

# RT-25 Overview: Requirements Management for Net-Centric Enterprises

By

**Doug Bodner, Nenad Medvidovic, Barry Boehm, Jo Ann Lane, Bill Rouse,  
George Edwards, Ivo Krka, Animesh Podar, and Daniel Popescu**

**Annual SERC Research Review  
October 5-6, 2011  
University of Maryland  
Marriott Inn and Convention Center  
Hyattsville, MD**

- Overview
  - Motivation and approach
  - Methodological framework
- Technical details
  - Capabilities to requirements
  - Requirements to architectures
  - Case study analysis/support
- Validation
- Future work

# Overview

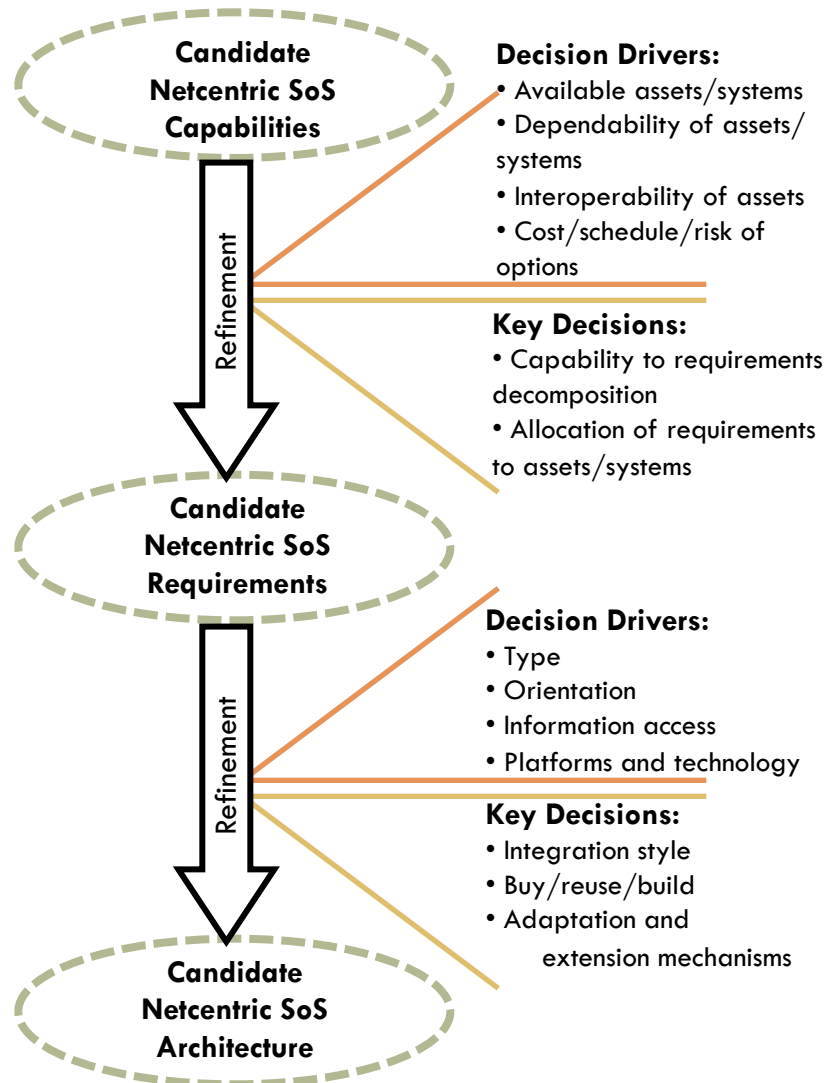
# Problem Statement



- Net-centric enterprises engage semi-autonomous business units, each with its own goals and methods for characterizing “requirements”
- These units often need to collaborate using common IT systems, involving integration or merging
- Missions and unit needs evolve over time
- Legacy systems exist and must be addressed
- How should capabilities and requirements be managed?

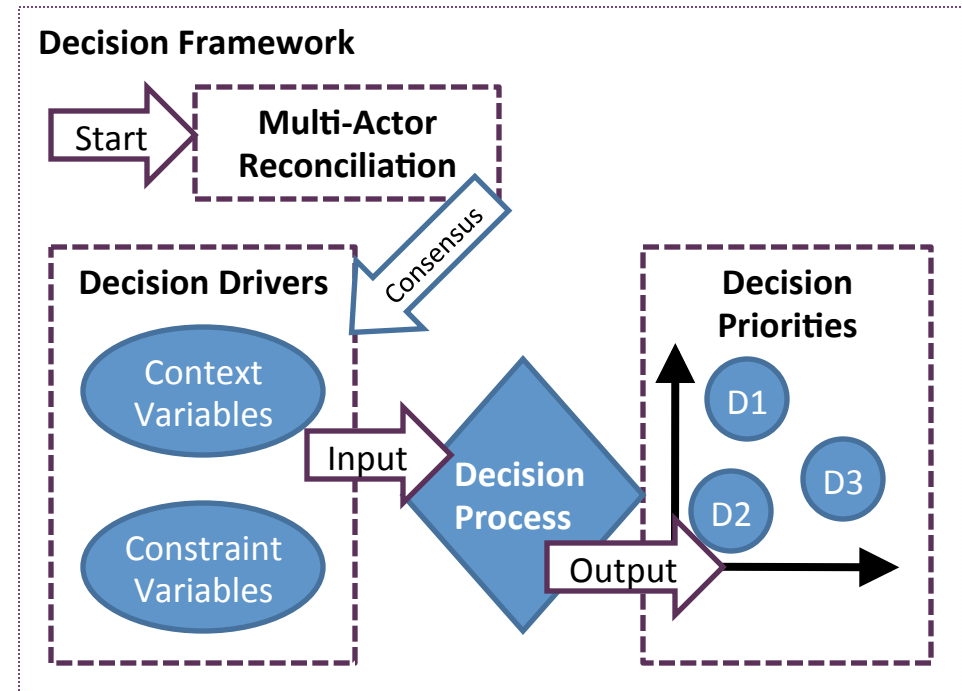
- Enable “requirements management” throughout integration lifecycle via a methodological framework and associated methods, processes and tools (MPTs)
  - Requirements definition and reconciliation
  - Traceability
  - Architecture specification
  - Balance between automation and decision support
- Address
  - Organizational differences
  - Selection-from-alternatives vs. design
  - Ambiguity and robustness
- Use concept of integration/mergers/connections as a framework
  - Integration case studies, including acknowledged SoS
- Validate with additional third-party IT integration experts

# Capabilities, Requirements & Architectures



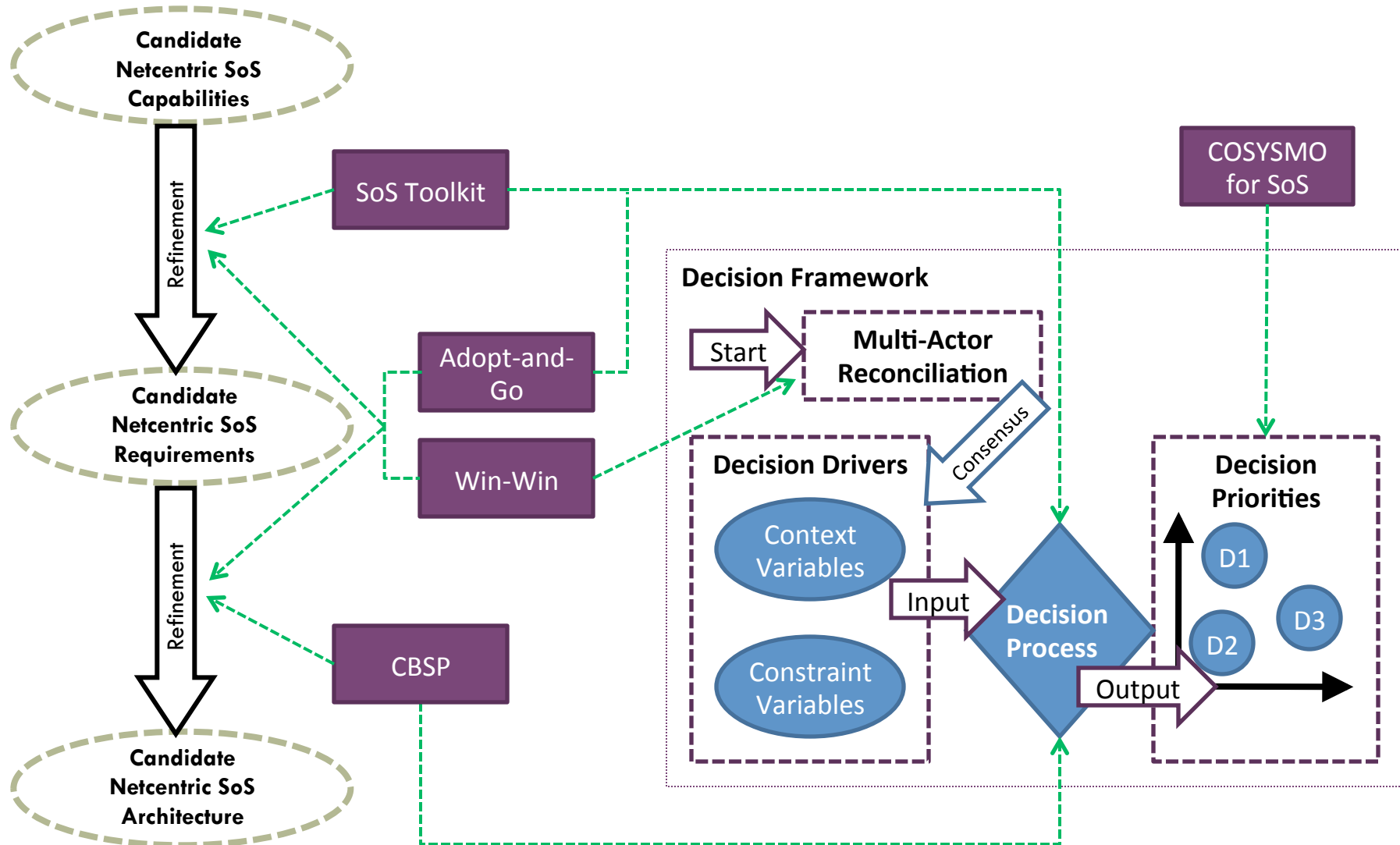
- Decompose high-level capabilities into software requirements
- Then into architectures
- Provide support for multiple stakeholders involved in net-centric integration
  - Conflicting needs
  - Compartmentalized information
- Provide support for traceability
- Use a spiral decision process to incrementally involve lower levels of detail and incorporate evolution of needs

- Facilitate reconciliation of conflicting capabilities and requirements among stakeholders
- Consider context of the system integration or merger (technical system characteristics, intended duration, etc.)
- Consider constraints that affect the integration or merger (technical constraints, cost/schedule, regulations)
- Evaluate priorities by considering value and risk
- Make decisions (select-from-existing vs. design-new, allocate capabilities to systems, etc.)



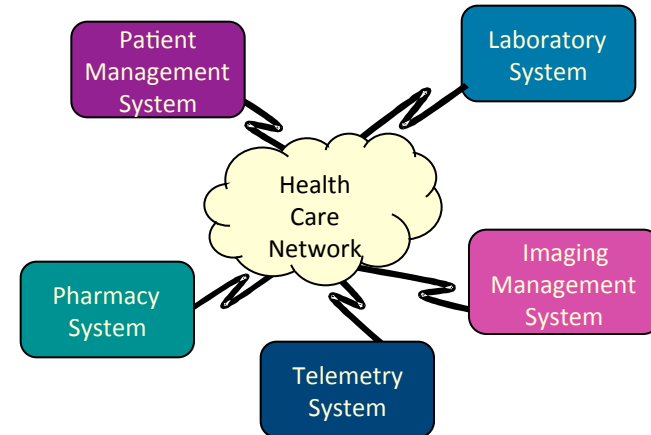
- Win-Win – method for negotiating and resolving multi-stakeholder conflicts regarding IT requirements
- System-of-systems toolkit – methods for going from capabilities to requirements
  - UML object models, reliability/dependability models, inter-operability/net-centricity matrices, use cases
- Adopt-and-Go – method for selecting one system from among multiple systems
- CBSP – method for deriving architecture design decisions from IT requirements
- COSYSMO for SoS – method for estimating cost of software-intensive system-of-systems given size factors and cost parameters
- These MPTs exist already, but need to be adapted for integration and net-centric enterprises





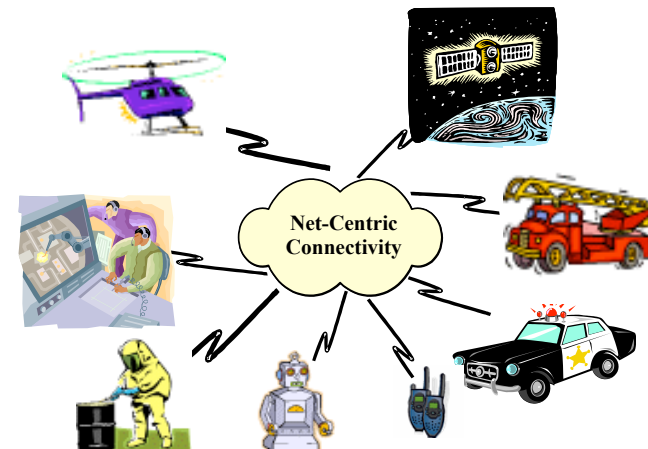
## • Goals

- Apply the methodology/MPTs:
  - Identify issues/challenges
  - Determine adaptations for MPTs
  - Evaluate methodology benefits/costs
- Expected outputs:
  - Manual/tutorial for methodology
  - Enumeration of remaining research problems
  - Evidence of value to the user

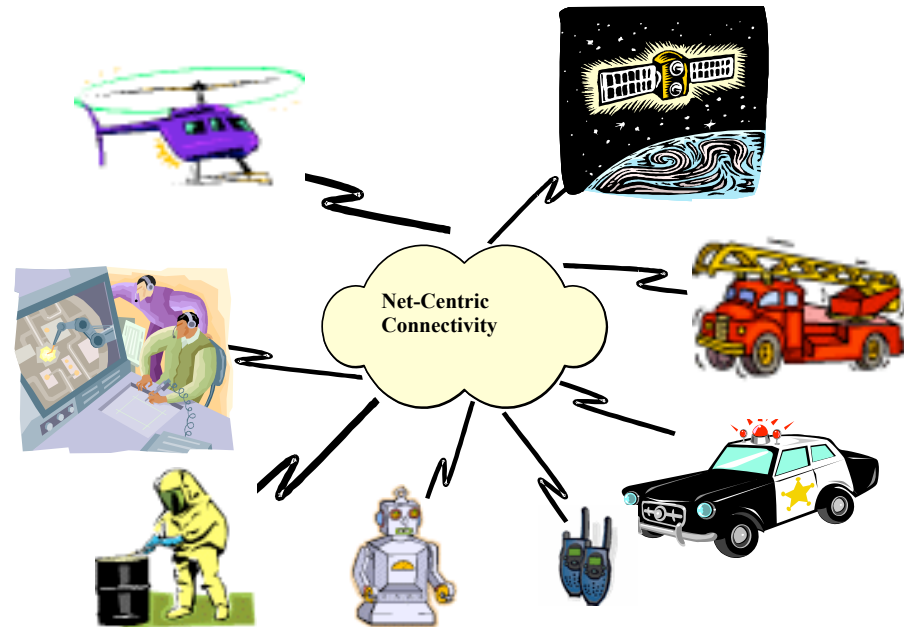
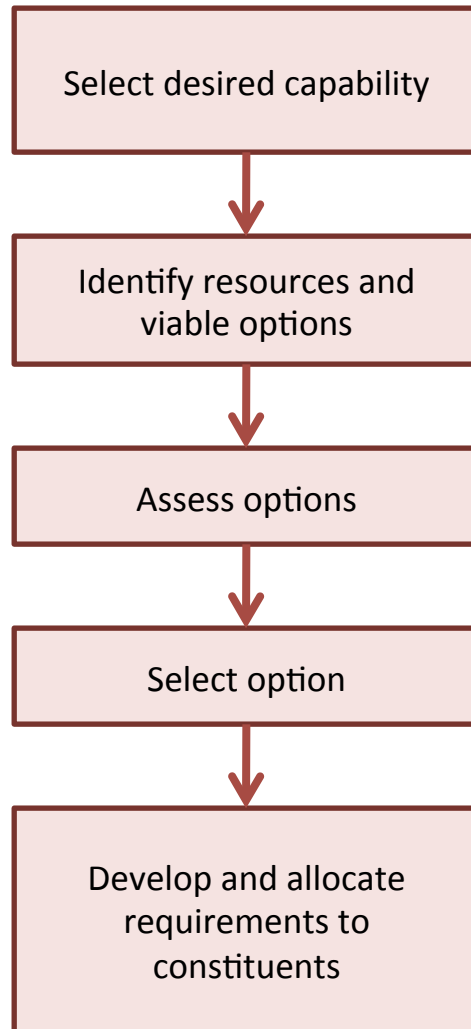


## • Case studies

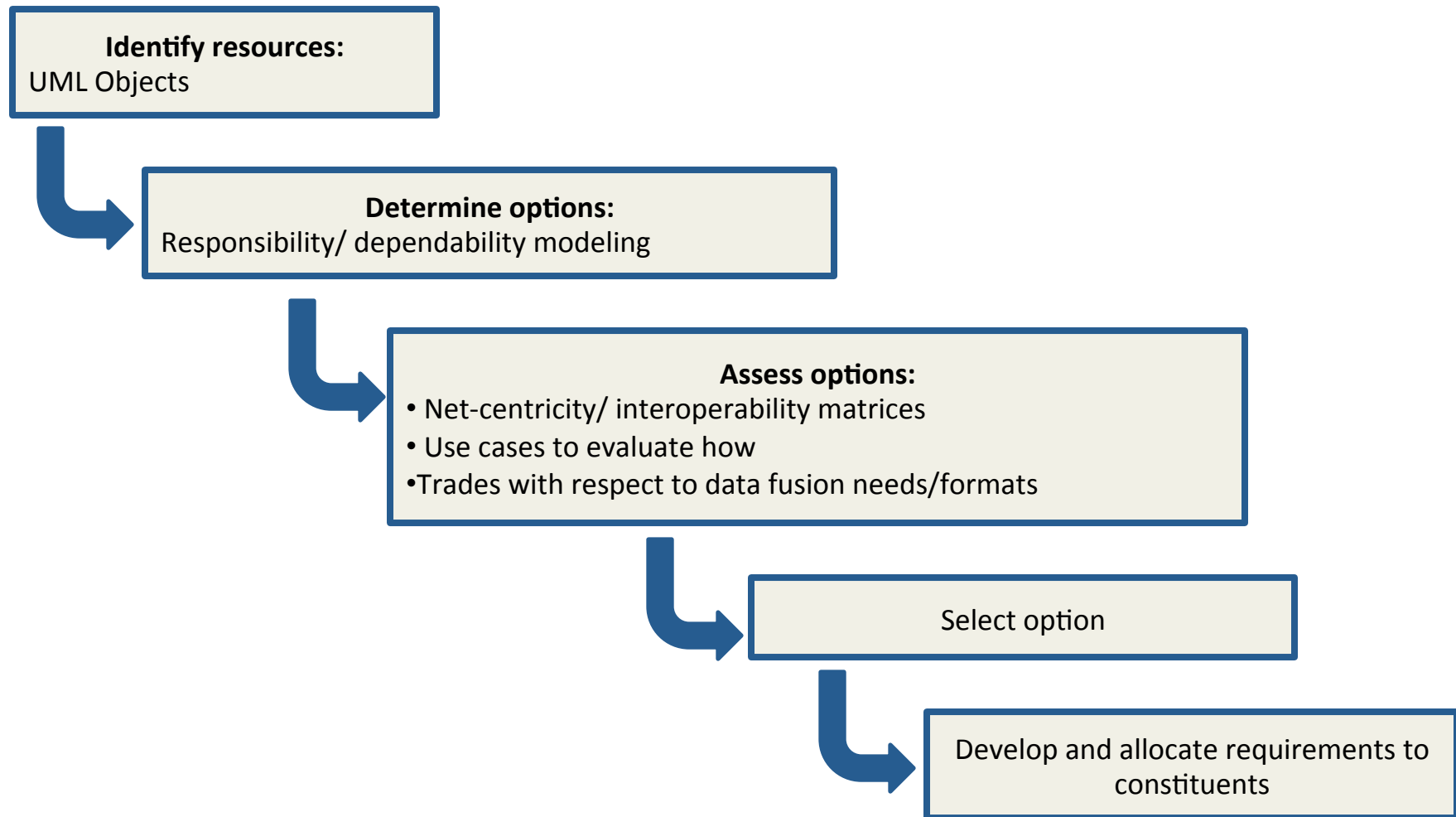
- Health IT
- Regional area crisis response system
- Corporate mergers (HP-Compaq)
- Back-office IT integration (ISP)



# Technical Details



**Illustrate using Regional Area Crisis Response SoS (RACRS)**



- Primary needs

- Improve number of fire-fighting resources available to fight major fires in the region
- Further reduce the time and number of official crisis management personnel resources required to evacuate a specified area
- Protect evacuated areas from looters

- Related goals

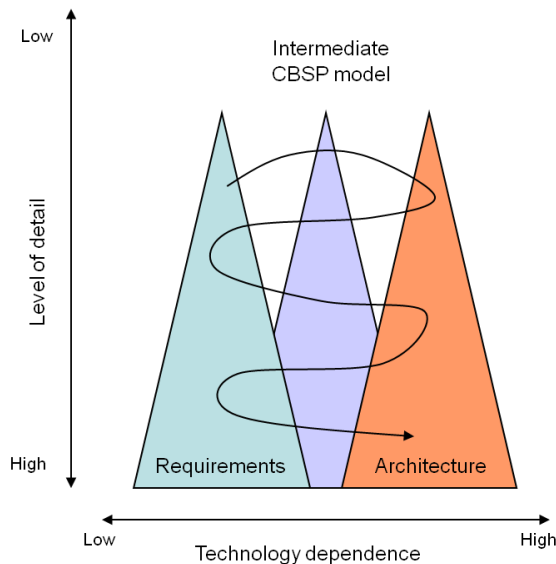
- Minimize local government expense (city, county)
- Minimize risk to human life (crisis responders and local population)
- Minimize workload on skilled personnel responsible for responding to crisis

- Potential Resources

- Local assets: professional responders and equipment, volunteers, low-risk inmates
- Military assets (personnel firefighting equip, UAVs, UGVs)
- TV/radio station announcers
- Satellite and local road camera images showing crisis area (e.g., fire) and traffic status
- Buses for transporting people
- New Reverse-911 system to support evacuation notifications.
- Homeowner alarm/security systems to support evacuation and protection

- iCBSP

- Proposed method for refining integration requirements into an integration architecture
- Augmented version of the CBSP method
- Retains strong traceability from architecture to requirements



- Process of use

- Pre-step: filter requirements for integration
- Step 1: stakeholders rate importance and feasibility
- Step 2: architects rate architectural relevance
- Step 3: architects negotiate and reconcile disagreements
- Step 4: requirements rephrased and traced to proto-architecture
- Step 5: integration architectural solution selected and applied to proto-architecture

- Style guides the composition of elements into an architecture
- Multiple styles may be used in a system or SoS
- Different styles result in different system qualities
- How styles are used in iCBSP:
  - Candidate styles are characterized according to advantages and disadvantages
  - Desired properties used to select appropriate style
- Integration Style =
  - Connector Roles + Topology + Linkage Mechanisms
  - Connectors
    - Adaptor, arbitrator, distributor, etc.
  - Topologies
    - Point-to-point, hub and spoke, shared bus, etc.
  - Linkage
    - Shared data, messaging, screen-scraping, etc.
  - Examples:
    - SOA = distributor, shared bus, messaging
    - Federated DB = arbitrator, hub and spoke, shared data



# Integration Matrix



Integration styles vs. Properties	Topology				Linkage				Connector			
	Point-to-Point	Hub and Spoke	Shared Bus	Peer-to-Peer	Shared Data	Messaging	Explicit invocation	Data Streaming	Adapter	Translator	Arbitrator	Distributor
Distributed	0	+	+	+	0	+	+	+	0	0	+	+
Local		-	+	-	+	0	+	+	0	0	0	-
Secure		-	0	+/-	-	0	0	0	0	0	+	-
Data intensive	+	-	-	+	+	-	0	+	0	-	+	+
Data formats incompatible	0	+	0	-	-	+	0	0	0	+	0	0
Data consistency	0	+	0	-	+	0	0	0	0	0	+	0
Interaction protocols incompatible	0	+	0	-	+	0	-	0	+	0	0	0
Reliable	+	-	+	+	-	+	+	0	0	0	+	0
Real time	+	-	+/-	-	+	-	+	+	+	+	+	0
One-to-many	-	+	+	+	+/-	+	+	+	+	+	+	+
Many-to-one	-	+	0	+/-	0	+	+	+	+	+	+	+
Always available	+	-	0	+	-	+	0	0	0	0	0	0
Periodically scheduled	+	0	0	-	0	0	0	0	0	0	0	0
Loose coupling	-	+	+	+/-	-	+	-	+	+	+	+	+
Robust	-	-	+	+	-	+	+/-	+	+	+	+	+
Dynamically reconfigurable	-	0	+	+	0	+	+	+	+	+	+	0
Scalable	-	-	+	+	-	+	0	0	0	0	+	+
Caching	-	+	+	0	+	0	-	-	0	-	+	+
Distributed transactions	-	+	+	+/-	+	+	+	0	0	0	+	+

P2P architectures effective at quickly disseminating data

- Provides data consistency and availability at seven San Diego County detention centers
- Interoperates with multiple external systems (Regional Area Crisis Response SoS)
- Security, privacy, performance, reliability & availability requirements
- Reasons for selection
  - Availability of requirements and data
  - Diverse issues and challenges (multiple systems, COTS, vertical and horizontal, transient and permanent integration, etc.)

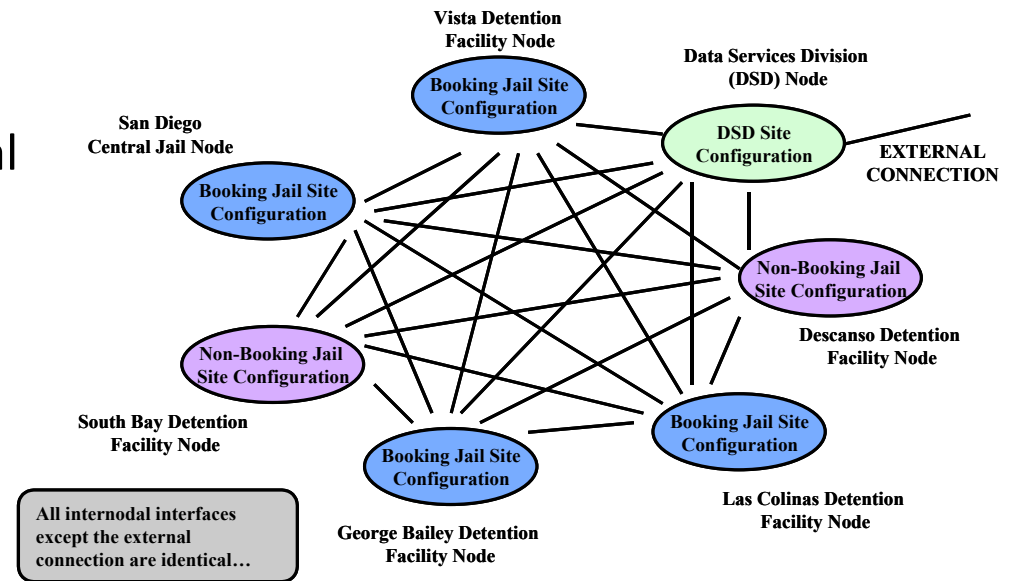


Figure 6. JIMS Top Level System View (SV-1) – Internodal Node Interfaces.

# Integration Matrix Application



Integration style vs. Properties	Topology				Linkage				Connector			
	Point-to-Point	Hub and Spoke	Shared Bus	Peer-to-Peer	Shared Data	Messaging	Explicit invocation	Data Streaming	Adapter	Translator	Arbitrator	Distributor
Distributed	0	+	+	+	0	+	+	+	0	0	+	+
Secure	+	-	0	+/-	-	0	0	0	0	0	+	-
Data intensive	+	-	-	+	-	-	0	+	0	-	+	+
Data consistency	0	+	0	-	+	0	0	-	0	0	+	0
Reliable	+	-	+	+	-	+	+	0	0	0	+	0
Real time	+	-	+/-	-	+	-	+	+	0	0	+	0
Robustness	-	-	+	+	-	+	+/-	-	0	0	+	+
Distributed transactions	-	+	+	+/-	+	+	+	0	0	0	+	+
Positive (+)	4	3	4	4	3	4	4	3	0	0	8	4
Neutral (o)	2	0	2	0	1	2	3	3	8	7	0	3
Negative (-)	2	5	1	2	4	2	0	2	0	1	0	1
Positive / Negative (+/-)	0	0	1	2	0	0	1	0				0

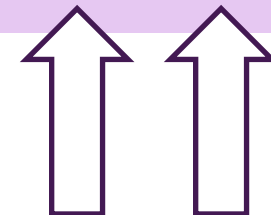
Summary of outcomes

# Decision Support Example



Integration styles vs. Properties	Topology				Linkage				Connector			
	Point-to-Point	Hub and Spoke	Shared Bus	Peer-to-Peer	Shared Data	Messaging	Explicit invocation	Data Streaming	Adapter	Translator	Arbitrator	Distributor
Distributed	0	+	+	+	0	+	+	+	0	0	+	+
Secure	+	-	0	+/-	-	0	0	0	0	0	+	-
Data intensive	+	-	-	+	-	-	0	+	0	-	+	+
Data consistency	0	+	0	-	+	0	0	-	0	0	+	0
Reliable	+	-	+	+	-	+	+	0	0	0	+	0
Real time	+	-	+/-	-	+	-	+	+	0	0	+	0
Robustness	-	-	+	+	-	+	+/-	-	0	0	+	+
Distributed transactions	-	+	+	+/-	+	+	+	0	0	0	+	+
Positive (+)	4	3	4	4	3	4	4	3	0	0	8	4
Neutral (0)	2	0	2	0	1	2	3	3	8	7	0	3
Negative (-)	2	5	1	2	4	2	0	2	0	1	0	1
Positive / Negative (+/-)	0	0					1	0	0	0	0	0

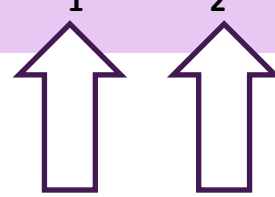
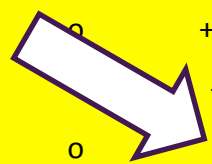
Adapters and translators are not needed as the interfaces are homogenous



# Decision Support Example



Integration styles vs. Properties	Topology				Linkage				Connector			
	Point-to-Point	Hub and Spoke	Shared Bus	Peer-to-Peer	Shared Data	Messaging	Explicit invocation	Data Streaming	Adapter	Translator	Arbitrator	Distributor
Distributed	0	+	+	+	0	+	+	+	0	0	+	+
Secure	+	-	0	+/-	-	0	0	0	0	0	+	-
Data intensive	+	-	0	+	-	-	0	+	0	-	+	+
Data consistency	0	+	0	-	+	0	0	-	0	0	+	0
Reliable	+	-	+	+	-	+	+	0	0	0	+	0
Real time	+	-	+/-	-	+	-	+	+	0	0	+	0
Robustness	-	-	+	+	-	+	+/-	-	0	0	+	+
Distributed transactions	-	+	+	+/-	+	+	+	0	0	0	+	+
Positive (+)	4	3	4	4	3	4	4	3	0	0	8	4
Neutral (o)	2	0	2	0	1	2	3	3	8	7	0	3
Negative (-)	2	5	1	2	4	2	0	2	0	1	0	1
Positive / Negative (+/-)	0	0	1	2	0	0						0



Combination of a peer-to-peer solution and a shared bus solution can circumvent such issues



The screenshot shows a web browser window displaying a wiki page. The page title is "Integration Styles vs. System Properties Table". The page content includes a table with columns for "Topology", "Linkage", and "Connector", and rows for various system properties like "Distributed", "Local", "Secure", etc. The table cells contain values such as 0, +, -, and +/-.

	Topology				Linkage				Connector			
	Point-to-Point	Hub and Spoke	Shared Bus	Peer-to-Peer	Shared Data	Messaging	Explicit Invocation	Data Streaming	Adapter	Translator	Arbitrator	Distributor
Distributed	0	+	+	+	0	+	+	+	+	0	+	+
Local	0	-	+	-	+	0	+	+	0	0	0	-
Secure	+	-	0	+/-	-	0	0	0	0	0	+	-
Data intensive	+	-	-	+	+	-	0	+	0	-	+	+
Data formats incompatible	0	+	0	-	-	+	0	-	0	+	0	0
Data consistency	0	+	0	-	+	0	0	0	0	0	+	0
Interaction protocols incompatible	0	+	0	-	+	0	-	0	+	0	0	0
Reliable	+	-	+	+	-	+	+	0	0	0	+	0
Real time	+	-	+/-	-	+	-	+	+	0	0	+	0
One-to-many	-	+	+	+	+/-	+	-	0	0	0	+	0
Many-to-one	-	+	0	+/-	0	+	-	0	0	0	+	+
Always available	+	-	0	+	-	+	0	0	0	0	+	0
Periodically scheduled	+	0	0	-	0	0	0	0	0	0	+	0

- Knowledge capture and management via wiki format (e.g., rationales)
- Platform for feedback on usefulness of proof-of-concept tool, plus relevance of row and column headings

# Benefits of Integration Matrix



- Capture and reuse knowledge
- Quickly “drill down” on a small set of potentially beneficial design options
- Identify potential issues early in the process
- Identify better alternative solutions

# Validation



- Determine capabilities and gaps with respect to managing requirements IT integration efforts in net-centric-like environments
- Determine extent to which our methods and tools address gaps
- Determine specific reactions
  - Enterprise systems integration
  - Health IT integration
- Surveys and interviews
  - Developed generic instrument
- Walk-throughs and usage

# Selected Findings



- Only sometimes is it the case that business capabilities are known beforehand and can be decomposed easily into IT requirements, as opposed to capabilities and requirements that must be elicited from the customer.
- Frequently, architectural conflicts between component systems are resolved by selection from existing alternatives rather than development of new software.
- IT integration projects very frequently involve data incompatibilities that must be discovered and resolved.
- Valuable knowledge for future projects is gained as legacy issues, data incompatibilities, and related concerns are addressed. Only sometimes is this knowledge captured and levered for future use.

- Project management
  - Capabilities driving IT requirements
  - Traceability of progress
- Data conversion
  - Data incompatibility between different systems is pervasive
  - Not only depends on the systems themselves, but also on how they are configured
- Knowledge management
  - How are solutions and experience captured for use in future, similar integration projects?
  - Ad-hoc methods used for the most part
  - Could accelerate productivity of new hires in gaining expertise

- MPT specification
  - Continue development of capabilities-to-requirements toolkit
  - Enhance iCBSP and integration matrix
- Methodology
  - Develop integration taxonomy and pointers to solutions
- Validation and usability
  - Incorporate additional validation via health IT partners
  - Elicit feedback on taxonomy and integration matrix
  - Devise configurability of integration matrix
  - Perform capability and gap analysis of commercial integration tools

# Questions?