



Can Real-Time Operate in an SOA Environment?

Precision Strike Technology Symposium
Charles Kille, Raytheon
John Link, VOLVOX, Inc.
October 23-25, 2007

Outline

- Real-Time in the Net-Centric/ SOA Transformation Environment
- The Net-Centric/SOA Paradigm
 - Network and Net-Centric Definition
 - SOA Definition
- The Real-Time Paradigm
- When Paradigms Collide
- Recommendations
 - Policy
 - Network Architectures
 - Technical Solutions
 - Policy & Culture
- Summary

Our Question...

- Started with the abstract question: ***Can “Real-Time” Operate in a Service Oriented Architecture (SOA) based Operational Environment?***
- Refined the question to: ***Is Real-Time part of the Net-Centric/SOA Transformation Environment?***

Short Answer is, Yes... It must be, but there are issues...

As DoD moves forward with Net-Centric Transformation focusing on SOA and shared services as architectural choices -- significant concerns remain for the DoD enterprise and its mission-critical timing-sensitive needs, and for the Real-Time Community

**Policy, Architecture & Technology Must All Support
Net-Centric/SOA Transformation & Mission Critical
Real-Time Operations**

Net-Centric/SOA Concerns

To prepare the ground to examine the Real-Time Community concerns about Net-Centricity/SOA, let's first clarify some related terms:

- Network-Centric

- Net-Centric

- SOA

Definition: Network-Centric

Network-Centric Warfare: “NCW relies on computer processing power and networked communications technology to provide a shared awareness of the battlespace for U.S. forces.”

Network Centric Warfare: Background and Oversight Issues for Congress CRS, June 2004

Network-Centric Warfare worked to aggregate existing “stove-piped” networks and applications at multiple operations centers to facilitate C2 joint forces through information superiority.

...Often called by Warfighters “Swivel Chair Integration”

Architecturally, Network-Centric systems are available to Commanders and analysts, separate but collocated, and primarily accessible in a Tactical Operations Center setting.

Definition: Net-Centric

- ***Net-Centricity is an “information sharing strategy” promoting:***
 - Secure connectivity and interoperability
 - Common technical standards
 - Common data and meta-tagging standards

- **Net-Centricity builds on the Network-Centric approach**
 - Net-Centricity leverages and extends connectivity and access to provide a much greater level of integration of services, information and interoperability -- across the Battlespace
 - Net-Centricity essentially mimics the seamlessness of the Internet solution space

Definition: SOA

Service-Oriented Architecture (SOA) involves:

- “(SOA is) the policies, practices, and frameworks that enable distributed application functionality to be provided and consumed as sets of services. Services in SOA are published, then discovered and invoked by service consumers at appropriate granularity levels and are abstracted away from the implementation using a standard-based interface definition to produce effects consistent with measurable preconditions and expectations”

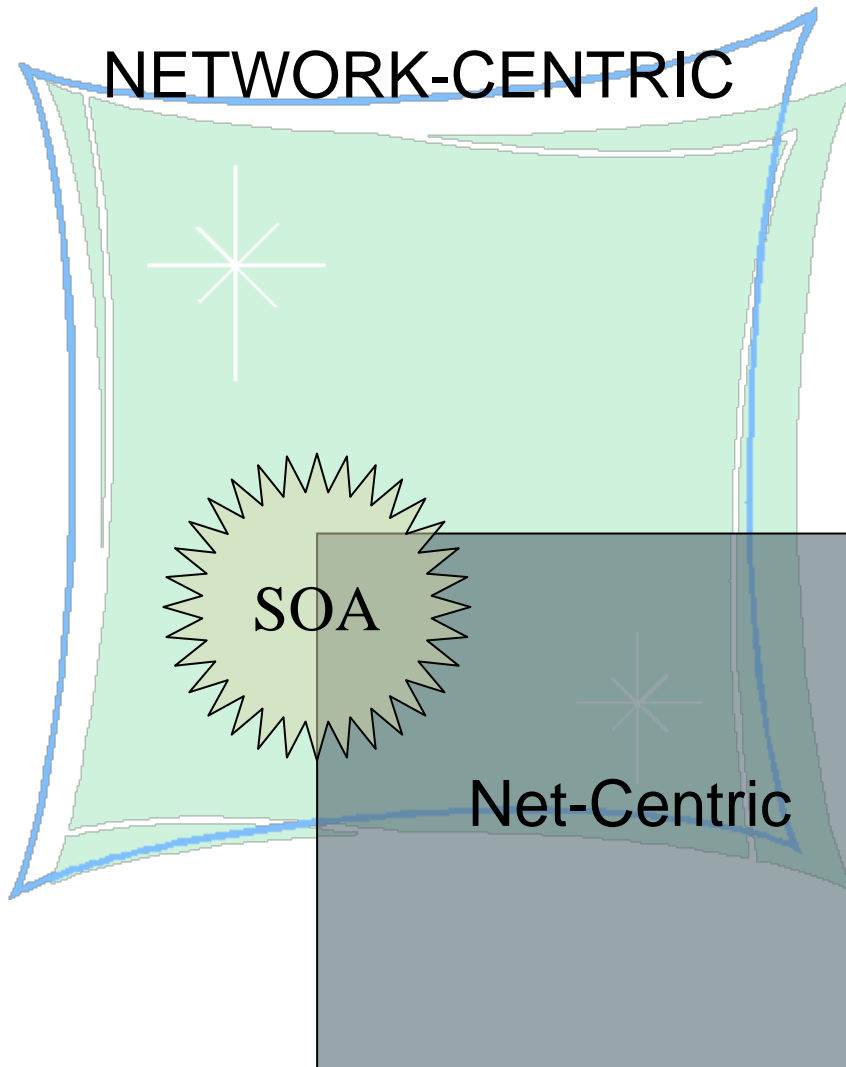
NCES CDD, adapted from CBDI Forum www.cbdiforum.com, also quoted by NCOW 1.1

- SOA shifts the focus further from large tightly-integrated (tightly-coupled/stovepiped) systems to policy- and standards-based services
- SOAs deliver capabilities for enterprise-wide solutions that are (or appear to have been) designed, developed, deployed as an integrated set of products that can be matured and maintained over longer periods

Services and Consumers Interoperate through Well-Defined Interfaces

Net-Centric/SOA Paradigm

VENN Relationship



- **Network-Centric** systems and **Net-Centric** systems exist on a spectrum and are not mutually exclusive
- **SOA** rides on and takes advantage of either context
- SOA focuses on the information system...the complex software-intensive system/ services/ capabilities
- ***This expanded information-sharing capacity has serious implications for the Real-Time Environment***

Real-Time Paradigm

The Real-Time Paradigm includes:

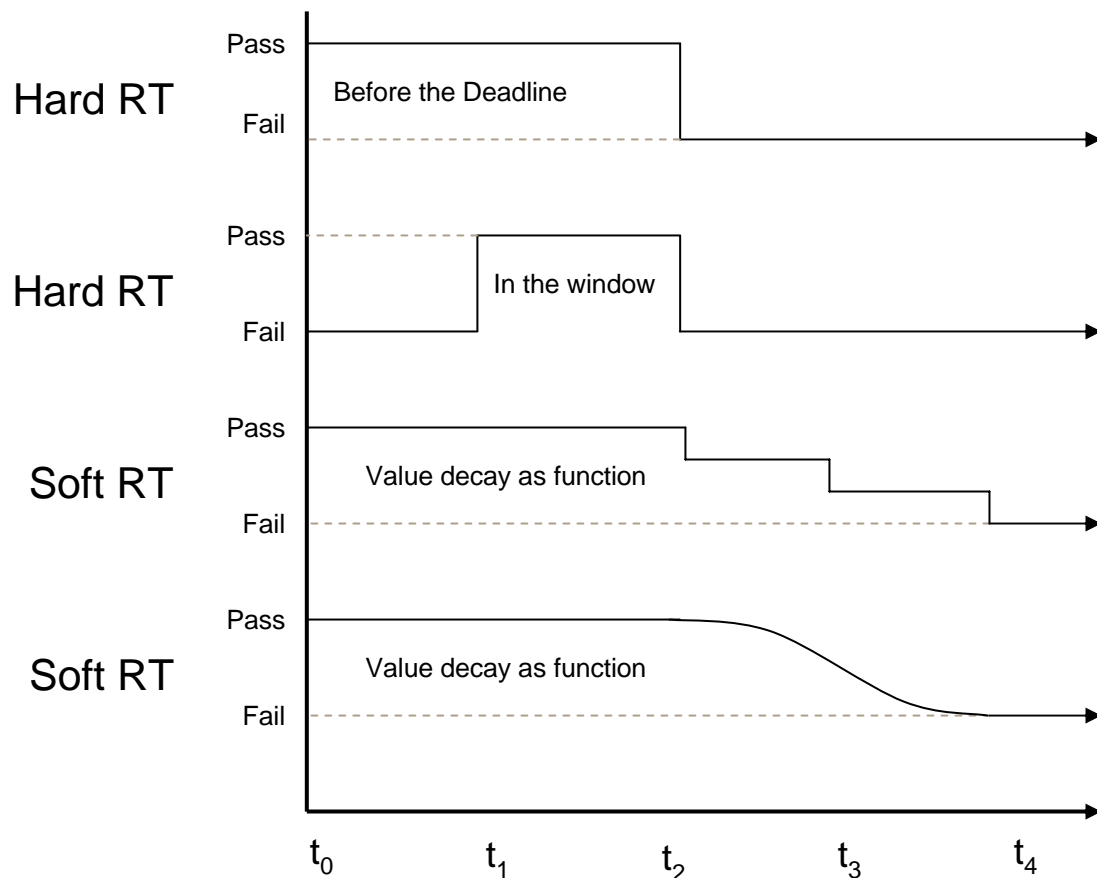
- **Validity of an Operation (Mission Success) predicated on:**
 - logical correctness -- the right data
 - delivery within defined timing constraints -- the right time
- **Timeline and time-scale constraints imposed by external conditions**
 - Dictated by one or more monitored or controlled physical processes or mission need-lines (threads)
 - Constraints satisfied for proper system behavior
 - Implementation deterministic... predictable... controllable

Implication – Significant Consequences for Breeches

Real-Time Paradigm

- **The Real-Time Paradigm includes vectors of speed, determinism, predictability, assurance, and reliability.**
 - **Hard real-time:**
 - Value/validity of results is nil if timeline is breached – *Late is Wrong*
 - Value curve looks like a step function
 - **Soft real-time:**
 - Value/validity of results diminishes over time or if timeline breached
 - Value/validity reaches nil at some finite time
 - **Near real-time:**
 - Used to indicate longer timelines or interrupted timelines
(man or IP-network in the loop)
 - **Non-real-time:**
 - No such thing (for non-trivial processing)

Value Pattern for Real-Time Paradigm



Time values for t_N can be μsec , msec , seconds , minutes , hours ...

Real-Time Is Right Time, Not Real Fast!

Real-Time Paradigm (cont'd)

Real-Time Continuum								
<i>Real-Time deadlines and timelines imposed by constraints outside control of the computer/ software</i>								
Near Real-Time (Some Latency Acceptable)			Soft Real-Time			Hard Real-Time		
<i>Involve longer timelines (or interrupted timelines) and often entail planning cycles</i>			<i>Deadlines are tight but not necessarily absolute; "value" of a computation diminishes after deadline expires</i>			<i>Deadlines must be satisfied for proper system behavior; processing timelines must be deterministic; "value" of computation is nil after deadline expires</i>		
Examples of Real-Time Patterns								
Logistics	Personnel	Fiscal	C&C	Intelligence	Medical	Time-Sensitive Targeting	Sensor/Machine Control	Effector/Flight control

Example 1- NASA Real-Time Scenario

Rocket Engine controller

- The space shuttle main engine controller needs to produce a set of commands for fuel flow valves every 5 ms. Miss one and the engine will burn through. Do them too fast and the control laws (being Z transforms) are incorrect.
- A discrete machine control loop, operating significantly close to the limits of digital control processor response times.
- The objective must be accomplished within a specified time window, or fail.
- Failure carries significant consequences.
- No Question of the Real-Time Pattern or status of this scenario...

Example 2 - Effects on Time on Target Real-Time Scenario

Effects on Time on Target in Theater

- A fighter-bomber mission to interdict traffic thread requires an effect on a specific target set within a specific window of time
- Must orchestrate command and control, sensors, and effectors of a system/enterprise to be sure the effect is correctly applied
- The specific success window, from a few seconds to a few minutes or even hours
- The window may be offset in time based on a decision cycle or set of trigger events or other guidance
- Success criteria are obviously set externally to the system
- The effect of arriving too soon or too late is a failure with dire consequences...
 - The effect might be applied to a wrong target, possible friendly, or applied to no useful target.

Analysis of the Real Time Scenarios

- In each case we have the same Real-Time Pattern defined by success/failure criteria that follows the second Hard Real-Time Pattern of a validity window.
- Each has significant consequences for failure.
- Each has a significantly different time scale... by orders of magnitude
 - Time scale drives trades, timing and sizing studies, significant design choices

Analysis of the Real Time Scenarios (cont'd)

Each Scenario presents unique issues for design and implementation but the Real-Time Pattern is inescapable

- **The first Scenario -- NASA machine control:**

- Is a pattern industry has considerable experience with

- **The second Scenario -- Effects on Target:**

- Involves a Net-Centric enterprise which presents a solution space with less experience to draw on, as an integrated Net-Centric solution

- Made more difficult as consists of services, infrastructures, heterogeneous computing

- Also includes integrations of products from multiple vendors and/or programs that will change asynchronously over time

When Paradigms Collide!

- Core of the “collision” -- as more services and operations move to a common network of networks -- potential risk that Real-Time Operations are likely to suffer due to increases in bandwidth constraints (GIG-BE notwithstanding) if networks are not judiciously-engineered and managed
- Concerns and resulting resistance to the Net-Centric/ SOA Transformation paradigm by the Real-time Community risks slowing the momentum of Transformation efforts -- albeit for very sound, observable reasons
- In progression from Network-Centric (“Swivel-Chair engineering”) to increasingly Net-Centric and SOA environments (real integration and interoperability), the Real-Time Community considers itself isolated

Recommendations Intro

- Since the Real-Time and Net-Centric/SOA communities must work together to provide critical Warfighter mission needs, the current gap in understanding and cooperation between these two vital communities must be bridged through:
 - Policy
 - Cultural Dynamics
 - Network Architectures
 - Technical Solutions

Recommendations: Policy

- DoD and the Services and the Real-Time Community need to create together agreed-upon typology, sets of standards, and architectural patterns for the Real-Time Community
- Policies that support the Net-Centric/SOA paradigm need to include the establishment of strategies and Advisory/Oversight bodies dedicated to support the Real-Time problem space
 - Network Strategies
 - Service Interface/Interoperability and Deployment Strategies
- The Real-Time Community and the Net-Centric/SOA Community are both integral to the DoD enterprise and need to collaborate actively in continued development of Net-Centric transformation policy through emerging CONOPS, Architectures, and Design Patterns

Define Needs - Set Goals - Drive Solutions

Recommendations: Cultural Dynamics

- Changes in policy, network architectures, and technical solutions are inter-related, interdependent, and dynamic:
 - Policy provides over-arching guidance
 - Architectures inform design and 1st step to technical solutions
 - Technical solutions embody design and implementation in alignment with architectures and specific mission needs
- The Real-Time and Net-Centric/SOA Communities need policy mechanisms to work together for best approaches
- As with many new directions in DoD -- in addition to changes in policy, architecture, and technical solutions -- culture change will be needed on the part of both the Real-Time Community and the Net-Centric/SOA Community

Recommendations: Network Architectures

- Real-Time Separation:
 - The use of dedicated network structures is one solution to insure QoS for Real-time users
- Real-Time Enclave through Segregation:
 - Segregation includes the set solutions that include VPN, “tunneling” and Encryption of “network routes” on existing networks.
 - This is a less robust solution but one that lends itself to more Net-Centric Architectures
- Analysis of Enterprise Networks to determine if current bounded areas are the result of Network Separation or Virtual Segregation -- critical because Virtually-Segregated networks can lose QoS due to others’ networks on the same backbone

Recommendations: Network Architectures (cont)

■ Policy & Network Architectures:

- Product development/design/deployment cycle support:
 - Policy that directs adherence to Network Architectures
 - Architectures that push run-time design choices as late as possible in the cycle facilitate service discovery, lead to less redesign and rework, and increase flexibility

■ Network Isolation or Separation must be:

- Pushed down to the lowest level of granularity, so that Real-Time needs don't Balkanize the emerging Net-centric DoD Enterprise
- Tempered by organizational needs and mission success goals
- Provide accessibility for mission critical information/command flow

Recommendations: Technical Solutions

- Infrastructure and Applications that prioritize Real-Time and Time-Sensitive packets over shared networks either using some kind of route management or on the fly compressions
 - Assumes viable, multi-phase network strategy that considers
 - Design Time
 - Integration Time
 - Pre-Deployment and Deployment Orchestration Time
 - Run-time Management

Summary

- To achieve success, Real-Time mission-critical operations must *engage* in dialogue and policy development with both:
 - Traditional Real-Time Communities
 - Net-Centric, SOA Communities
- Technical solutions must be developed to allow Real-time network management
- Network segregation or separation must be:
 - Pushed down to the lowest level of granularity to avoid “Balkanization” of Net-centric enterprise
 - Tempered by mission needs and success goals
 - Provide accessibility for mission critical information/command flow, and not for sake of organizational turf
- Real-Time components must live within the Enterprise need-space and interoperate as a service
- Net-Centric Enterprises need to move forward, respecting Real-Time component constraints

Questions and Discussion