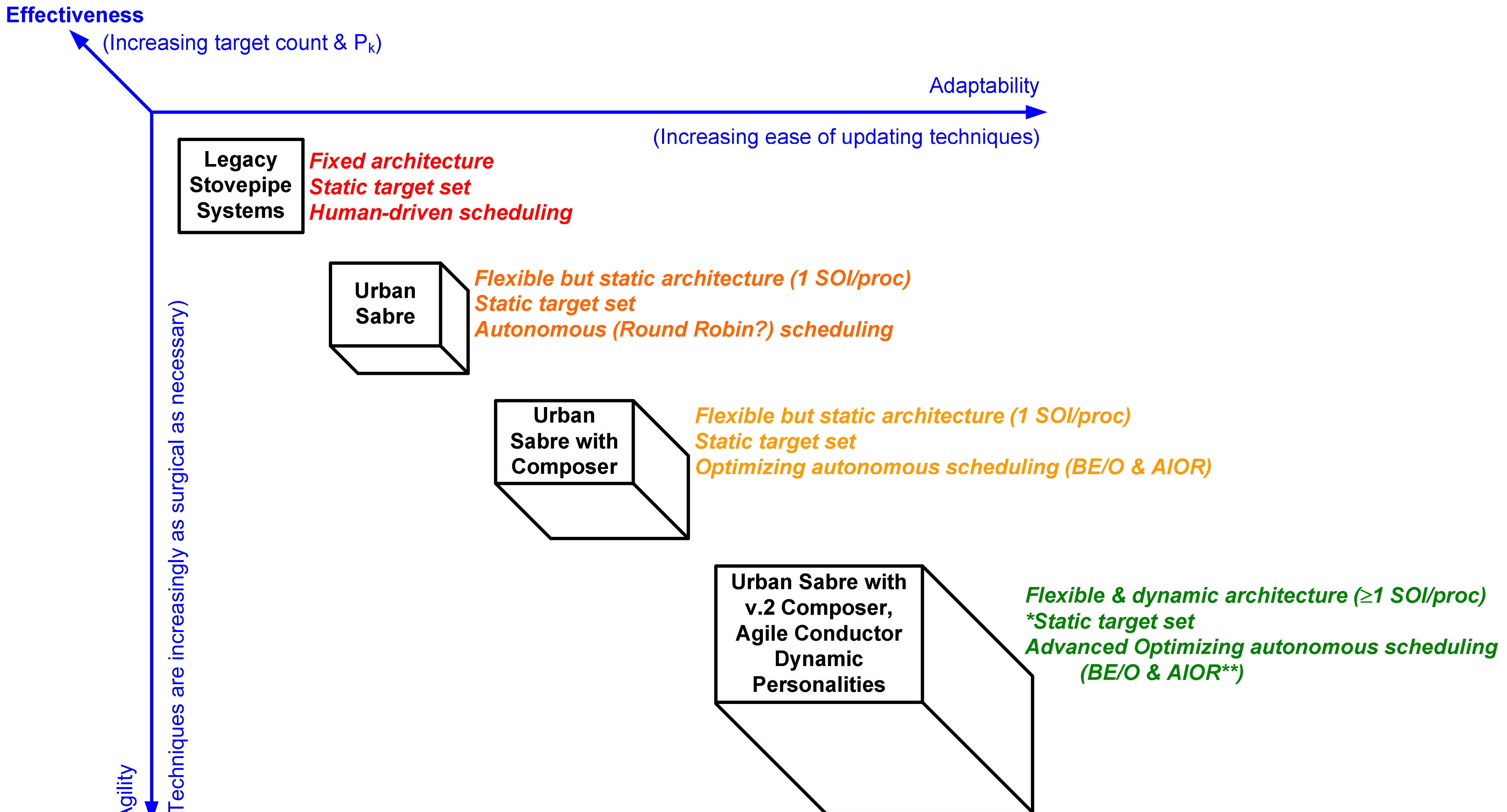
- SimData
- SchedDur = 12
- SigGen = [50:100]
- SigMix = 25%/SOI
- Box = 4 x 4
- Stealth Level = 1
- Early AIOR—BUT all processors are able to prosecute all four SOIs

Observations

- NO Lost EA jobs
- Still have some idle processors
  Procs 3-13



Road Map for Next Gen EW Scheduling



*Could be reprogrammable over-the-air per JTRS/SDRF
**Increase technique density via interleaving bursts

Conclusion

Some M&S Lessons Learned

- The nits will always get you
  - Whether you roll your own framework
  - Or build on someone else's framework
- Pay attention to dynamic range
  - Numerically
    - ILP limitations using TORSCHE (use other ILP solvers?)
  - Temporally
    - When some techniques are measured in μs and some in tens of seconds, it can be a problem
- Read the fine print—especially w.r.t. freeware
  - Did not catch that TORSCHE could not run in a M/L block inside Simulink
    - EML must be used in M/L blocks
    - TORSCHE is NOT; it's object based (∴ utilizes dynamic variables) and cell arrays
  - Fortunately, ∃ workaround to run the TORSCHE-based AIOR routines in the simulation
    - EML routine that calls TORSCH-based AIOR routines is **`Scheduling.m`**
    - **`Scheduling.m`** must declare AIOR #*n* as an EML extrinsic routine ***before*** calling AIOR #*n*: **`eml.extrinsic('AIOR`*n*`')`**
  - When going to prototype C/C++ equivalents of TORSCHE routines will be required.
- When in doubt…
  - Call MathWorks tech support for help; don't waste time spinning your wheels

Wrapping Up…

- Summary
  - EW M&S
  - Lessons learned
- Any more questions???
- Thanks for attending!
  - We hope you found this very helpful
- Contact info:
  - Email: Janka-Randall@zai.com
    - Or use janka@ieee.org; bounces to Zeta
  - Office: 703-272-1052