

# Node Information Services – Joint Processing Patterns for the C2 Enterprise

**Bob Wilson**

**781-271-4867 • [bwilson@mitre.org](mailto:bwilson@mitre.org)**

**Air Force MOIE**

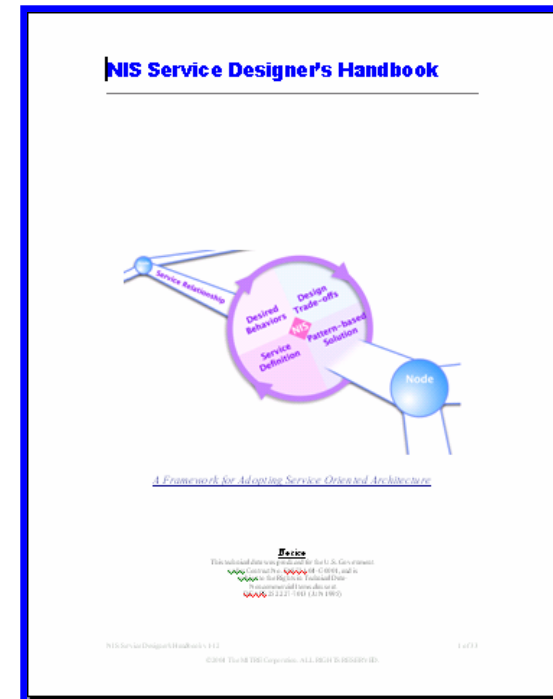
**MITRE  
Technology  
Program**

# Problem

- Integration across organizational entities is our most challenging unsolved problem when considering critical processes that must jointly manage context and state.
- What are the Joint Process patterns that provide the Air Force with competitive advantage and when are they appropriate?

# Background

- Service oriented architecture (SOA) requires a complete set of practices from design to production in order to be successful.
- NIS MOIE FY04 work captured DOD domain-specific integration options in patterns for knowledge re-use in order to build a single service methodically .
- Now we are developing comprehensive SOA strategies and agents to cover the complete service portfolio life cycle.
- This addresses timely, cost-effective, and value-based adoption of SOA capabilities.



**MITRE**

© 2005, The MITRE Corporation

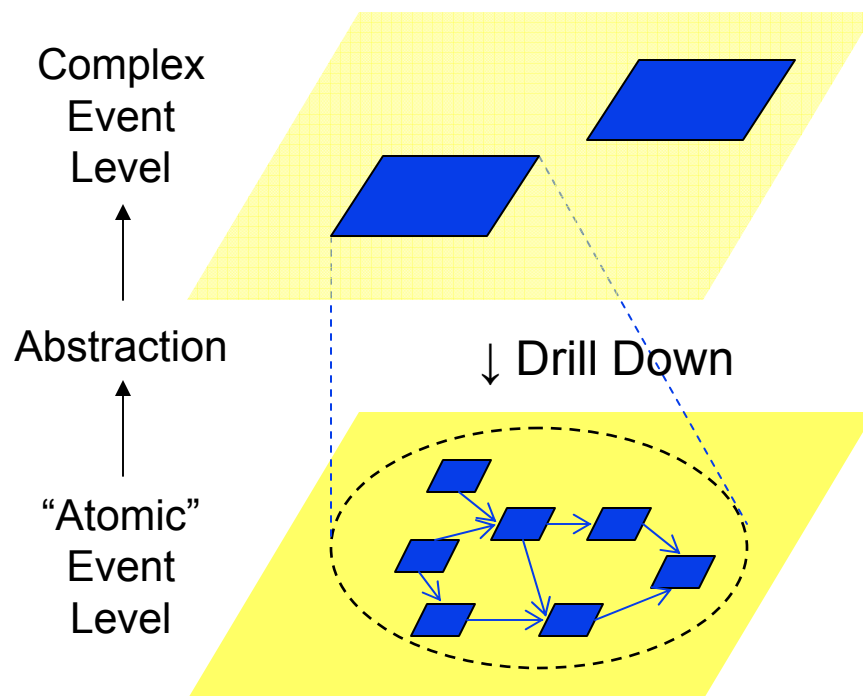
# Objective

- Define complex processing patterns for reuse: ACID, LRT, adaptive activity, events, disconnected and hybrid
- Provide context for operational success
- Accelerate the adoption of advanced processing models that provide competitive advantage
- Influence Navy (RAPIDS), Joint Programs (GCCS FoS), and DISA (NCES/GIG-ES)
- Impact Enterprise Engineering's key emerging issues

# Activities

- **Joint Processing focuses on the types of stateful integration patterns and the tools to adopt them.**
- **The NIS team identified six major areas of state or “context management,” which are:**
  - **Service-Based ACID Transactions**
  - **Long-Running Transactions**
  - **Hybrid Joint Processing**
  - **Disconnected Transactions**
  - **Adaptive Stateful Activities**
  - **Complex Event Processing**

# Highlight: Complex Event Processing

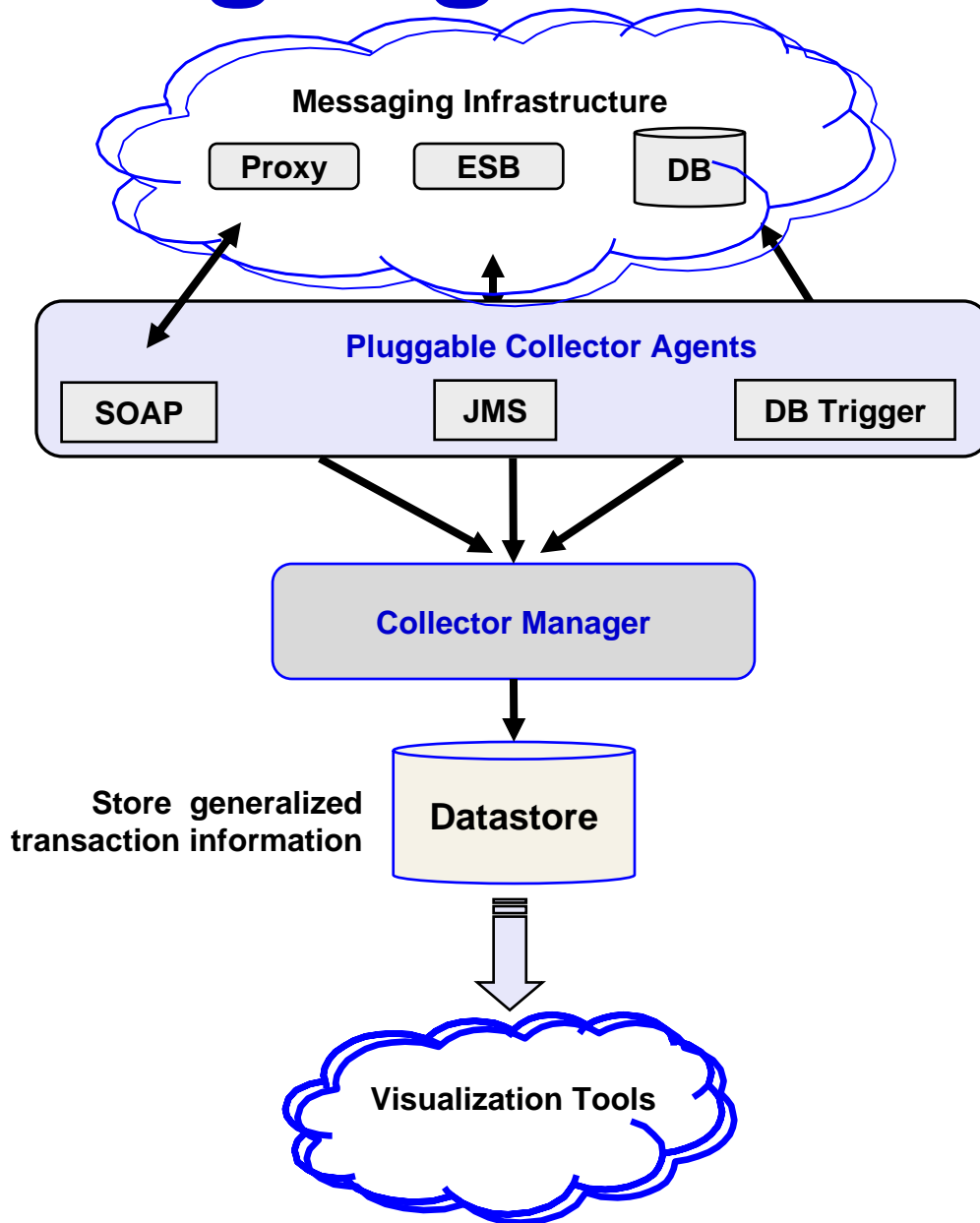


**Complex Event Processing (CEP)** is a new approach to information management that shifts the focus away from conventional procedural logic applied to data and toward analysis of recorded events and their causes.

Events objects are associated via causality pointers, forming causal tracks.

When specific complex events are defined, they may be predicted when a growing number of their prerequisite events transpire.

# Highlight: Collector Agents

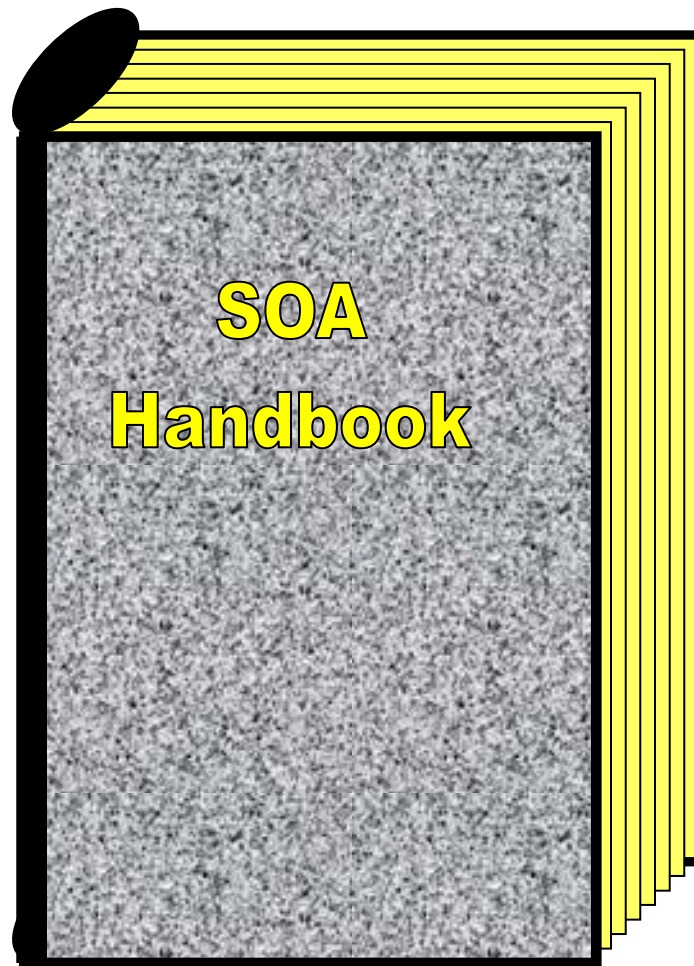


- Intelligent agents can be used to collect service transactions and monitor boundary conditions across various messaging infrastructures.
- HTTP proxies, database triggers, and event monitors implement a common agent architecture for data collection and transcription.
- Individual agents communicate with a centralized manager that provides business logic for collecting data elements and mapping core integration points.

# Impacts

- Provide practices for service engineering deliverables over program life cycles
- Provide working agents for building “real-time-as-is” dynamic service management map
- Provide “decision trees” for complex system integration problems
- Provide management processes for operational service life cycle
- Provide service reuse with pattern library

# Future Plans



- ✓ Service Engineering Deliverables
  - ✓ Instrumentation Agents
    - ✓ Design Strategies
      - ✓ Service Management
        - ✓ Service Design Process
          - ✓ Patterns for Reuse