



Complex Event Processing and U.S. Surface Navy Use Cases

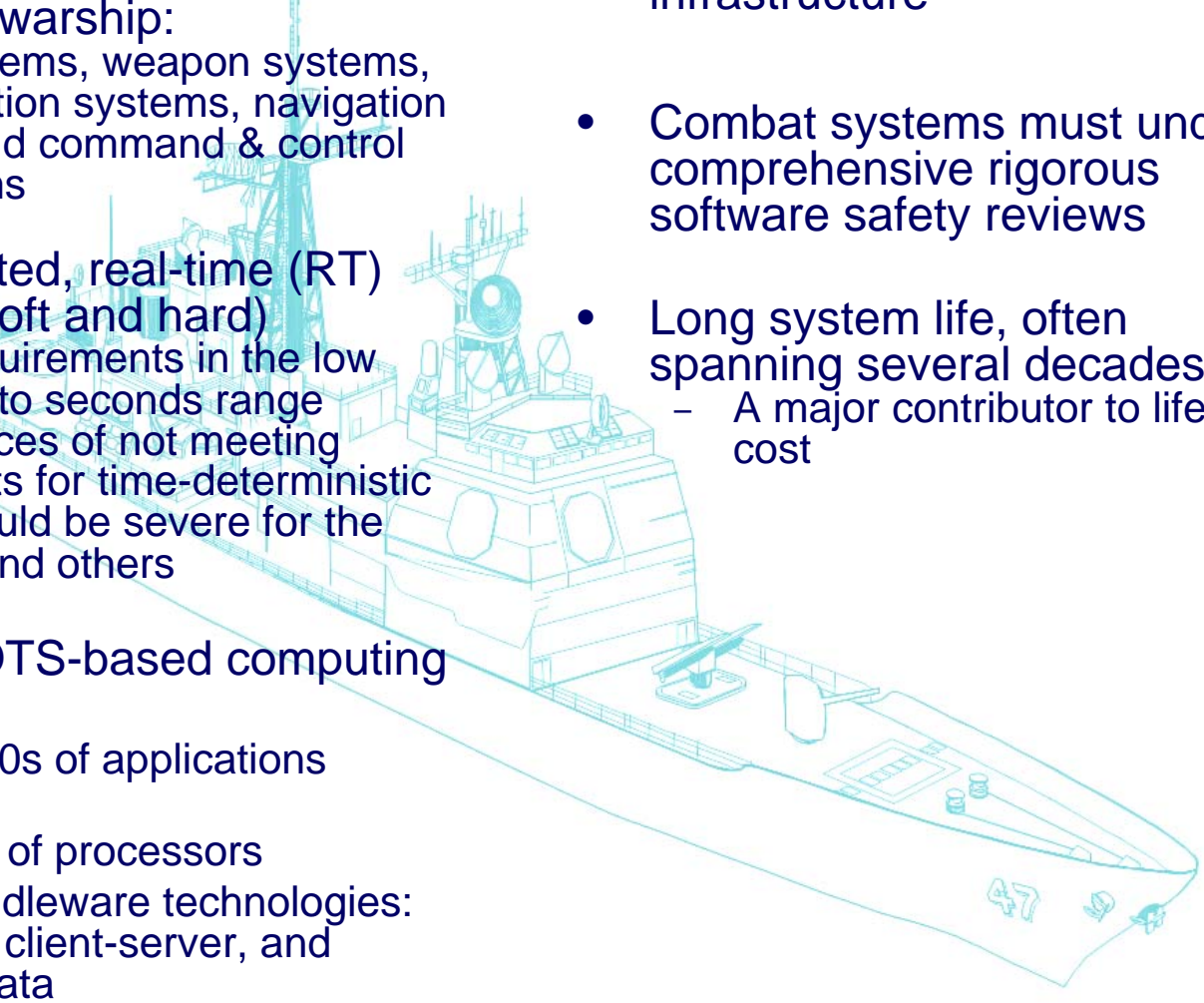
Paul V. Werme, NSWCDD

Paul A. Haynes, UK MoD Exchange Scientist to NSWCDD

Nathan J. Rodecap, NSWCDD

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

Surface Navy Combat System Characteristics

- The set of human and machine resources that comprise the fighting capability of a warship:
 - Sensor systems, weapon systems, communication systems, navigation systems, and command & control (C2) systems
 - Large, distributed, real-time (RT) applications (soft and hard)
 - Latency requirements in the low millisecond to seconds range
 - Consequences of not meeting requirements for time-deterministic behavior could be severe for the warfighter and others
 - Networked COTS-based computing environment
 - 100s to 1000s of applications (processes)
 - 10s to 100s of processors
 - Multiple middleware technologies: messaging, client-server, and persistent data
 - Fault tolerant applications and infrastructure
 - Combat systems must undergo comprehensive rigorous software safety reviews
 - Long system life, often spanning several decades
 - A major contributor to life cycle cost
- 

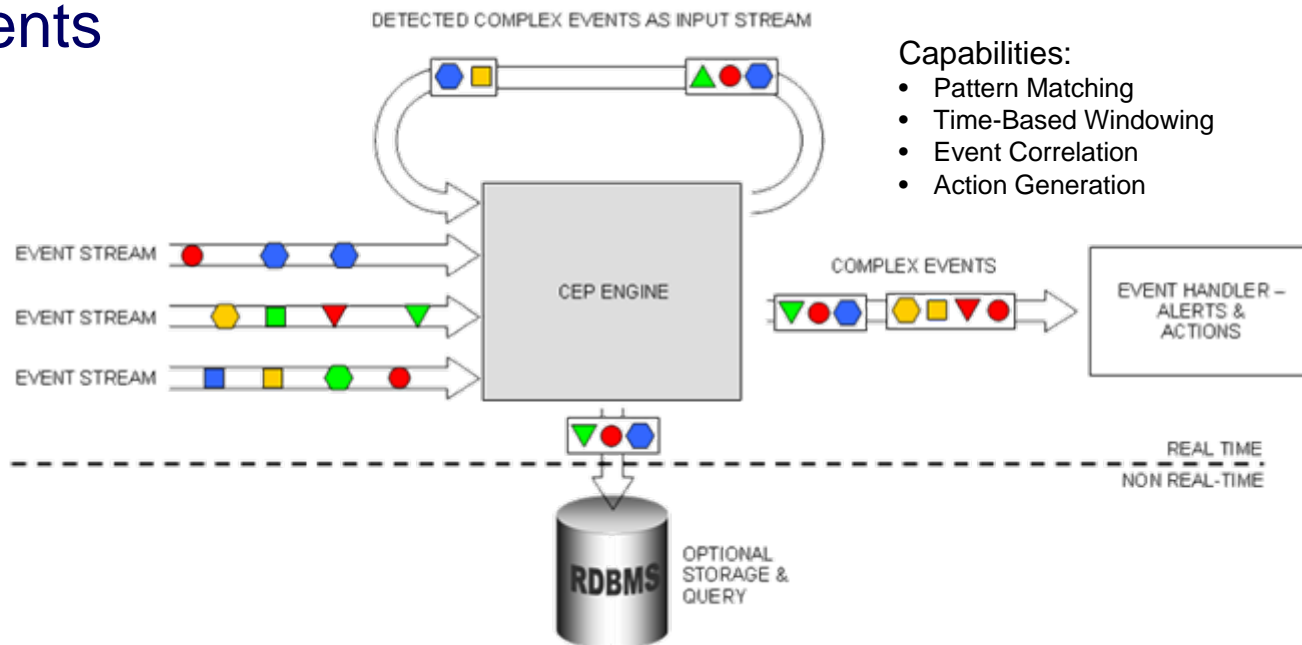
The Problem

- Combat systems generate large volumes of data, often at high rates, which is aggregated, correlated, assessed, and acted upon at run-time
 - Much of this functionality is accomplished via hard-coded logic
- Can we get more out of this raw data, opening up enhanced capabilities?
 - The difficulty lies in sorting through the data and putting the pieces together to form actionable information
- Traditional data correlation techniques follow the store-then-query paradigm
 - This doesn't scale efficiently to applications on the scale of combat systems
 - Can't deliver real-time performance

Complex Event Processing (CEP)

- Promises real-time/run-time correlation of high-throughput data
- An *event*: a state change of note
- A *complex event*: an event that cannot be directly measured but whose occurrence can be inferred from the logical and temporal relationships of multiple simple events

1000s msgs/sec
10s of data feeds
100s of rules



Capabilities:

- Pattern Matching
- Time-Based Windowing
- Event Correlation
- Action Generation

CEP Technology Landscape

- Commercial interest in Event-Driven Architecture and Service-Oriented Architecture is driving investment in CEP
 - “...SOA enables deep, real-time business intelligence and CEP is the technical mechanism for realizing this benefit.” — ZapThink LLC
 - 10+ companies have entered the marketplace in the last few years
 - Application Server vendors are developing CEP capabilities within their products
 - Smaller firms are offering CEP engines for integration with customer IT architectures
 - Marketplace still being fought over
 - Divergent approaches (SQL based, rule based, proprietary language based, GUI based)
 - Standardization way ahead unclear until a winner emerges
- Between 1998 and 2004, NSWCCD (via HiPer-D, DARPA Quorum, and OA) prototyped and actively advocated the development of commercial instrumentation data correlation engines
 - Commercial vendors capable of developing products were found
 - But, there was no business case
 - ***Various markets have now created the business case!***

CEP Technology Take-up

- CEP products are at an advanced stage of maturity and take-up
 - Financial industry (fraud detection, algorithmic trading)
 - Telecommunications industry (network monitoring, intrusion prevention/detection)
- CEP is more than the latest technology fad



CEP Technology Providers

- **Commercial Vendors:**

- StreamBase
- Coral8
- SENACTIVE InTime
- TIBCO BusinessEvents
- Progress Apama
- Aleri Streaming Platform
- Aptsoft Director / IBM WebSphere Business Events
- Cognos IBM (acquired Celequest)
- Syndera Real-time Business Intelligence Suite
- Skyler C3
- Vhayu Velocity
- Agent Logic
- GemStone GemFire
- SeeWhy
- Avaya Event Processor (was iSpheres)
- IBM Haifa AMIT
- BEA WebLogic Event Server
- RTI Event Processing (DDS & Coral8 integration)

- **Open-Source Products:**

- Esper
- NEsper

- **Free-ware Products:**

- StreamCruncher

- **Consortiums / Working Groups:**

- Event Processing Technical Society (EPTS)
 - Event Processing Reference Architecture Working Group (EPRAWG)
- OMG SOA Special Interest Group

Potential Naval Applicability

- CEP has potentially very broad applicability throughout the combat system, at the force level, and into the GIG
- Possibly, eventually all the way to tactical functionality
 - Not a great leap of imagination from detecting online fraud to detecting indicators and warnings of asymmetric attack
 - Extreme example: IBM Haifa presentation on Anti-Ship Missile Defense advocates re-architecting the combat system by using CEP as the middleware backbone and implementing tactical functionality within the CEP ruleset
- The Navy needs to be an intelligent customer when CEP products are offered for use within our systems



NSWCDD CEP Technology Evaluation Approach

- Independent assessment of the appropriateness and limitations of CEP technologies for our data correlation problems
- Determination of the viability of an open and reusable CEP engine approach
 - Demonstrate approach to avoid lock-in to problem-specific products
- Initial investigation and prototyping of functionality with near-term transition potential to Programs of Record
 - Quality of Service management
 - Fault detection / fault isolation
 - Health and status monitoring
 - Readiness assessment
- Assessment of COTS technologies
 - Application Server vendors' products have potential for integration between the combat system and Enterprise domains (IBM, BEA)
 - Standalone products are optimized for various purposes (e.g., ease of use, performance, ease of integration); potential for less disruptive integration

CEP Technology Assessment

- **Real-time performance:** throughput, latency, and determinism
- **Architectural scalability:** number of data feeds, number of messages, number of rules, and query complexity
- **Fault tolerance:** architectural suitability of the fault recovery approach, impact of individual points-of-failure on data correlation stability, correctness, and effectiveness
- **Compatibility and conformance with existing and emerging architectures:** capabilities that can viably be incorporated into the transition target system architectures
- **Architectural suitability of the technologies:** Is the technology a good fit for each domain capability? Does the technology meet goals of openness, reusability, and portability? Does the technology result in undesired system architectural constraints and limit design alternatives? To what degree does the technology require bridging and translation layers?

Opportunities

- Where is CEP a good fit?
 - Low risk:
 - Rule-oriented processing
 - Policy-based processing (supports best practice of separation of policy and mechanism)
 - Stateless or small well defined state models
 - Areas where mechanisms are well defined but policy (for using the mechanisms) is expected to change
 - Higher risk:
 - Mathematically or algorithmically intensive calculations
 - Large, complex, and/or intertwined state models

- Where would CEP be applicable and beneficial?
 - For each Use Case, assess:
 - number and complexity of events
 - number and complexity of data streams and required Adapter logic
 - complexity of data correlation logic
 - CEP is a good answer for a specific range of data correlation problems
 - not needed for very simple problems (where hand-coding would be appropriate)
 - not the right answer for many complex problems (based on current state of the art)
 - Primary interest is CEP capabilities supported by multiple vendors (to avoid product lock-in)

- Near-term transition potential (< 5 years to build, test, and field capabilities)
 - Computing Infrastructure Monitoring and Alerting
 - Minimize impact on system architecture by using:
 - existing application middleware interfaces
 - existing HW / OS monitoring interfaces
 - additional application-internal instrumentation only if needed

- Longer term transition potential
 - Operational (Tactical) Functionality
 - Computing Infrastructure Control
 - Higher Risk with Strict V&V requirements

Benchmarking and Evaluations

- Performance Testing:
 - Evaluate throughput, latency, and determinism
 - Vary:
 - # Rules (10's to 1000's)
 - Complexity of rules (Simple to Multiple Events/Multiple Joins)
 - # Rules triggered by each event (1 to 10+)
 - #, rate, and complexity of events
- Ease of Use Evaluation:
 - Two scenarios:
 - Simple doctrine processor (~10 simple rules)
 - More complex QoS monitoring scenario
- Ease of Integration Evaluation:
 - Criteria:
 - Availability of adapters
 - Complexity of using adapters
 - Complexity of adding adapters
 - Adapters to assess: DDS, files, sockets, WSDL, ODBC/JDBC
- Fault Tolerance Evaluation:
 - Criteria:
 - FT models supported
 - Architectural impacts and complexity of implementation
 - Failover latencies

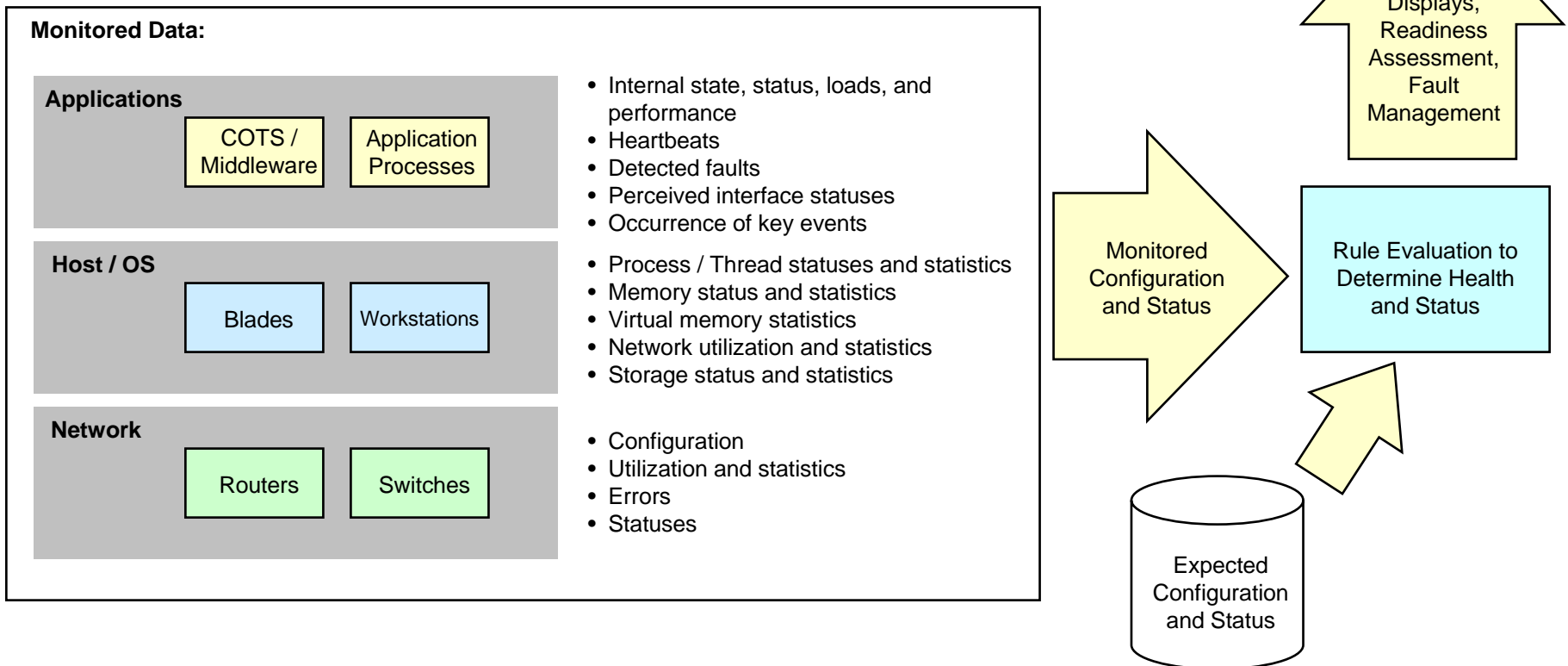
CEP Use Cases

- Use Case Categories:
 - Benchmarking
 - *Performance Evaluation*
 - *Ease-of-Use Evaluation*
 - *Ease-of-Integration Evaluation*
 - **Fault Tolerance Evaluation**
 - Infrastructure Capability Demonstrations
 - **Syntactic and Semantic Conversions**
 - *Computing Infrastructure Health and Status Monitoring*
 - **Fault Detection / Fault Isolation**
 - **Readiness Assessment**
 - **Run-Time Validation**
 - Operational Capability Demonstrations
 - *Doctrine Processing*
 - *Checkpoint Monitoring*
 - **Sensor Data Fusion**
 - **Determination of ID and Intent**
 - **Sensor Grid Monitoring for Riverine Operations**
 - Capability Boundary Evaluations
 - *End-to-End Timeline Monitoring and Management*

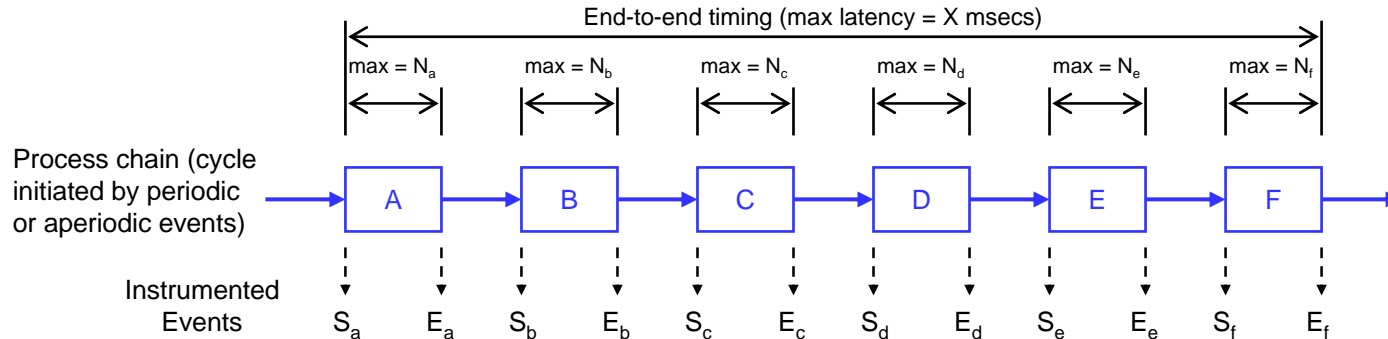
• Highlighted Use Cases are being worked in FY08 for demo purposes and to provide evaluation metrics

Infrastructure Use Case: Health and Status Monitoring

- Provide rolled-up hardware and software status and health information
- Leverage existing status, performance, load, fault, and state data provided within the system
 - reported by applications, hardware/operating systems, and network switches/routers
- Leverage historical data (hours or days) to support “drill-down” capabilities
- Benefits of CEP approach:
 - flexibility of the rule set
 - ability to input different status reporting formats and information from different classes of components



Infrastructure Use Case: End-to-End Timeline Monitoring



Event Format:

- string EventID
- double Timetag

Note: There is no defined "cycleID" in the event!

Example Event Sequence:

- EventID == "Process_A_Start", Timetag == 192.270
 - EventID == "Process_A_Stop", Timetag == 192.282
 - EventID == "Process_B_Start", Timetag == 192.284
 - EventID == "Process_B_Stop", Timetag == 192.289
 - **EventID == "Process_A_Start", Timetag == 192.290**
 - EventID == "Process_C_Start", Timetag == 192.292
 - EventID == "Process_C_Stop", Timetag == 192.294
 - EventID == "Process_D_Start", Timetag == 192.296
 - **EventID == "Process_A_Stop", Timetag == 192.300**
- ... }

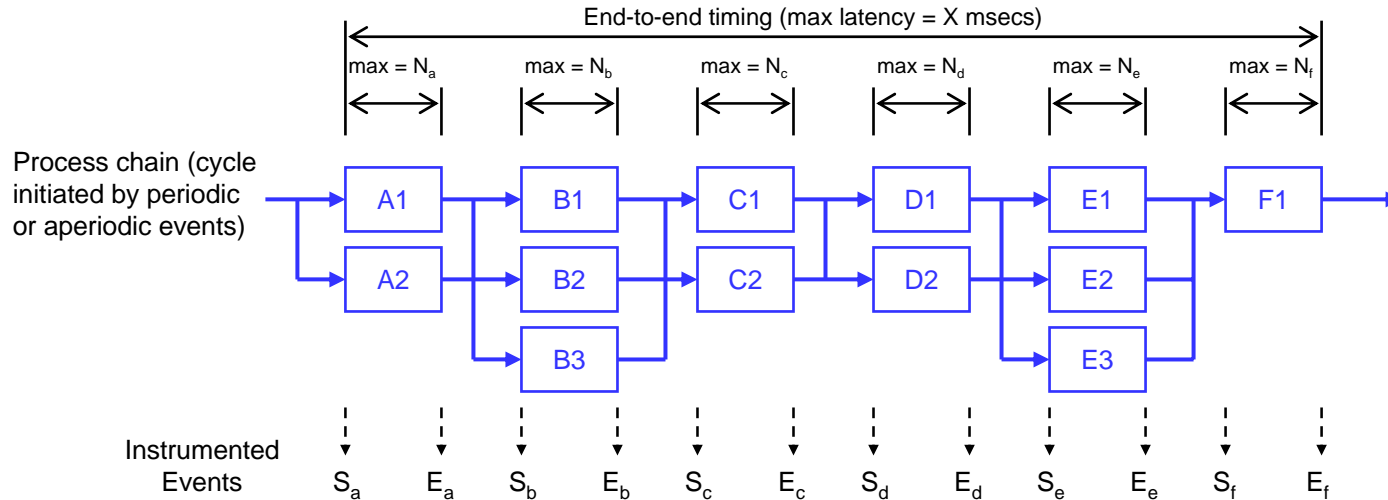
• For an End-to-End RT Processing Path:

- 1 or more processes make up the path (this is a fixed number for each defined path)
- End-to-End max latency is defined for the path
- Usually process-level internal time budgets are also defined
- The message processing approach within each process of the path will be known:
 - async msg processing (network delay only)
 - async msg processing with a max internal queuing delay
 - synchronous msg processing with max polling delay
- Overlapping cycles are allowed, however, at the process level, processing for the current cycle will always complete prior to processing of the subsequent cycle

• Goals:

- Detect end-to-end timing faults and sub-path timing faults and provide alerts when faults are detected with minimal latency
- Detect trending indicating expected future end-to-end timing faults and sub-path timing faults and provide alerts with minimal latency

Infrastructure Use Case: End-to-End Timeline Monitoring



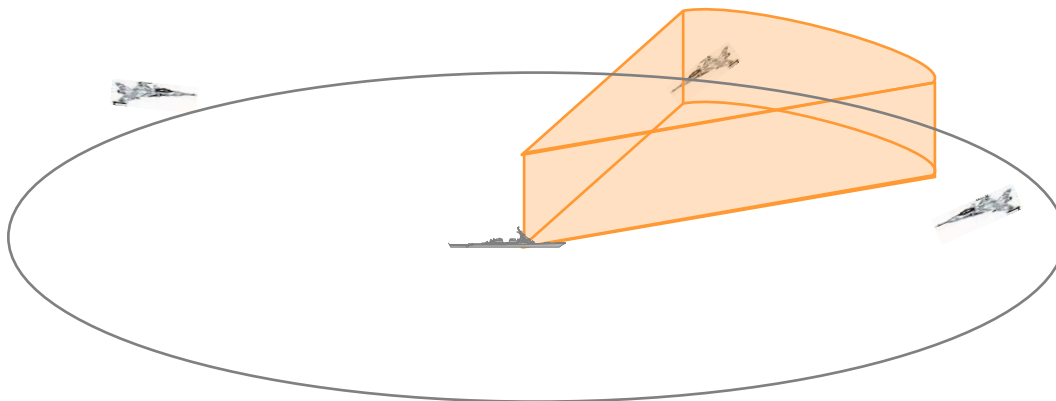
Event Format:

- string EventID
- string InstanceID
- double Timetag

- Use Case assesses support for combinations of Surface Navy domain "hard" issues:
 - correctly correlating a set of path data that does not contain explicit cycle or transaction ID information
 - handling replicated processes within the path
 - supports survivability and possibly load-balancing
 - fault tolerance approach and role for each process will be known
 - e.g., passive, semi-active, active, ...
 - handling out-of-order event receipt
 - handling unreliable event receipt (i.e., lost events)

Operational Use Case: Doctrine Processing

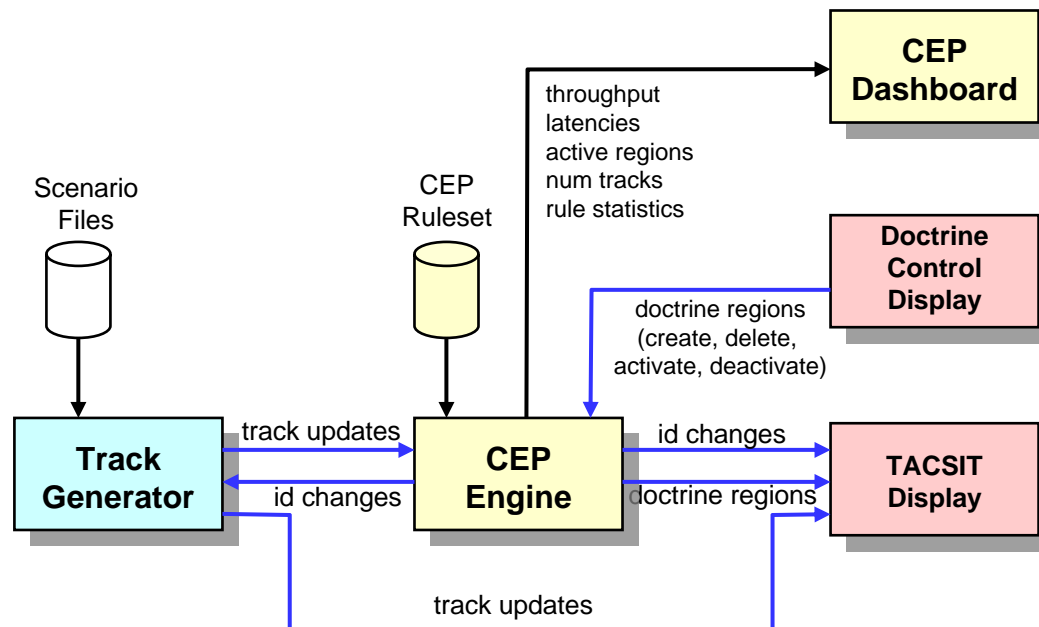
- Simple rules for track identification and engagement recommendations
 - Region criteria:
 - Sectors:
 - relative to Ownship or a reference point
 - min/max range and min/max altitude
 - Polygons:
 - fixed location or relative to Ownship or a reference point
 - lat/lon or x/y points, optional min/max altitude
 - Kinematic and attribute criteria:
 - Speed, closing rate, track category (e.g., air), current track ID (e.g., unknown), etc.



Example: Engagement Doctrine
IF (in_region MissileEngage)
 AND (closing_rate > 300 kts)
 AND (track_id <> Friend)
THEN Engage

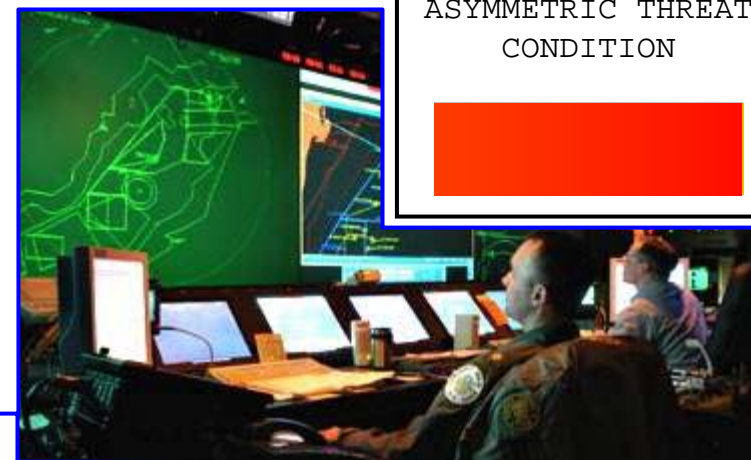
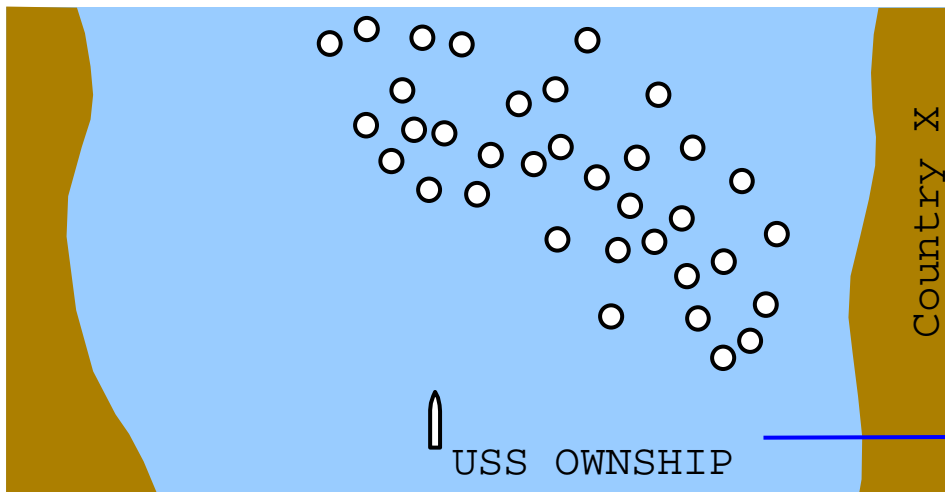
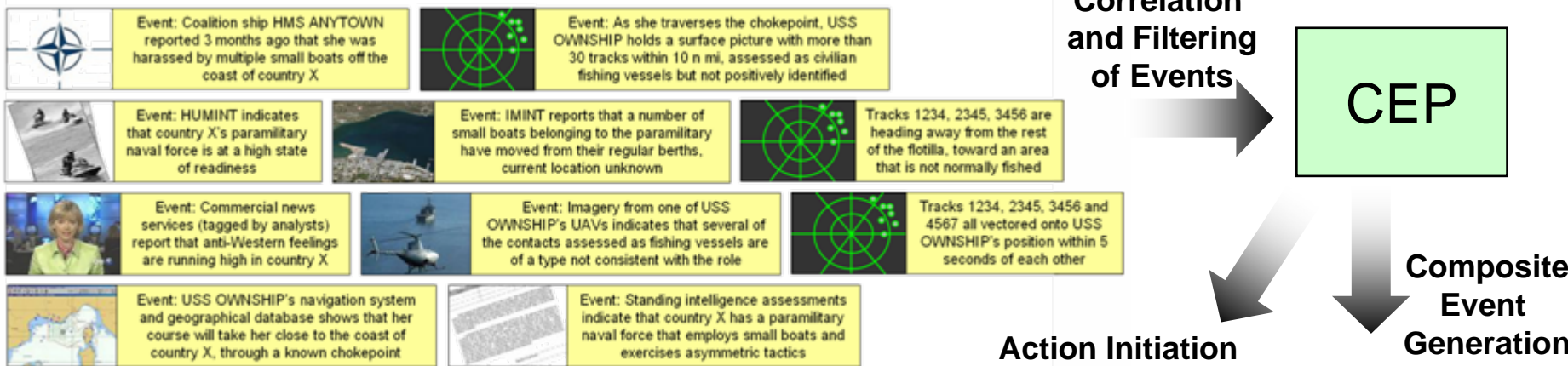
Doctrine Processing Prototype

- Instrumentation: *Track Generator*, *TACSIT*, and *Doctrine GUI*
 - latency and throughput measurements
- Ability to capture *Track Generator* and *Doctrine GUI* outputs to files
 - provided to CEP vendors for off-line playback / testing
- Provides functional test for run-time manipulation of CEP rules
 - creation, deletion, activation, deactivation of rules
 - ability to create and change parameterized rules



Future Investigation: Determination of ID and Intent

- Scenario: Ownship is operating within a congested surface picture, comprised mostly of civilian fishing vessels. Hostile units may be using commercial vessels as cover for an asymmetric attack



CEP Standardization Landscape

- General observations:
 - CEP vendors currently do not see a strong business case for standardization
 - Standardization of a common CEP correlation grammar does not appear to be likely in the near-term
 - Diverse range of approaches are being used; valid tradeoffs among the approaches
 - A common language across CEP approaches is an unsolved (and unworked) research problem
 - A common SQL-based grammar among SQL-based CEP products would appear to be the only near-term possibility
 - There is a significant commonality at the CEP Adapter level in regard to event definition and how event data is read, written, and normalized

- Recommendation:
 - **In the near term, focus on standardizing interfaces, APIs, and common event semantics at the CEP Adapter level**
 - Benefits:
 - Customers (and potential customers) could design their systems in a manner that does not lock them architecturally to a specific product
 - Within DoD, standards compliance facilitates adoption of specific technologies and products
 - Issue: Availability of CEP Adapters and ease of integration with specific middleware technologies is a differentiator among CEP products

Summary

- The potential benefits of data correlation capabilities and technologies are recognized within the U.S. Surface Navy
- CEP technology is commercially available at a level of maturity and market acceptance that makes it a promising technology option
 - Investigation of the capabilities, limitations, and viability of CEP technology for use within the combat system domain is ongoing
- Promising opportunities exist for applying CEP technology within U.S. Navy combat systems
 - Near-term standardization of CEP capabilities would make this even more attractive