Systems, Capabilities, Operations, Programs, and Enterprises (SCOPE) Model for Interoperability Assessment

Version 1.0

March 4, 2008

Authors

Richard Creps, Lockheed Martin

Hans Polzer, Lockheed Martin

John Yanosy, Rockwell Collins

Frank Sutton, Boeing

Allen Jones, Boeing

Reginald Ford, SRI

Amin Soleimani, Rockwell Collins

Todd Schneider, Raytheon

Developed on behalf of the
Network-Centric Operations Industry Consortium (NCOIC)

FINAL VERSION – CLEARED FOR RELEASE
NCOIC SCOPE Approved Version1 2008-03-04-V1.0

Table of Contents

1. INTRODUCTION 1

1.1 IDENTIFICATION AND PURPOSE 1
1.2 SCOPE AND APPLICABILITY 1
1.3 AUDIENCE 2
1.4 DOCUMENT OVERVIEW 2

2. BACKGROUND AND MOTIVATION 3

2.1 THE EMERGING NET-CENTRIC ENVIRONMENT 3
2.2 SHIFTING INTEROPERABILITY NEEDS AND CHALLENGES IN THE NET-CENTRIC ENVIRONMENT 3
2.2.1 What Is Interoperability 3
2.2.2 Key Interoperability Challenges 4
2.3 HISTORICAL APPROACHES TO INTEROPERABILITY 5
2.3.1 Recent Interoperability-Relevant Customer Initiatives 6
2.3.2 Levels of Information Systems Interoperability 7
2.3.3 NATO Industrial Advisory Group 8
2.4 THE NEED FOR AN ALTERNATIVE APPROACH 8

3. SYSTEMS, CAPABILITIES, OPERATIONS, PROGRAMS, AND ENTERPRISES (SCOPE) MODEL 11

3.1 NET-READINESS DIMENSIONS – 1.0 19
3.1.1 Discovery – 1.1 19
3.1.2 Autonomic Networking Dimension – 1.2 25
3.1.3 Information Assurance Capability – 1.3 37
3.1.4 Semantic Interoperability – 1.4 38
3.2 CAPABILITY/DOMAIN-INDEPENDENT SCOPE DIMENSIONS – 2.0 39
3.2.1 Enterprise Type – 2.1 39
3.2.2 Overall Scope – 2.2 40
3.2.3 Enterprise Breadth – 2.3 42
3.2.4 Enterprise Depth – 2.4 47
3.2.5 Semantic Interoperability – 2.5 49
3.2.6 Organizational Business Model and Culture – 2.6 64
3.2.7 Life Cycle Control – 2.7 65
3.3 CAPABILITY/DOMAIN-SPECIFIC SCOPE DIMENSIONS – 3.0 65
3.4 TECHNICAL AND ECONOMIC FEASIBILITY DIMENSIONS – 4.0 66
3.4.1 Inter-Element Time-Binding Sensitivity – 4.1 66
3.4.2 Transport Capacity Needed – 4.2 68
3.4.3 Run-Time Computing Resources Needed – 4.3 69
3.4.4 Enterprise Service Management Feasibility Dimension – 4.4 73
3.4.5 Interface Development Complexity Dimension – 4.5 76
3.4.6 Technology Readiness Level for System Connections Dimension – 4.6 77

4. EMERGING DIMENSIONS 80

4.1 EMERGING NET-READINESS DIMENSIONS 80
4.1.1 Information Assurance Capability Dimension - 1.3 80
4.1.2 Semantic Interoperability - 1.4 83

4.2 EMERGING CAPABILITY/DOMAIN-INDEPENDENT SCOPE DIMENSIONS 123
4.2.1 Organizational Business Model and Culture–(3.2.6) *123*
4.2.2 Life Cycle Control (3.2.7) *133*
4.2.3 Emerging Semantic Interoperability Dimensions *141*
4.3 EMERGING TECHNICAL AND ECONOMIC FEASIBILITY DIMENSIONS 145
4.3.1 Emerging Run-Time Computing Resources Needed Dimensions *145*

GLOSSARY 148

ACRONYMS 149

REFERENCES 152

APPENDIX A - FUTURE SCOPE DIMENSIONS 153

List of Figures

FIGURE 2-1 MAPPING OF INTEROPERABILITY APPROACHES 6
FIGURE 2-2 LISI DIMENSIONS AND LEVELS 7
FIGURE 2-3. MAPPING SYSTEMS, CAPABILITIES, OPERATIONS, PROGRAMS, AND ENTERPRISES 9
FIGURE 3-1 GRAPHICAL VIEW OF SCOPE DIMENSION HIERARCHY 12
FIGURE 3-2. DODAF VIEWS AND CAPABILITY ASSESSMENT DIMENSIONS/CRITERIA 16
FIGURE 3-3. DIMENSION DESCRIPTION STRUCTURE 17
FIGURE 3-4. AREAS OF CONCERN, USER INTEREST, AND RELEVANT SCOPE DIMENSIONS 18
FIGURE 3-5. COMPLEXITY BURDEN (HIGH AND LOW AUTONOMIC CAPABILITY) 25
FIGURE 3-6. NETWORK TENSION BETWEEN BENEFITS AND COSTS 27
FIGURE 3-7. REDUCING NEGATIVE COST EFFECTS THROUGH AUTONOMIC SOLUTIONS 28
FIGURE 3-8. AUTONOMIC DIMENSION CHARACTERIZATION MODEL 29
FIGURE 3-9. SEMANTIC INTEROPERABILITY DIMENSIONS 50
FIGURE 3-10. SEMANTIC INTEROPERABILITY CONCEPTUAL FRAMEWORK 54
FIGURE 3-11. EXAMPLE CAPABILITY/DOMAIN-SPECIFIC DIMENSIONS 66
FIGURE 3-12. NASA MAPPING OF TRLS INTO TECHNOLOGY LIFE-CYCLE PHASES 79
FIGURE 4-1. SEMANTIC INTERACTION ELEMENTS 84
FIGURE 4-2. SEMANTIC INTERACTION LAYERS IN A NETWORK AND COIS THAT USE AND DEFINE SEMANTICS 84
FIGURE 4-3. SEMANTIC INTEROPERABILITY DIMENSIONS (RED SHOWS NET-READY DIMENSIONS, BLUE SHOWS CAPABILITY DIMENSIONS) 85
FIGURE 4-4. HUMAN – TECHNOLOGY MEDIATED SEMANTIC INTERACTIONS 88
FIGURE 4-5. NETWORKED USERS TECHNOLOGY MEDIATED SEMANTIC INTERACTIONS 89
FIGURE 4-6. HIERARCHICAL SEMANTIC ABSTRACTION LAYERS FROM A NETWORK TECHNOLOGY PERSPECTIVE 90
FIGURE 4-7. EXPLICIT AND IMPLICIT SEMANTIC INTERACTIONS 92
FIGURE 4-8. EXPLICIT VERSUS IMPLICIT INTERACTION DEFINITIONS AND EFFECTS ON MUTUAL UNDERSTANDING AT HUMAN LEVEL AND COMPATIBILITY AT TECHNOLOGY LEVEL 93
FIGURE 4-9. SEMANTIC DEFINITIONAL AVAILABILITY DURING TECHNOLOGY LIFE CYCLE 96
FIGURE 4-10 SEMANTIC INTERACTION BETWEEN COGNITIVE AGENTS (CA-CA) 98
FIGURE 4-11. HYBRID SEMANTIC INTERACTIONS BETWEEN COGNITIVE AND REACTIVE AGENTS (CA-RA) 99
FIGURE 4-12. REACTIVE AGENT TO REACTIVE AGENT INTERACTIONS (RA-RA) 100
FIGURE 4-13. SEMANTIC INTERACTION BETWEEN NETWORKED ENTITIES 103
FIGURE 4-14. CONTEXT AND DOMAIN KNOWLEDGE MAPPINGS AND USE OF INTERACTION INTENT TYPES 106
FIGURE 4-15. DISJOINT SEMANTIC RELATION PATTERN 110
FIGURE4-16. OVERLAP SEMANTIC RELATION PATTERN 110
FIGURE 4-17. SUBSET SEMANTIC RELATION PATTERN 111
FIGURE 4-18. EQUIVALENT SEMANTIC RELATION PATTERN 111
FIGURE 4-19. SEMANTIC INTEROPERABILITY OF CONTEXT AND DOMAIN INTERACTIONS 112
FIGURE 4-20. SEMANTIC EXPRESSIVENESS ANALYSIS PROCESS 115
FIGURE 4-21. META ARCHITECTURE OF SEMANTIC ABSTRACTION LAYERS AND SEMANTIC EXPRESSIVENESS 117
FIGURE 4-22. CATEGORIES OF SEMANTIC EXPRESSIVENESS 119
FIGURE 4-23. STANDARDS FOR LEVELS OF SEMANTIC EXPRESSIVENESS – STANDARDS DIMENSION VALUES 122

FIGURE 4-24. SEMANTIC EXPRESSIVENESS COMPATIBILITY RULES 123
FIGURE 4-25. NETWORKING BUSINESS MODEL ELEMENTS, SUBORGANIZATIONS AND COMMUNITIES OF INTEREST (COIS) 127
FIGURE 4-26. GLOBALIZATION VERSUS SPECIALIZATION 142
FIGURE 4-27. GLOBAL VERSUS SPECIALIZED KNOWLEDGE CLASSIFICATIONS 145

List of Tables

TABLE 3-1. SCOPE DIMENSION HIERARCHY 13
TABLE 3-2. SERVICE DISCOVERY VALUE SET 20
TABLE 3-3. SERVICE DESCRIPTION RICHNESS VALUE SET 21
TABLE 3-4. SERVICE DESCRIPTION PUBLICATION/ACCESS MECHANISM VALUE SET 22
TABLE 3-5. SERVICE DISCOVERY TIME VALUE SET 23
TABLE 3-6. INFORMATION DISCOVERY DIMENSION VALUES 23
TABLE 3-7. AUTONOMIC HUMAN DEPENDENCY SUBDIMENSION VALUES 31
TABLE 3-8. GENERIC AUTONOMIC DEPENDENCY DIMENSION VALUES DEFINITIONS 34
TABLE 3-9. AUTONOMIC NETWORK OPERATIONS SUBDIMENSIONS AND VALUES 35
TABLE 3-10. ENTERPRISE TYPE VALUE SET 39
TABLE 3-11. ENTERPRISE SCALE VALUE SET 40
TABLE 3-12. ENTERPRISE HETEROGENEITY VALUE SET 41
TABLE 3-13. ENTERPRISE COHESIVENESS VALUE SET 42
TABLE 3-14. OPERATING CONCEPTS VALUE SET 43
TABLE 3-15. FUNCTIONAL CONCEPTS VALUE SET 43
TABLE 3-16. INTEGRATING CONCEPTS VALUE SET 44
TABLE 3-17. SWIM LANE VALUE SET 45
TABLE 3-18. DOTMLPF VALUE SET 46
TABLE 3-19. PMESII VALUE SET 47
TABLE 3-20. SICF DIMENSION—MODEL KNOWLEDGE ASSESSMENT VALUES 58
TABLE 3-21. SEMANTIC INTERACTION DIMENSION COMPATIBILITY VALUES 60
TABLE 3-22. HUMAN SEMANTIC DEPENDENCIES AND RELATIONSHIPS TO NETWORK PERSPECTIVE 61
TABLE 3-24. INTER-ELEMENT TIME-BINDING SENSITIVITY VALUE SET 67
TABLE 3-25. NETWORK LATENCY VALUE SET 69
TABLE 3-26. TECHNOLOGY READINESS LEVELS (NASA) 77
TABLE 4-1. SEMANTIC NETWORK LAYERS SUBDIMENSION VALUES 91
TABLE 4-2. TECHNOLOGY LIFE CYCLE SEMANTIC DEPENDENCY VALUES 96
TABLE 4-3. COGNITIVE SEMANTIC DIMENSION VALUES—COGNITIVE AND REACTIVE DESIGNS 101
TABLE 4-4. INTENTION SPEECH ACT CLASSIFICATIONS 105
TABLE 4-5. SEMANTIC INTEROPERABILITY PATTERNS FOR INTERACTING NETWORK ENTITIES ... 108
TABLE 4-6. SEMANTIC EXPRESSIVENESS SUBDIMENSIONS AND VALUES 116
.TABLE 4-7. DEFINITIONS OF LEVELS OF SEMANTIC EXPRESSIVENESS 119
TABLE 4-8 COMMUNITY OF INTEREST DEPENDENCY VALUE SET 129
TABLE 4-9. CULTURAL DEPENDENCY VALUE SET 130
TABLE 4-10. DYNAMIC STRUCTURAL DEPENDENCY VALUE SET 130
TABLE 4-11. ACCESSIBILITY VALUE SET 131
TABLE 4-12. STAKEHOLDER ALIGNMENT VALUE SET 135
TABLE 4-13. LIFE CYCLE TIMELINE CONGRUENCE VALUE SET 137
TABLE 4-14. COST OF TIMELINE INCONGRUENCE VALUE SET 139

1. Introduction

1.1 Identification and Purpose

This document describes the Systems, Capabilities, Operations, Programs, and Enterprises (SCOPE) interoperability assessment model developed by the Services and Information Interoperability Working Group (SII WG) of the Network-Centric Operations Industry Consortium™ (NCOIC). The SCOPE model is designed to characterize interoperability-relevant aspects or capabilities of a system or set of systems over a network in terms of a set of dimensions and values along those dimensions. The document is intended to articulate the motivation for developing the model and describe the dimensions of the model in detail. Although network-centric operations was originally defined in a military application context, it is also applicable to any operational domain that has the objective of achieving the benefits of collaborative human or machine networking through technology mediation. In these cases, the SCOPE model can be applied to characterize networking capabilities and its impact on human and machine networking. Examples of nonmilitary applications include the following:

- Emergency disaster response
- Supply chain logistics
- Inter-enterprise networks
- SOA networks

An earlier version of the document was developed before adoption by NCOIC, based on preexisting descriptive material. A general goal of Version 1.0 of the model is to evolve/extend the SCOPE dimension set and to be more specific and detailed in describing the dimensions.

1.2 Scope and Applicability

The purpose of this document is to define the SCOPE model and its dimensions. It is intended to be the authoritative source for detailed information about the SCOPE dimensions. This document is to be one of a set of SCOPE-related documents (or other published material, including Web-based Wikis) produced by NCOIC. These include the following:

- A SCOPE Practitioner's Guide that provides guidance to those who are interested in applying the SCOPE model and how best to use the model in performing SCOPE assessments.
- A SCOPE OWL ontology that can be used to create a knowledge base or SCOPE application tool to capture the results of SCOPE analyses.

Broadly speaking, the SCOPE model can be used as a descriptive or a prescriptive tool. That is, it can be used to assess the qualities of a set of capabilities as they currently exist (as-is), or to define the objectives for a set of capabilities to be built or evolved in the future (to-be). These applications of the model are complementary, in that doing both a descriptive assessment and a prescriptive goal-setting can help determine where the critical gaps and shortfalls exist in a capability evolution effort.

1.3 Audience

The intended audience of this document includes the acquisition community, governance organizations, standards organizations, managers of capability development efforts, system/capability architects and designers, testing and compliance organizations, and organizations involved in producing net-centric guidance and support products.

1.4 Document Overview

This document is organized into the following sections and supplementary material:

- *Section 1: Introduction*—An overview of the document (this section).
- *Section 2: Background and Motivation*—How the evolving networking environment, including DOD NCO, motivated development of the SCOPE model.
- *Section 3: Systems, Capabilities, Operations, Programs, and Enterprises (SCOPE) Model*—A description of the SCOPE model.
- *Section 4: Emerging Dimensions*—Dimensions addressing emerging areas, for which significant and useful, but not finalized, descriptions have been developed.
- *Glossary*—Definitions of key terms used in the document.
- *Acronyms*—A list of the acronyms and abbreviations referenced in the document.
- *References*—External works referenced in the document.
- *Appendix A: Future SCOPE Dimensions*—A list of dimensions that have been identified for future inclusion in the model, but have not yet been described.

2. Background and Motivation

2.1 The Emerging Net-Centric Environment

Allied nations in NATO and the U.S. DOD are in the process of evolving their traditional mode of combat (including system-to-system interaction) to a paradigm of network-centric operations (NCO). This concept is described in [1], [2], or the DOD's Net-Centric Operations and Warfare Reference Model (NCOW RM), as an example. Traditional approaches to managing the relatively predictable interaction patterns in the pre-net-centric world must evolve to address more dynamic and less predictable patterns in a net-centric environment.

2.2 Shifting Interoperability Needs and Challenges in the Net-Centric Environment

The growing adoption of NCO by commercial, government, and military organizations, enabled by an ever-expanding networking capability, increasingly exposes previously isolated systems for interaction and possible interoperability. The need for speed and operational effectiveness across organizations or force structures changes the tradeoffs between information protection and information sharing between systems. These factors are creating new perspectives on interoperability needs and challenges in the coming decades.

2.2.1 What Is Interoperability

There are numerous definitions of interoperability, which is why it is such a challenging topic to cover, and why our customers are struggling with it. Following are some relevant definitions:

> **(DOD/NATO)** *The ability of systems, units, or forces to provide services to, and accept services from other systems, units, or forces and to use the services so exchanged to enable them to operate effectively together. (Joint Pub 1-02)*
>
> **(DOD only)** *The condition achieved among communications-electronics systems or items of communications-electronics equipment when information or services can be exchanged directly and satisfactorily between them and/or their users. The degree of interoperability should be defined when referring to specific cases. (Joint Pub 1-02)*
>
> **(NATO)** *The ability to operate in synergy in the execution of assigned tasks. (AAP-6 [2005])*
>
> **(IEEE)** *the ability of two or more systems or components to exchange information and to use the information that has been exchanged.*
>
> **(Wikipedia[1])** *Interoperability is connecting people, data, and diverse systems. The term can be defined in a technical way or in a broad way, taking into account social, political, and organizational factors.*

[1] http://en.wikipedia.org/wiki/Interoperability#_note-0

Interoperability, as it applies to information systems, has meaning in the specific operational context of both the systems involved and their users/sponsors context. While there are aspects of interoperability that may be viewed as context-independent at the physical and network connection level, the exchange of information between systems typically has a purpose. As these definitions illustrate, that purpose is to provide some service to other systems and their users in order to achieve an operational task or objective. So one should always ask "for what purpose(s) must these systems interoperate?" These definitions imply a certain degree of collaborative interdependence between people, systems, units, forces, communities of interest (COIs), organizations, and any other elements comprising the network. Few are centrally designed, provisioned, and controlled. There is not a bright line between the concepts of interoperability and system integration. Rather, interoperability is the increasingly preferred term as the degree of collaborative interdependence of the systems involved increases. Being able to adequately characterize this degree of collaborative interdependence is important to understanding interoperability challenges and developing appropriate architecture and design strategies to address those challenges. This document does not address physical interoperability of military or commercial equipment or forces. It *does* address doctrinal and business process interoperability to the extent that such interoperability manifests itself in information systems. Increasingly, doctrine, rules of engagement, task representations, plans, collaborative command and control, and similar concepts are embodied in information systems along with more traditional representations of entities in the operational space. Systems that provide services dealing with these concepts require consistency of concept, and an analysis and mitigation plan for those conceptual/doctrinal conflicts that exist where human intervention is required.

2.2.2 Key Interoperability Challenges

Interoperability is especially challenging for individual programs due to limited program scope in a changing environment that has increasingly broad operational contexts. This issue is amplified in the net-centric environment, where a program may make its services available on the network without knowledge of who the clients for those services are.

Application context of a system is usually implicit in the nature and purpose of the system, and is not usually represented in the implementation of the system itself, although much can be gained by observing the interfaces of a system and any standards used there. There are usually some explicit context description in system requirements and specification documents, but little of this finds its way into a run-time representation in the system itself. Most systems cannot be interrogated to self-describe their capabilities or context of application, but this is changing with the advent of self-describing web services using Web Services Description Language (WSDL) as one example. One system attempting to interact with another system is expected to know what the context of a given system is by virtue of its identity and other "out of band" knowledge that the system developers and operations users might have about the system. Interoperability brittleness typically occurs when the application domain of a system is expanded beyond its original concept of application.

Systems may be viewed has having services or capabilities that are application domain specific and elements that are application domain independent. Those elements that are application domain specific are the brittle elements that break when application context changes. For example, when a COI-specific data model is supported by a system that is not appropriate for another application domain even though the services required appear to be equivalent in each domain, the processing of information unique to each domain provides for the break in interoperability. In addition, a given program or system is usually part of multiple capabilities, supporting multiple enterprises, if only indirectly. Even if all the systems in a system-of-systems are centrally sponsored, managed, and

integrated, it is rare that they are all acquired simultaneously, or through the same sources. They are seldom on the same acquisition and technology timeline. Likewise, this centrally managed and integrated capability is still faced with security considerations that create obstacles to information flow, and require balancing operational security risks with operational effectiveness, especially when crossing functional domains and coalition boundaries.

One of the most significant interoperability challenges is the fact that systems have explicit operational context and represent operational entities in ways that best support a unique operational context. When these systems interact with others, they expect to exchange information about domain objects as if operational context was shared across the systems involved. While it seems burdensome and may be unnecessary in some circumstances, systems need to be more explicit about representing their operational context in the interfaces and services they provide to other systems.

2.3 Historical Approaches to Interoperability

The information system interoperability challenge is not new, as evidenced by the many efforts and approaches that have been pursued by users, industry, research organizations, standards bodies, government organizations, and consortiums. One way to characterize these approaches is *top-down* versus *bottom-up*. Top-down approaches typically come at the problem from an enterprise architecture or broad scope perspective (usually directive/mandate oriented), and through different layers of abstractions using modeling tools, while bottom-up approaches seek to achieve interoperability by adoption of specific technologies or information representation standards. Another way of looking at top-down is having a specific organization's enterprise or mission objectives drive interoperability, while bottom-up generally is focused on technical approaches that are independent of the particular organization that might adopt them.

While contrasting top-down with bottom-up approaches is a useful distinction, *commonality-based* approaches focus on the execution environments of systems and try to achieve interoperability across a given enterprise scope by having every system within that scope adopt a particular set of standard elements for their execution environment. These approaches usually emphasize compliance, although some—such as Domain Name System (DNS) and Java—gain acceptance through technology adoption dynamics and market pressures as much as by mandate. The flip side of the commonality approach is the *system interaction-based* approach. This approach focuses on the space between systems rather than on what the execution environment might be inside a given system's boundary. Interaction-based approaches operate outside the bounds of execution environment as a "black box" approach to interoperability.

Figure 2-1 lays out these two approaches in a grid and makes a notional attempt to position specific concepts, initiatives, technologies, and products within that space.

Most programs will have elements of all of these approaches. In areas where the program sponsor exerts some control, commonality-based approaches can apply, while in areas where interacting systems are under heterogeneous control, interaction-based approaches will most likely be more appropriate and tractable.

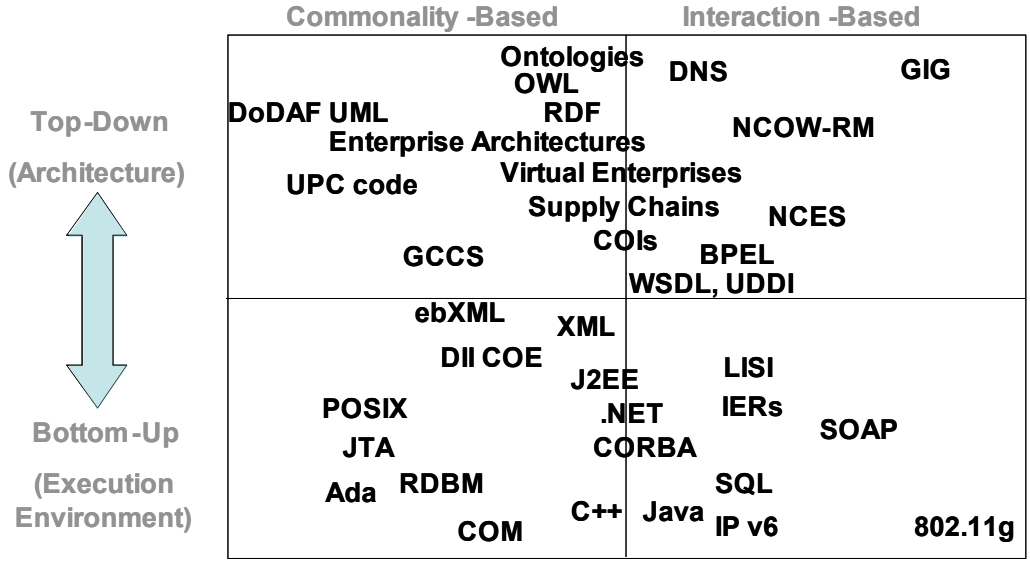


Figure 2-1 Mapping of Interoperability Approaches

2.3.1 Recent Interoperability-Relevant Customer Initiatives

The DOD Joint Capability Integration and Development System (JCIDS) [3] was presented in early 2003 to formalize a new process for developing requirements for operational capabilities to be implemented by one or more programs and systems. It is an architecture-driven process and includes a new set of deliverable documents. A subsequent DOD policy released in late 2003 has augmented the JCIDS process with the "Net Ready" Key Performance Parameter (NR KPP) concept, replacing the Information Exchange Requirements (IER)-based process that existed in earlier versions of the JCIDS document series. More recently, NATO has conducted a NATO Network Enabled Capability (NNEC) Feasibility Study [4], which explored similar issues and is considering adopting a NATO Maturity Level (NML) assessment that is similar to the NR KPP approach. While there is still work to be done to fully integrate the NR KPP concept with the acquisition process, the key concept it embodies is that systems should publish or expose a service on the network independent of which other systems might use it, rather than the point-to-point system interaction implicit in the IER approach to interoperability.

The DOD Architecture Framework (DODAF) [5] was updated to version 1.5 in late 2006 to better address net-centric concepts. The DODAF Technical Working Group has also resumed work on DoDAF version 2.0 to further develop its support for net-centric approaches to systems and to take advantage of work done in other, similar architecture frameworks such as the Ministry of Defense Architecture Framework (MODAF) [6] and The Open Group Architecture Framework (TOGAF) [7].

The U.S. DOD is establishing the Global Information Grid (GIG) [8] as a key enabler of network-centric systems of systems. Their Net-Centric Enterprise Services (NCES) [9] are being developed to provide infrastructure services that make the GIG more usable by constituent systems. NCES also makes it feasible for COIs to establish common services above the program/system level to support multiple systems and capabilities. NATO's NNEC initiative is contemplating a similar approach and service set, as are other countries such as Sweden and Australia.

To help assess interoperability needs and/or readiness and support decisions about the interoperability approaches to be taken in a given context, interoperability assessment frameworks and models have

been developed in recent years. Two of these, the Levels of Information Systems Interoperability (LISI) and the NATO Industrial Advisory Group (NIAG) models, are discussed in the following.

2.3.2 Levels of Information Systems Interoperability

The LISI [10] model and associated process were developed by MITRE in the late 1990s as a means of assessing the interoperability readiness of a system or set of capabilities. The LISI model is organized into four dimensions: procedures, applications, infrastructure, and data (PAID). The dimensions are assessed in terms of five hierarchical levels of interoperability readiness: isolated, connected, functional, domain, and enterprise. These dimensions and levels are shown in Figure 2-2.

The model implies that it is always good to target the highest level possible, in contrast to the SCOPE approach that enables finding the right level that best balances all factors in a given context. The LISI levels are an important characterization of the scope of networking capability in an NCO environment, since each hierarchical level determines the level of interoperability and access to wider and wider networked environments. The issue of the relative breadth of networking coverage as defined by the LISI levels is important and is encompassed within the SCOPE model.

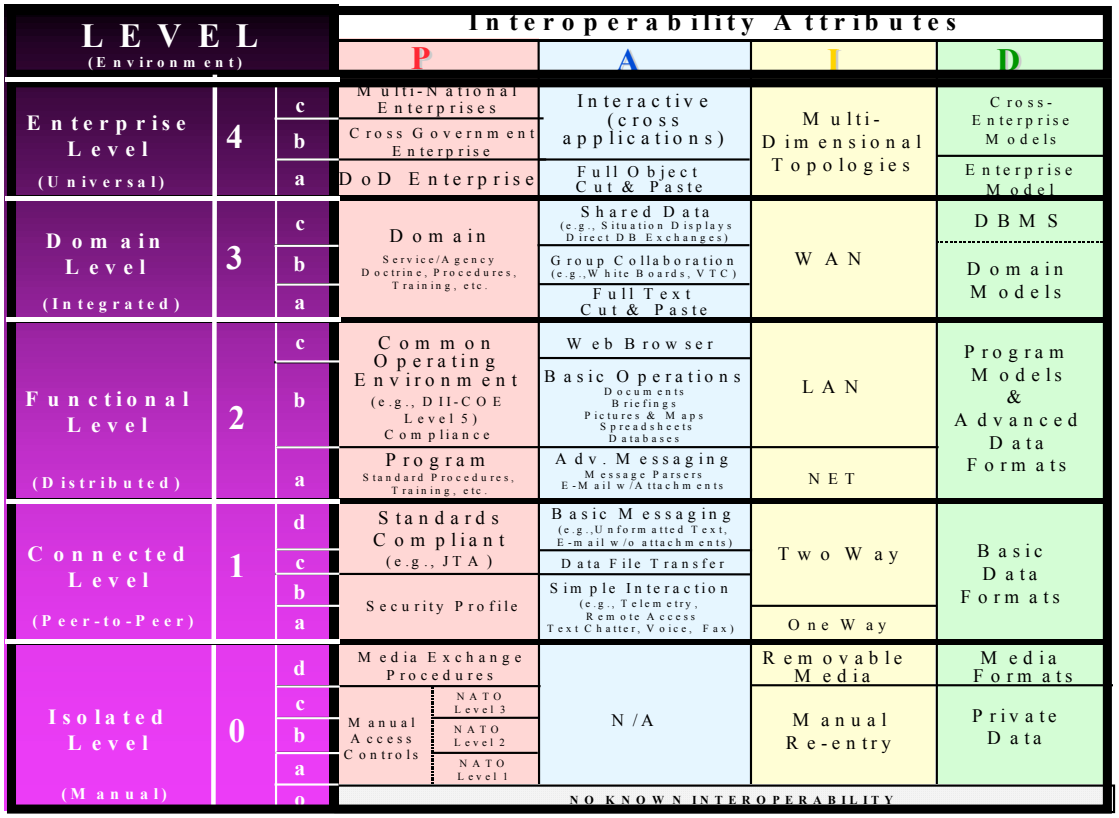


Figure 2-2 LISI Dimensions and Levels

The emerging NR KPP process until recently required that a LISI assessment be performed and a LISI profile be developed as part of the capability development process. This was problematic because LISI was developed before net-centricity became a common goal, and therefore, LISI does not incorporate an explicitly net-centric perspective. As a result, the LISI requirement has been removed from the NR KPP, and work has begun on a new interoperability assessment model to replace LISI. Perhaps this

new model would benefit from how the SCOPE model addresses the shortcomings of LISI from a net-centric perspective.

2.3.3 NATO Industrial Advisory Group

From late 2002 through 2003, NIAG conducted a study on naval C2 interoperability. This study group looked at an amphibious assault involving a joint, multinational task force, which included non-NATO force elements. This work confirmed that a scope boundary decision had significant downstream impact on the study because it made it clear that establishing a NATO standard (STANAG) would not solve the interoperability problem, leaving aside the problem that STANAGS were not consistently enforced and compliance was limited by individual NATO nations' budget priorities.

The study group looked at numerous NATO and U.S. architecture documents and standards, including various interoperability models such as LISI. There were significant shortcomings in all approaches. They did not fit well with naval C2 systems at the platform and individual weapons system level, nor did they provide much guidance on functional/domain/enterprise scope issues in a multinational and joint force structure.

The outcome of this study was a revised dimensional model that addressed many of the noted shortcomings. The model formed the core of what would become the SCOPE model described in this document.

2.4 The Need for an Alternative Approach

The motivation for developing the SCOPE model is that the original LISI model and DODAF views do not offer a means to characterize a capability scope in any definitive way. In addition, terms such as enterprise, capability, operations, system of systems, and programs are often used in various and overlapping ways. In the net-centric world, the boundaries between these concepts become less distinct and more situational. An operation such as Iraqi Freedom can be viewed as an enterprise or a system of systems from some perspectives. A particular program is certainly viewed as a (mini-)enterprise by its program manager, who may commission creation of information systems that support the program enterprise. The system may need to interact with systems at higher enterprise levels in the performing and customer organizations. Figure 2-3 [11] illustrates the complexity of mapping these boundary and scope concepts.

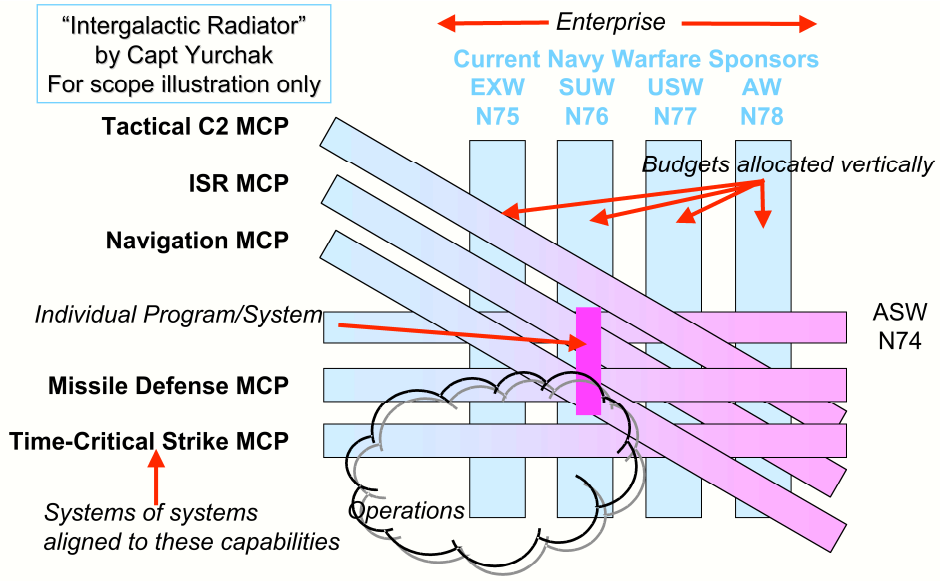


Figure 2-3. Mapping Systems, Capabilities, Operations, Programs, and Enterprises

It is expected that the capability analysis of the SCOPE model will help identify those constraining or missing issues that affect overall quality of interoperability capability dimensions. The model acknowledges that the issue of interoperability is not always reasonably characterized by a binary "true" or "false" state, but in many instances the interoperability answer will depend on the context and questioner. As with many types of knowledge, the context of the question may influence the answer, so that simple statements about net-centric tenets without the ability to set the scope of the question can result in misleading or erroneous capability characterizations. The SCOPE model attempts to clarify this capability characterization.

Another factor is that enterprises can intersect as well as interact with each other. This may be more evident in large-scale efforts, such as multinational coalitions, but is a factor even in a specific program. Such a program may be responsible for implementing a system of systems within its scope boundary, but it may also be responsible for building a component system of some larger system of systems outside its scope boundary. In both cases, it may be developing a complete capability or contributing to some larger operational capability. Multiple capabilities may be used in a given operation to varying degrees, depending on the nature of the operation.

An additional issue is the dependency between systems, services, and operational capabilities. Not all of the dependencies on the capabilities provided by a particular deployed system are obvious. Explicit representation and discovery of dependencies between systems, services, and data are required to enable networks to be maintained in a viable state for all users, or at least to be notified when a system, service, or data may not be available.

Ultimately, measuring the degree and scope of interoperability between systems and services on the network is more a question of relative scope scale and operational appropriateness than it is a matter of achieving some arbitrary level of interoperability "maturity." The employment of value-neutral terms such as dimension, scale, value, and measure or metrics better conveys this perspective. Similarly, the LISI PAID dimensions have an information technology bias and do not offer a way to assess how big or diverse an enterprise or capability a specific collection of systems is intended to support. Instead, it focuses on specific information exchanges with other systems that are assumed as "givens," because

they were the only elements in place that contributed to the proposed operational capability and architecture.

3. Systems, Capabilities, Operations, Programs, and Enterprises (SCOPE) Model

The SCOPE model is designed to characterize interoperability-relevant aspects of a system or capability in terms of a set of dimensions. The dimensions are organized hierarchically so that each dimension may contain subdimensions and these, in turn, may contain their own subdimensions, etc.. At the lowest level of the hierarchy, each dimension represents a specific aspect of a system/capability or its surrounding environment/context, and enables assessment of that aspect within a range of possible values (discrete or continuous) that is unique to that dimension. The SCOPE dimensions are not arrayed into a uniform set of "maturity" levels. Each dimension is characterized independently and in general without *a priori* assumptions about which values within the dimension are more or less desirable—such judgments depend on context.

The SCOPE dimensions described herein are not presumed to be complete. Ongoing evolution to add, refine, and organize the dimensions is expected. Furthermore, no strong claim is made here that the SCOPE dimensions are fully independent, as examination reveals that there are some interdependencies and cross-dimensional effects between the dimensions. An effort has been made to make the dimensions as independent as possible, while retaining the expressiveness needed to capture information in terms with which system/capability planners and designers are accustomed to working.

Dimension Structure Overview

The SCOPE model includes four top-level categories of dimensions:

- **Net-Readiness:** Ability to deliver capability in a network context.
- **Capability/Domain-Independent Scope:** The range of scope or context supported.
- **Capability/Domain-Dependent Scope:** The nature, quantity, quality, speed, etc., of capability provided to meet operational needs.
- **Technical/Economic Feasibility:** The feasibility or risk associated with providing capability.

An effective way of viewing these dimensions is as follows:

- The Capability/Domain-Independent and Capability/Domain-Dependent Scope dimensions define the **what**, **why, and where** of a capability (i.e., what specific capability is needed, in what context, and for what purpose?).
- The Net-Readiness dimensions define the **how** of a capability (i.e., how can that capability be implemented technically and delivered over a network?).
- The Technical/Economic Feasibility dimensions define the **how much** of a capability (i.e., given technical and economic tradeoffs, how much of the ideal technical solution is feasible and affordable?).

Within each of these top-level dimension categories, the SCOPE development team endeavored to make the initial set of SCOPE dimensions at the next level in the hierarchy as comprehensive as possible, while recognizing that more dimensions likely will be identified in the future. Some of the dimensions at this level are classified as *Emerging Dimensions*. These dimensions address important aspects of net-centric interoperability, but still need further development. They focus on technical areas

that the engineering community is becoming aware of and for which standards are emerging, yet for which there may be alternative ways of viewing and characterizing the problems encompassed by the dimension concepts. As a result, the underlying dimension structure, value sets, and technical narratives in these areas are "works in progress."

The Emerging Dimensions are included within the dimension hierarchy described in Section 3, but only at their top level, to indicate where in the model these dimensions will be positioned in their completed form in the future. Substantial material has been developed to characterize the Emerging Dimensions, and this information is included in Section 4, both because it conveys valuable information to the reader and because it may serve to attract the attention of those who would like to contribute further in these areas.

Figure 3-1 provides a graphical overview of the top three levels of the dimension hierarchy.

Table 3-1 describes the entire SCOPE dimension hierarchy. The column on the left identifies the section of the document in which each dimension is described.

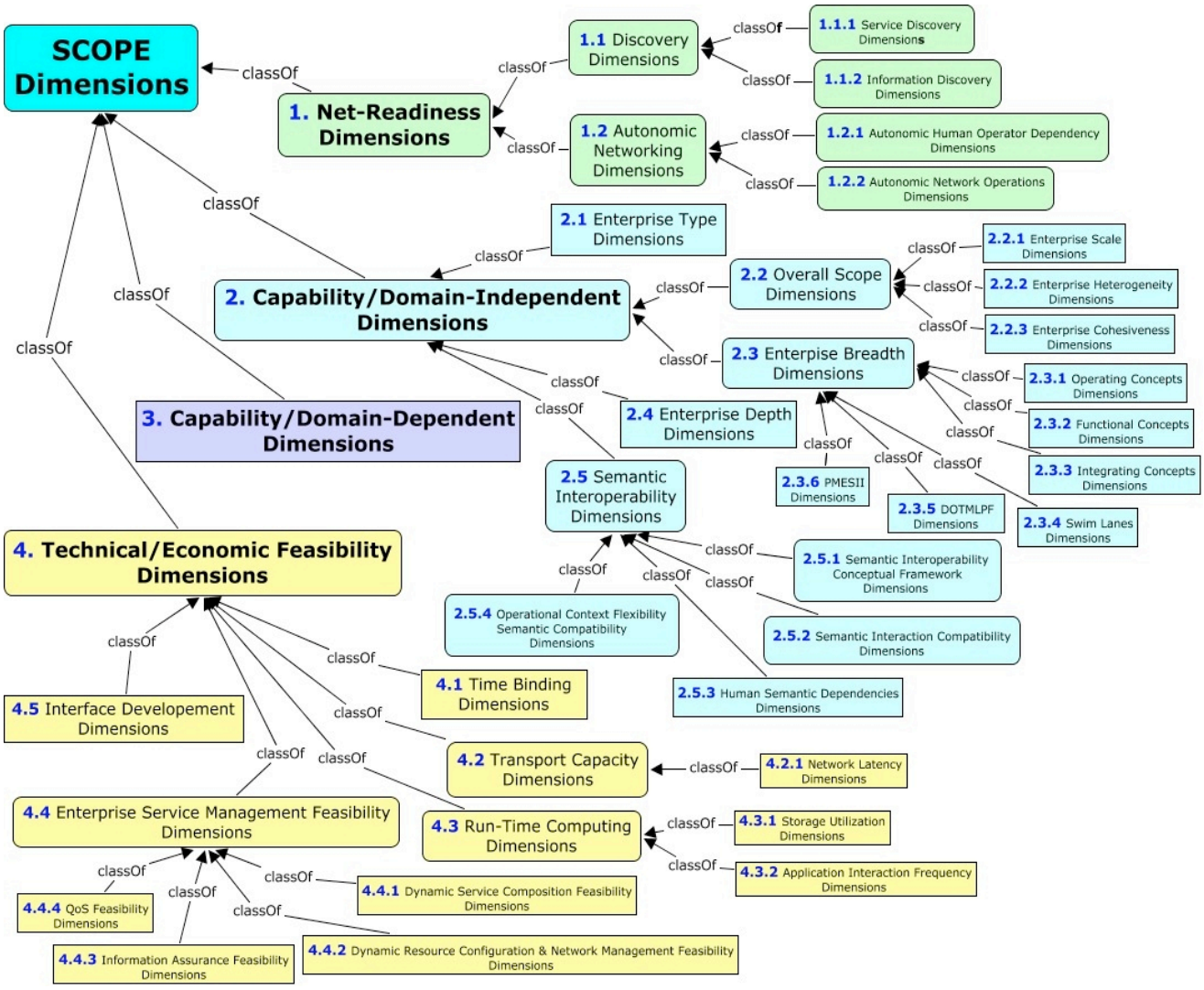


Figure 3-1 Graphical View of SCOPE Dimension Hierarchy

Table 3-1. SCOPE Dimension Hierarchy

Section 3	Subsection 1.0 Net-Readiness Dimensions
	1.1 Discovery 1.1.1 Service Discovery 1.1.1.1 Service Description Richness 1.1.1.2 Service Description Publication/Access Mechanism 1.1.1.3 Service Discovery Time 1.1.2 Information Discovery 1.1.2.1 Metadata Availability and Accessibility 1.1.2.2 Standard Metadata Language 1.1.2.3 COI Metadata Relevance 1.1.2.4 Query Metadata Matching Capability 1.1.2.5 Multiple Information Sources Relevance Metrics 1.1.2.6 Context Informed Queries 1.1.2.7 Information Model Pre-Agreement *1.1.2.8 Level of Semantic Expressiveness* 1.2 Autonomic Networking 1.2.1 Autonomic Human Operator Dependency 1.2.1.1 Autonomic Behavior 1.2.1.2 Autonomic Level of Interaction 1.2.1.3 Autonomic Validity Checking 1.2.1.4 Autonomic Guidance 1.2.2 Autonomic Network Operations 1.2.1.1 Autonomic Configuration 1.2.1.2 Autonomic Healing 1.2.1.3 Autonomic Optimization/Performance Management 1.2.1.4 Autonomic Protection 1.2.1.5 Autonomic Composeability 1.2.1.6 Autonomic Asset Management 1.2.1.7 Autonomic QoS *1.3 Information Assurance* *1.4 Semantic Interoperability*
Section 3	**Subsection 2.0 Capability/Domain-Independent Scope Dimensions**
	2.1 Enterprise Type 2.2 Overall Scope 2.2.1 Enterprise Scale 2.2.2 Enterprise Heterogeneity 2.2.3 Enterprise Cohesiveness 2.3 Enterprise Breadth 2.3.1 Operating Concepts 2.3.2 Functional Concepts 2.3.3 Integrating Concepts 2.3.4 Swim Lanes 2.3.5 DOTMLPF 2.3.6 PMESII 2.4 Enterprise Depth 2.5 Semantic Interoperability 2.5.1 Semantic Interoperability Conceptual Framework (SICF) 2.5.1.1 Common Environment Knowledge – Physical/ 2.5.1.2 Common Environment Knowledge – Social 2.5.1.3 Agent Classification 2.5.1.4 Agent Context 2.5.1.5 Agent Domain Knowledge 2.5.1.6 Agent Intentions – Speech Acts 2.5.2 Semantic Interaction Compatibility 2.5.2.1 Semantic Interaction Model Compatibility

	2.5.2.2 Language Compatibility 2.5.2.3 Intention Compatibility 2.5.2.4 Domain Knowledge Compatibility 2.5.2.5 Context Compatibility 2.5.2.6 Collaboration Compatibility 2.5.3 Human Semantic Dependencies 2.5.3.1 Individual Role Dependency 2.5.3.2 Social/Cultural Background Knowledge Dependency 2.5.3.3 Organizational Mission/Focus Dependency 2.5.3.4 Language Dependency 2.5.3.5 Domain Knowledge Dependency 2.5.3.6 Situational Context Dependency 2.5.4 Operational Context Flexibility 2.5.4.1 Time Context 2.5.4.2 Object Situational Context 2.5.4 2.1 Geospatial Location Context 2.5.4 2.1 Domain Classification Context 2.5.4.3 Multi-Agent Context (Shared Situational Context Knowledge) 2.5.4 3.1 Collaborative Relationship 2.5.4 3.2 Complementary Roles 2.5.4 3.3 Agent Capability Representation 2.5.4 3.4 Agreement/Commitment Protocol 2.5.4 3.5 Environment Domain Knowledge *2.5.5 Globalization versus Specialization of Domain Knowledge for Communities of Interest* *2.6 Organizational Business Model and Culture* *2.7 Life Cycle Control*
Section 3	**Subsection 3.0 Capability/Domain-Specific Scope Dimensions**
	<Varies per domain>
Section 3	**Subsection 4.0 Technical/Economic Feasibility Dimensions**
	4.1 Inter-Element Time Binding Sensitivity 4.2 Transport Capacity Needed 4.2.1 Network Latency 4.3 Run-Time Computing Resources Needed 4.3.1 Storage Utilization 4.3.2 Application Interaction Frequency/Pattern *4.3.3 Processor Utilization* *4.3.4 Nodal Quality of Service* 4.4 Enterprise Service Management Feasibility 4.4.1 Dynamic Service Composition Feasibility 4.4.2 Dynamic Resource Configuration and Network Management Feasibility 4.4.3 Information Assurance Feasibility 4.4.4 Quality-of-Service Feasibility 4.5 Interface Development Complexity 4.6 Technology Readiness Level for System Connections
Note: Dimensions in *orange italics* are Emerging Dimensions and are described more fully in Section 4.0, Emerging Dimensions. The structure of Section 4.0 mirrors this dimension structure for the Emerging Dimensions.	

Interpretation of the Model in DODAF Terms

The SCOPE model can be correlated with a number of other interoperability and architecture models. One illustrative way to interpret the SCOPE model is in terms of the DODAF operational, system, and technical architecture views. This offers a useful perspective on the major SCOPE dimension categories and how they interrelate. However, this does not imply that the SCOPE model is tied to or derived from DODAF. Mapping to other models is possible and encouraged, and the discussion of DODAF here is illustrative, but not essential.

The major SCOPE dimension categories identified previously can be viewed as representing mappings between each of the three DODAF architecture views to each other. They also reflect the fact that systems are often part of multiple operational capabilities.

The Net-Readiness dimensions help assess the degree to which the system architecture and associated views for the constituent systems of a system of systems map to the technical architecture views and standards. In some sense, this mapping is independent of the specific operational architecture that the constituent systems implement.

The Capability/Domain Scope dimensions help assess how well the information that flows among constituent systems satisfies the operational architecture capabilities, independent of the specific technical architecture elements employed by the systems to implement the operational capabilities.

The Technical/Economic Feasibility dimensions help assess the degree to which an operational capability is achievable, given the technical architecture standards and constraints. Alternatively, these dimensions characterize what the necessary technical architecture elements need to be to implement a desired level of operational capability, and how feasible or mature such technical elements might be. These dimensions are independent of the particular system architectures that might be used to implement the operational capability.

Figure 3-2 illustrates how the SCOPE dimensions relate to the DODAF architecture views. It also shows the kind of information that is contained or assessed along each interoperability dimension. The kind of information involved for each of the three major dimension sets typically is driven by different types of stakeholders. The capability scope dimensions are driven primarily by the institutional sponsors of a given enterprise, net-centric capability, or system. Their primary motivator is increased potential for greater/broader operational effectiveness. The net-readiness dimensions are driven primarily by program managers and capability/system implementers. Their primary motivators are to minimize the risks of system interdependence, reduce system development costs, and make their systems as adaptable as possible (within cost constraints) to a wide range of potential customer needs. The technical feasibility dimensions are driven primarily by the concerns of infrastructure providers and CIOs of organizations that are striving to deploy net-centric capabilities. Their motivators include use of open standards and keeping down the costs of common network transport and hosting services by reducing the variability and common resource costs associated with deploying network-centric capabilities. In the past, this was a major concern for program managers, but as capabilities become more network centric, such infrastructure concerns will increasingly be addressed at the “enterprise” level, rather than at the program level, thus promoting greater commonality and more effective resource utilization from an enterprise perspective.

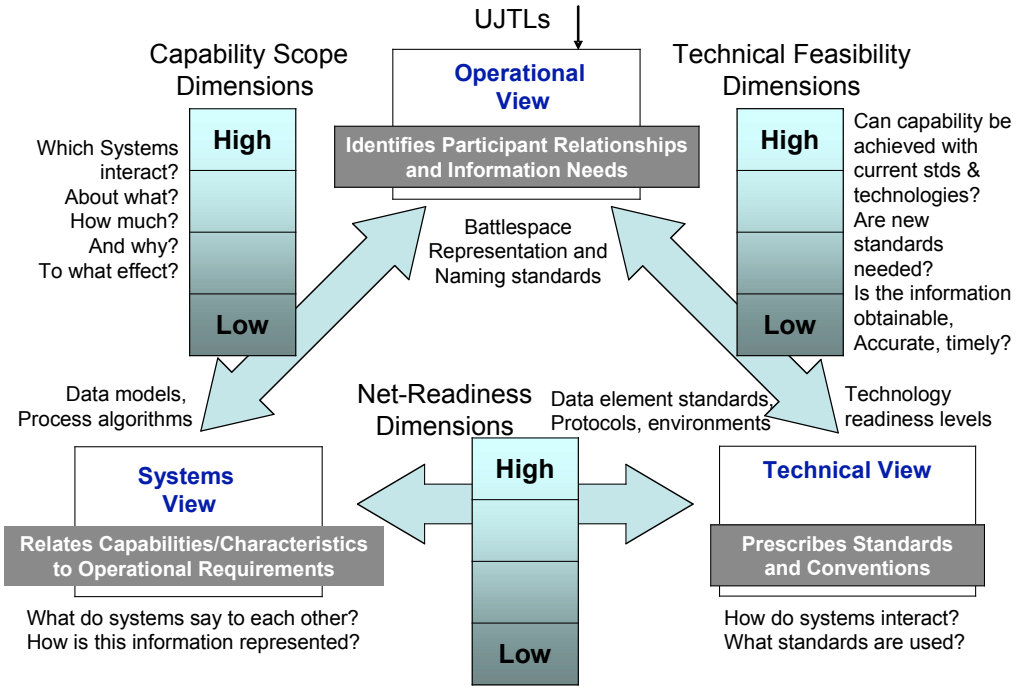


Figure 3-2. DoDAF Views and Capability Assessment Dimensions/Criteria

Dimension Presentation Approach

The SCOPE dimensions and subdimensions are presented in the following. The subsections that follow are structured hierarchically in accordance with the hierarchical dimension structure of the SCOPE model.

A key consideration in using SCOPE as an assessment model is that well-defined value sets should be established for each dimension so that (1) a specific value or range of values can be assigned to an assessed object to characterize that object's properties within the realm of that dimension and (2) any two objects assessed within that dimension are assigned values that are from the same set, and thus, comparable in some sense.

This version of the SCOPE document emphasizes defining value sets for the dimensions that are at the leaves of the dimension hierarchy, so each such dimension has a value set defined for it. In some cases, there are value sets that are applicable across all subdimensions of a dimension, and these are defined in the description of the higher-level dimension.

Higher level dimensions may have values that are in some sense aggregations or "roll-ups" of the values in its lower level dimensions, but this may involve the development of normalized value scales across the model and an associated computational model (with weighting schemes, roll-up rules, etc.) for performing "quantitative" assessments. These issues are deferred to future versions of the model and document.

Each section describing the higher level (nonleaf) dimensions in the current version offers a summary or overview of that subtree, which describes the collective qualities or properties of that subtree. It includes conceptual foundations to aid in understanding the subdimensions, and may also define a general valuation framework used across the value sets in that dimension and its subdimensions.

The intended general structure of the dimension description sections that define value sets is straightforward, although the concepts relevant to each dimension may not be. This structure is outlined in Figure 3-3.

3.x.x….<Dimension Name>

A concise discursive description and explanation of the dimension. This will describe which aspect or aspects of an item under examination is/are encompassed or characterized by this dimension. It essentially defines the "meaning" of the dimension. It also describes or suggests why the dimension is important or useful in its context within the SCOPE model.

Value Type

A simple identification or description of the type of values encompassed by this dimension. For example, the type can be numeric (real or integer) with a specified range, an enumerated set (ordered or unordered), a conceptual continuum of values, a set of ranges or continuums defining discrete categories or levels, and so on. No predefined list of types is prescribed here because they highly depend on the dimensions and the dimensions may vary widely in nature.

Value Set

A description of the value set associated with this dimension, with as much detail and precision as seems feasible and reasonable. This may include information that can help guide users of the model in applying the value set during the assessment process. As noted above, the dimensions and their value types are diverse and there is thus no simple, uniform means of describing the value sets. Values with different types will generally be described in different ways, and values for two dimensions with the same value type may also be described in different ways if, for example, they are described at different levels of detail or to emphasize different issues.

Some value set descriptions are enhanced using graphics where appropriate or useful. These can help illustrate how values are organized within the space and what their relationships are to each other. Other information that more formally describes value interrelationships or the means by which values are derived from an object under examination may also be useful.

Example

An example of how the dimension can be applied in some practical context. The example can be realistic or contrived, but should concisely illustrate how the value set would be interpreted and applied to evaluate some specific item under examination.

Figure 3-3. Dimension Description Structure

Guidance for SCOPE Model Application

The SCOPE model is designed to characterize the collaborative networking potential with technology mediated support through a set of dimensions that describe capabilities at the following architecture intersections:

- Capability SCOPE Dimensions—Operational X Systems Views
- Net Ready Dimensions—Systems X Technical Views
- Technical Feasibility Dimensions—Operational X Technical Views

Areas of Concern When Applying the SCOPE Model

Identifying the areas to characterize with the SCOPE model and considering the user community that is intended to benefit from the SCOPE model analysis, and tailoring the details of analysis to that user community, should guide the level of detail and the concepts characterized. Figure 3-4 illustrates a default mapping between areas of concern and the types of users interested in these areas. This indicates that for most SCOPE analyses there can be multiple applications of the SCOPE model to different areas of concern, thus necessitating the selection of the relevant SCOPE dimensions for each area of concern.

SCOPE Model Process

The following is a recommended process to achieve desired granularity and coverage:

1. Characterize the operational areas of concern through a set of appropriate capability-specific dimensions. These dimensions should indicate the capability to collaborate, to expose and share information, to expose and share services or capabilities, to jointly plan, model, execute collaborative tasks, and areas of concern in the concepts and language of the application domain

2. Identify areas of concern that are of most interest to SCOPE analysis and prepare to specify specific Net-Ready, Capability Domain Independent, and Technical Feasibility dimensions for each area of concern, according to Figure 3-4.

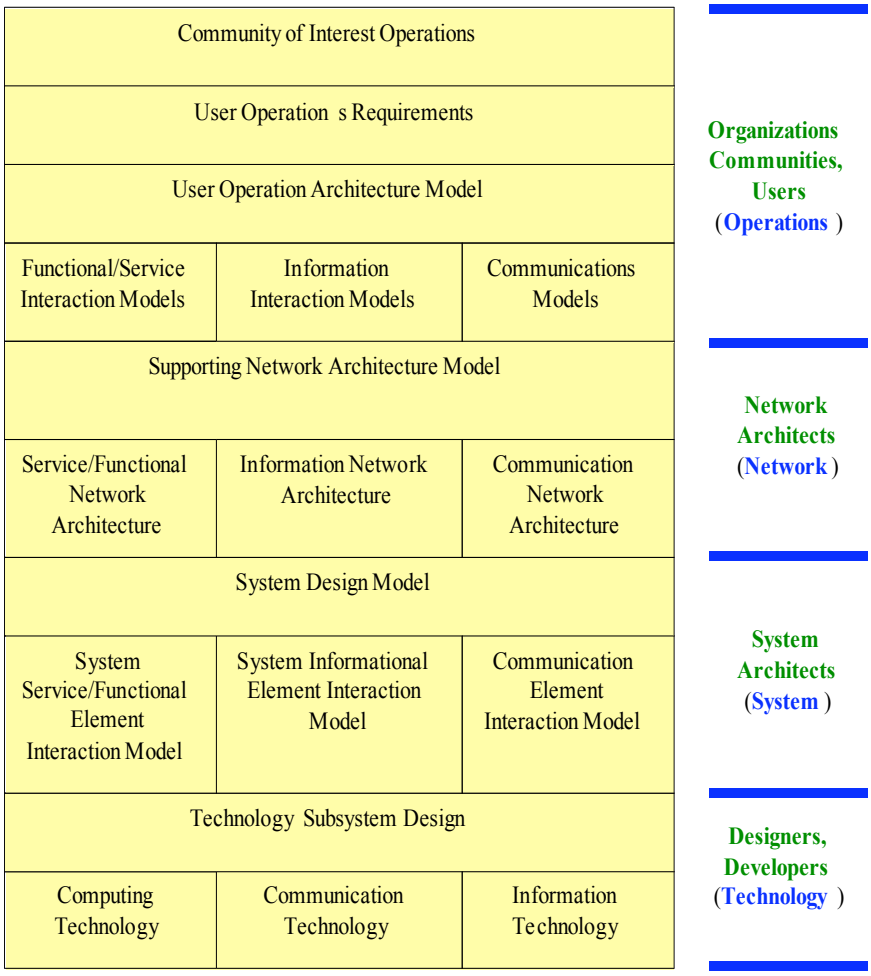


Figure 3-4. Areas of Concern, User Interest, and Relevant SCOPE Dimensions

3. Apply the SCOPE model analysis to each area in the following sequence:
   - Capability Domain Independent dimensions
   - Net-Ready dimensions
   - Technical Feasibility dimensions

SCOPE Model Dimensions and Independence

In this version of the SCOPE model, interrelationships between dimensions were not specified. Future versions will identify the degree of interdependence in context with the area of concern. For example, the Inter-Element Time Binding Sensitivity dimension defines the maximum amount of time in which a service requestor and provider must bind to each other and interact in order to effectively provide an operational capability. The Network Latency dimension defines the minimum amount of time it takes to make a network round trip between two nodes. These two dimensions are interdependent in the sense that longer Network Latency makes capabilities requiring shorter Inter-Element Time Binding Sensitivity infeasible. Thus, when evaluating whether a capability is technically feasible in a net-centric environment, the SCOPE evaluator may need to consider multiple dimensions and their interrelationships and mutual impacts.

In the current version, when using SCOPE to evaluate some capability, it is best to select those SCOPE dimensions relevant to the capability of interest and then to proceed in a manner that accounts for any effects of selections across dimensions to ensure that values selected across these dimensions are consistent with the overall evaluation for the capability. The evaluator is warned that care must be taken to ensure that the SCOPE evaluation is consistent for the capability.

3.1 Net-Readiness Dimensions – 1.0

The Net-Readiness dimensions measure system attributes that may support multiple capabilities. In NCO environments, these attributes may be offered as services on the GIG, built on top of NCES and COI-wide services by a given program, and in other cases affect the design to incorporate models and languages for explicit semantic representation of exchanged data. Other environments may leverage the World Wide Web (WWW) service-oriented architecture to broaden public access and discovery. Such services can be made discoverable on the GIG by other systems and service providers such as NCES Discovery Core Services. Even if the GIG and NCES services are not available to a given program, it can still be assessed based on the degree to which it offers discovery services on its own. If a system is isolated from the GIG for some acceptable reason, it can still be assessed in terms of its inherent "net-readiness." Interoperability with coalition partners on the Combined Enterprise Regional Information Exchange System (CENTRIXS) network might be an example of this situation.

The Net-Readiness dimensions are intended to capture the major characteristics of each system or system of systems as seen from the perspective of other systems on the network. The notion of net-readiness incorporates the network-centric service-oriented architecture paradigm and the inclusion of various forms of metadata and other technology to make information more understandable and discoverable in an open network environment.

3.1.1 Discovery – 1.1

Discovery capabilities offer mechanisms for locating resources that are relevant to client needs or interests. Discovery of resources such as services, information, and people is an important enabler for net-centricity. This becomes evident as the scope of the enterprise or context in question becomes larger and the number of entities to be discovered increases to a point requiring assistance. For large organizations that are procedurally and technically complex, automated discovery capabilities are essential to promote net-readiness. There are three interdependent aspects to discovery: (1) visibility of resource information to potential users of those resources, (2) location and relevance of those resources to potential users, and (3) vccess to the resources to support different applications.

Discovery capabilities support the acquisition and location of reference information about available resources. Such information is stored in directories, catalogs, or registries on a network, which are organized by, and queried, using metadata that describes and indexes the information. The dimensions described in the following address discovery needs and capabilities that span a spectrum of sophistication.

3.1.1.1 Service Discovery – 1.1.1

Service resources offer specific, defined functionality to clients. These resources can be viewed as automated computer services, although any given computer-based service may be a proxy for a human or other kind of "real world" service (e.g., a computer service may be used to request or reserve a human service).

As noted previously about discovery in general, service discovery has three complementary aspects: making resources visible, locating visible resources, and supporting a range of discovery times. Time in this context relates to when information is available to aid in the discovery process. Thus, any assessment of service discovery capabilities must take these aspects into account. These are captured in the subdimensions described in the following.

Table 3-2 offers a perspective on how combinations of values within the three subdimensions can be interpreted. These combinations of values within the Service Discovery subdimension continuums can represent discovery capabilities, ranging from no discovery at all to fully dynamic discovery of services and their semantics with each service invocation. The most discoverable approach is a metadata-rich discovery repository, which has metadata with high semantic content in operational domain relevant to the searching community. Metadata is always about a resource and as such has clear location information about locating a resource.

Table 3-2. Service Discovery Value Set

				Service Description Publication/Access Mechanisms			
Service Descriptions	**None**	**Unmanaged Hardcopy or Personal Knowledge**	**Hardcopy Library With Simple Catalog**	**Electronic File Directory**	**Web Page**	**Web Page With Limited Metadata**	**Metadata-Rich Discovery Repository**
None	No discovery						
W/no metadata		Ad hoc person-to-person discovery	Random discovery via browsing	Local targeted discovery via browsing and text searching	Remote targeted discovery via browsing and text searching		
Keywords/ indexes		Ad hoc person-to-person discovery	Targeted discovery via catalog	Local targeted discovery via browsing and text searching	Remote targeted discovery via browsing and text searching	Limited automated discovery	Probably little to no discovery due to metadata type mismatch
Basic structured metadata		Ad hoc person-to-person discovery	Targeted discovery via catalog	Local targeted discovery via browsing and text searching	Remote targeted discovery via browsing and text searching	Limited automated discovery	Automated discovery, relevance assessment, comparison, and invocation

Semantic metadata		Ad hoc person-to-person discovery	Targeted discovery via catalog	Local targeted discovery via browsing and text searching	Remote targeted discovery via browsing and text searching	Limited automated discovery	Semantically-rich automated discovery, relevance assessment, comparison, and invocation

= Design Time = Design/Configuration = Run Time / = No/limited applicability

3.1.1.1.1 Service Description Richness - 1.1.1.1

One or more descriptions of the service can be consulted in a discoverable service environment to identify it and determine if it is appropriate to client needs. In more advanced forms, these descriptions are defined using metadata, and typically include a description of the function the service performs, the information that is provided to the service, and the information the service produces. Such descriptions also may include the context in which the service is applicable; preconditions, post-conditions, and constraints that may apply to the service; error conditions that the service can sense; formal semantic descriptions of the behavior of the service and/or the data associated with the service; and so on. Any or all of this information can be used to help support the task of finding candidate services and determining whether or not they satisfy specific client needs.

Value Type

Conceptual Continuum

Value Set

Table 3-3. Service Description Richness Value Set

Value	Description
No description	No services are described.
Description with no metadata	Free text service descriptions with no metadata to facilitate indexing and searching.
Description with keywords/indexes	Free text service descriptions accompanied by simple metadata such as keywords or index identifiers to facilitate searching.
Description with basic structured metadata	Service descriptions accompanied by structured metadata (e.g., WSDL, XML tags) defining basic characteristics of service.
Description with semantic metadata	Service descriptions accompanied by structured metadata incorporating semantic information (e.g., via RDF, OWL) defining detailed characteristics of service and enabling semantic interpretation of information.

Example

A system that incorporates basic Universal Description Discovery and Integration (UDDI) registry capabilities with WSDL descriptions offers service descriptions with basic structured metadata.

3.1.1.1.2 Service Description Publication/Access Mechanism – 1.1.1.2

A variety of mechanisms can be used for representing the service descriptions, as well as storing and querying for visibility and location. Service discovery occurs in nonnet-centric as well as net-centric environments, and the mechanisms used to support it can be as prosaic as paper and pencil. It is axiomatic that service discovery is an important enabler for achieving the dynamic coupling

characteristic of net-centric environments, and the service discovery mechanisms will thus be more sophisticated in such environments.

Value Type

Conceptual Continuum

Value Set

Table 3-4. Service Description Publication/Access Mechanism Value Set

Value	Description
No mechanisms	No mechanisms are used to store, publish, and access service description information.
Unmanaged—hardcopy or personal knowledge	Service descriptions are in form of unmanaged personal knowledge or hardcopy documents.
Hardcopy library with simple catalog	Service descriptions are in hardcopy form, housed in some form of library with a simple catalog enabling search and discovery.
Electronic file directory	Service descriptions are stored in electronic form in a hierarchical file directory structure on a local or shared hard drive, which can be browsed or searched.
Web page	Service descriptions are stored as web pages on a network site accessible from afar, which is amenable to browsing and free text search
Web page with limited metadata	Service descriptions are stored as web pages on a network site accessible from afar, where browsing and searching are facilitated by metadata involving keywords and simple tags.
Metadata-rich discovery repository	Service descriptions are stored in an electronic repository (e.g., UDDI) on a network site accessible from afar, which enables sophisticated queries based on rich metadata.

Example

A system that incorporates basic UDDI registry capabilities with WSDL descriptions offers service descriptions published in a metadata-rich discovery repository.

3.1.1.1.3 Service Discovery Time – 1.1.1.3

Today, most systems tend to "discover" services at design time. Even this primitive process is hampered by the lack of widely accessible repositories of service interface specifications. Standard discovery services, such as those by NCES, will allow publication of service descriptions and permit run-time discovery of the published services to enable dynamic coupling. The effectiveness depends on the richness of the service description and whether the query for a service can be matched to the description. Over time, it will become increasingly commonplace for systems/capabilities to select the most appropriate service provider for a given mission or capability instance at run time (point of use). Technology standards like the Web Ontology Language (OWL) will be used to extend the service description so that the metadata will have concepts defined for a specific application domain. These standards will provide enough information about services in service directories to permit dynamic reasoning and coupling to specific services best suited to the users' needs.

Value Type

Conceptual Continuum

Value Set

Table 3-5. Service Discovery Time Value Set

Value	Description
No discovery	There is no time at which service discovery is performed.
Discovery at design time	Service discovery is performed as service client processes are being designed, and discovered information is used to select which services will be invoked when client processes are installed and executed.
Discovery at configuration time	Service discovery is performed during installation and configuration of potential service client processes (either by those processes or through manual intervention) based on execution environment attributes, and discovered information is used to select which services will be invoked in future, up until reconfiguration.
Discovery at run time	Service discovery is performed dynamically by potential service client processes while those processes are running, and discovered information is used to select at run time which services to invoke.

Example

A system that incorporates basic UDDI registry capabilities with WSDL descriptions offers service descriptions that are readily discoverable at run time.

3.1.1.2 Information Discovery – 1.1.2

Information discovery success in an open, dynamically changing technology-based network is strongly dependent on a variety of concepts, including the following:

1. Availability and discovery of metadata to describe the information.
2. Conformance to net-centric tenets to have all data described by metadata and a standard metadata language.
3. Relevance of the metadata to a community and its domain-specific terms and language for representing its domain information.
4. Level of semantic expressiveness used in the metadata to describe the information.
5. Ability of a network to match queries against stored metadata describing network available information sources complying with the metadata description.
6. Ability to provide evaluation metrics for query-metadata-information source matches.
7. Ability to specify and utilize context information that would bias the query-metadata-information source match.

3.1.1.2.1 Information Discovery Dimension and Values – 1.1.2.1

Concentrating on the net-centric tenet to utilize metadata for information discovery, Table 3-6 provides a method for characterizing each of the eight discovery concepts. Each of these represents an Information Discovery subdimension.

Table 3-6. Information Discovery Dimension Values

Information Discovery Subdimension	Dimension Values
Metadata availability and accessibility	1. Available and accessible
	2. Available, not accessible

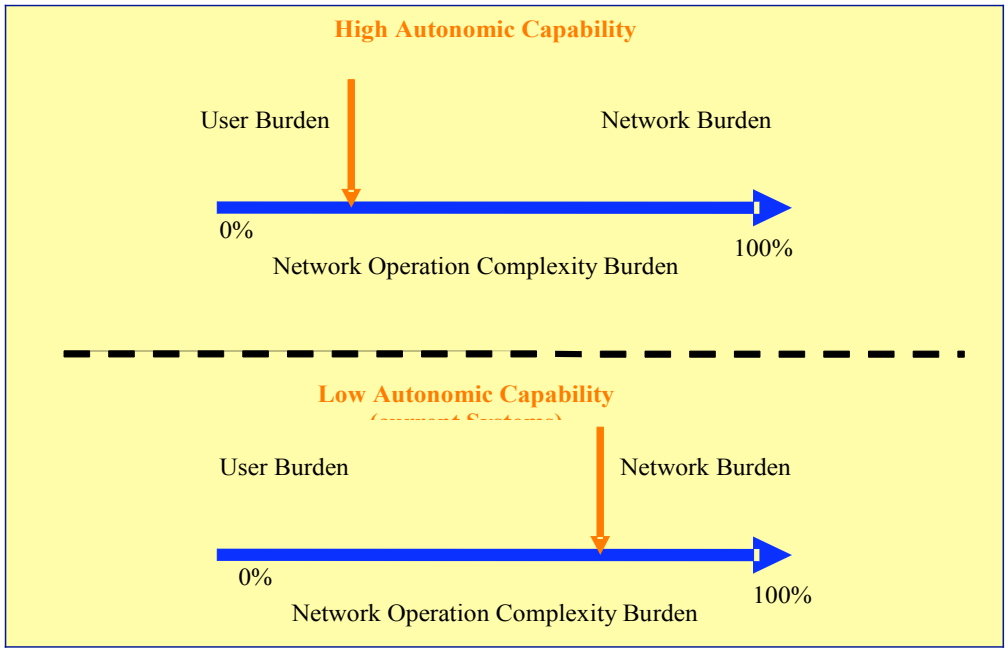
Information Discovery Subdimension	Dimension Values
	3. Not available, not accessible
Standard metadata language	1. Standard metadata language (DC, DDMS, ?)
	2. Unique nonstandardized language
COI metadata relevance	1. Not relevant
	2. Partially relevant
	3. Completely relevant
Query metadata matching capability	1. No mediation, network responds with list based on exact metadata content match
	2. Partial matching with some capability to evaluate multiple matches across metadata keywords
	3. Full semantic match with metadata matching to related semantic domain models associated with metadata content
Multiple information sources relevance metrics	1. No metrics to identify better matches
	2. Simple metrics used for metadata key word matches
	3. Full semantic metrics defining level of uncertainty of information sources against query
Context informed queries	1. No context information defined for modifying query match
	2. Some time, location, role-based criteria used to refine query match
	3. Extensible semantic context model and software agents used to interact with multiple mediation and information services
Level of semantic expressiveness	An emerging subdimension – refer to emerging SCOPE semantic interoperability subdimension on Semantic Expressiveness, Section 4.1.2.7.

3.1.1.2.2 Information Model Pre-Agreement – 1.1.2.2

The Information Model Pre-Agreement dimension represents the degree of pre-agreement and information representation standards compliance among systems interacting on the network. A low level is not necessarily bad, in that it allows more flexibility in information exchange with other systems (like web pages), but it also requires more processing power and sophisticated algorithms to extract or infer the underlying information model from the information exchanged. Higher levels of information model pre-agreement generally permit more efficient exchange of information between systems with a high degree of mutually consistent interpretation to arrive at meaning, but require each participating system to comply rigorously with the same model and knowledge model representation languages, such as RDF or OWL. Lack of pre-agreement also creates significant impact on participating systems if any of the implicit assumptions change, usually in the form of rewriting existing code or implementing/evolving an adaptor layer (if the nature of the change makes it feasible to do so). Some aspects of this also exist in the Service Discovery dimension (because services have information models as well), and in the Capability/Domain-Specific Scope dimensions (because capabilities have an inherent information model that manifests itself in service interfaces).

The more complex or domain-specific the representation of certain information might be, the greater the pre-agreement that might be required to ensure that the same models are used to interpret

exchanged



information, but clearly this will provide the highest degree of mutually consistent interpretation of the meaning of exchanged information.

Value Type

Conceptual continuum

Value Set

The following is a representative set of points within the continuum represented by this dimension:

- *Extensive syntactic and semantic pre-agreement required* – e.g., complex battlespace object such as air tasking order or deployment plan (Ontology domain models with layered business rules, and mapping to common logic and/or upper ontologies)
- *Significant syntactic and semantic agreement required* – e.g., Semantic Web domain ontology
- *Significant syntactic and some semantic pre-agreement required* – e.g., COI domain-specific XML Schema with data element standards (C2IEDM, EDI)
- *–Minimal pre-agreement required* – e.g., ASCII text, URLs (implies greater net-readiness)

3.1.2 Autonomic Networking Dimension – 1.2

NCO is based on a fundamental principle that the positive benefits of networking people, COIs, organizations, systems, applications, data, and services can accelerate decision making and enable optimized resource use and allocation. Counterbalancing these positive effects of networking is the increased complexity (Figure 3-5) of network operations and interoperability, both indicating an increase in the cognitive load required for more complex human interactions with the network. High autonomic network capability places a lower burden on the user, while low autonomic capability, in contrast, places a heavy burden on the user. A proposed solution is to shift some of the cognitive burden to autonomous capabilities within the network as indicated by high autonomic capability.

Figure 3-5. Complexity Burden (High and Low Autonomic Capability)

Types of Networking Complexity

NCOIC has defined three types of networking complexity:

1. Operations complexity
2. Technology noninteroperability complexity
3. Social noninteroperability complexity

The first is a well known phenomenon of increasing operations costs and network brittleness in the commercial computer and telecommunications industry due to complexity of operating and maintaining a network. The second networking problem is defined as technical noninteroperability complexity where increasing numbers of entities in the technology-based network result in information, services, and communications incompatibility. The third networking complexity problem occurs when networks of people and communities attempt to collaborate in new ways.

The autonomic computing traditional approach focuses on the reduction of operations complexity as opposed to the technology or social interoperability complexity problem. This focus for autonomic solutions will be retained for this version of the SCOPE Autonomic dimension. It may be that autonomic concepts can be applied to solving the technology and social noninteroperability problems, but that is not the focus for this version of the document.

Networking Tensions (Benefits vs. Costs)

Wikipedia[2] defines the problem for distributed computer networks as one where complexity is increasing and becoming the limiting factor for further development. For NCO networks, the problem space appears to be identical. It is exacerbated by the extreme dynamic nature of tactical networking environments. Complexity, as defined here, causes an increase in operational costs as a result of the cognitive load associated with using, configuring, managing, protecting, and operating the network

Thus we have a tension between the positive and negative effects of increased networking and interoperability, from an operations, technology, and social perspective. Autonomic solutions are defined here as a means to reduce operations and interoperability complexity, e.g., to minimize the requirement for complex human interactions, thus reducing the cognitive load that negates the positive benefits of networking.

Though there are many sources for this increasing complexity, service networking complexity, information sharing incompatibility, nonoptimal resource allocations for collaboration, noncomplementary network operations activities; the goal here is to define universal characterization information concepts for this autonomic dimension that can be applied to a wide variety of situations and sources of tension.

In addition to supporting fairly static intranet or enterprise environments, NCO networks will also have to operate in dynamic networks of heterogeneous networked elements and resources, and user communities with changing roles and goals, further increasing cognitive complexity.

Due to the open and heterogeneous nature of the NCO environment, operations complexity will be an inherent property and by its nature provide a negative counter force against the positive benefits

[2] http://en.wikipedia.org/wiki/Autonomic_Computing

achievable from networking, reducing the positive effects to A – Net Low Autonomic shown in Figure 3-6.

Early concepts of autonomic computing recognized that the majority of complexity issues were derived not from the underlying hardware, but from the networked computation systems supporting business logic, information processing, information exchange, and services, and from cognitive complexity from operations activities associated with managing and operating such networks. From an NCO perspective, this tension is similar to what has been identified for commercial computer networks and various types of telecommunication networks, e.g., the natural tension between the concept of positive benefits achievable from networking and the concept of negative costs and brittleness due to increased complexity of cognitive operations.

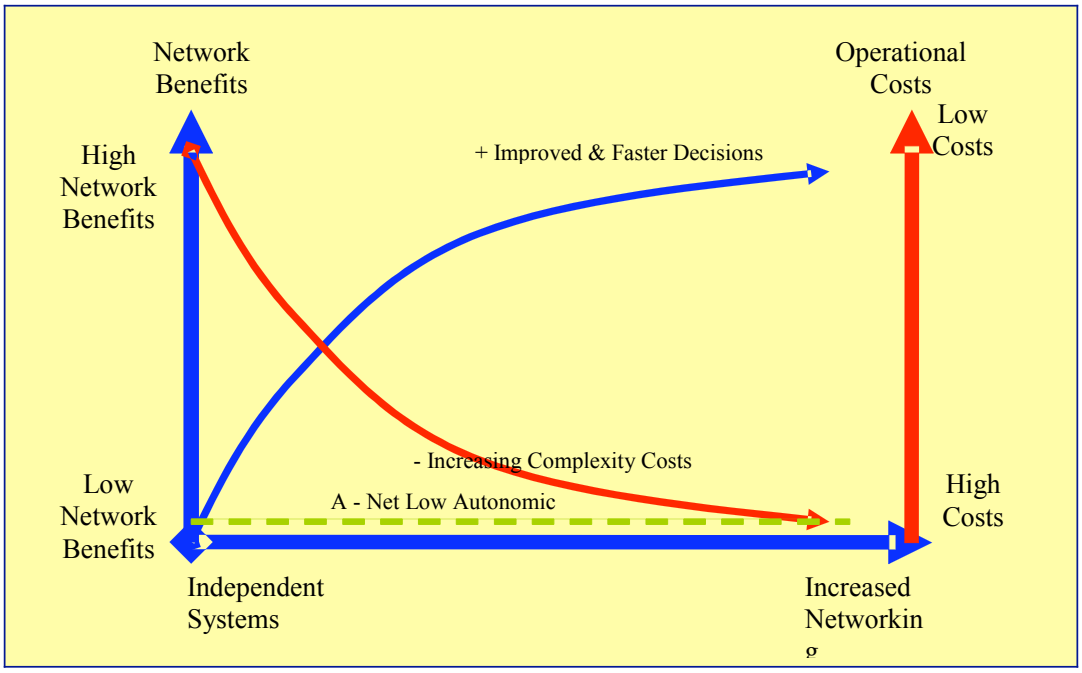


Figure 3-6. Network Tension between Benefits and Costs

Similar concerns and tensions also exist with respect to the transport and communications networks, especially the operations activities necessary to manage them. As the communications network has an increasing and evolving set of services that are offered to applications, and to end users, and also is evolving to support many different types of quality of service (QoS)-supported bearer capabilities and will involve ad-hoc and dynamic configurations, it will become more complex to operate and maintain. Autonomic approaches are also being considered to reduce the complexity of operations activities in communications network.

Reducing the Cognitive Operations Costs

As illustrated in Figure 3-6, autonomic solutions can mitigate the negative effects of increased cognitive complexity of operations by automating some of the activities and by simplifying the human interactions to a type of interaction that is more policy based. Ideally, these interactions are defined at an easily understood human level. The autonomic capability is also based on a more intelligent network, which can interpret the policy based on its internal understanding of what actions to take to achieve the policy. In essence, the human interacts with the network at an appropriate level and the network hides its internal complexity by both removing the need for certain operations activities and

by simplifying the interactions between the human and system. This influences the positive effects of networking to occur as illustrated by the B – Net High Autonomic curve in Figure 3-7.

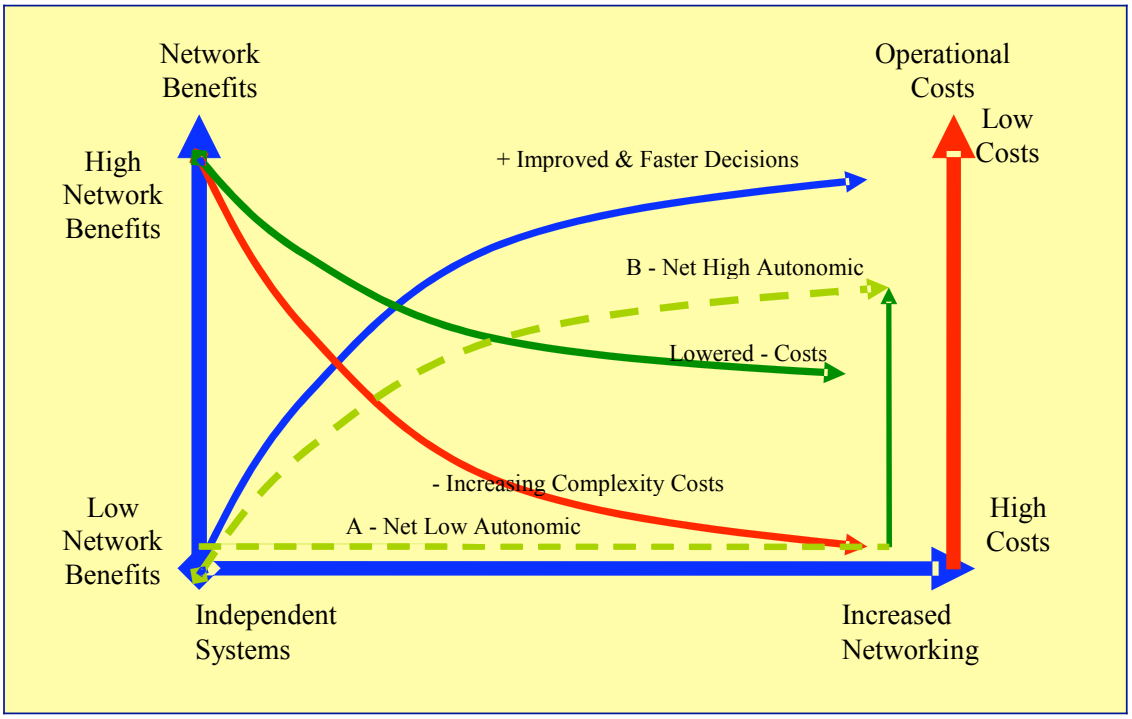


Figure 3-7. Reducing Negative Cost Effects through Autonomic Solutions

The operational negative effects due to complexity derive from manual operations, which are time-consuming, expensive, and error-prone, either in managing the network or in using it in some networked collaborative fashion. If the manual effort needed to control a growing networked computer system is increasing, the availability of an NCO network in dynamic, fast-changing environments will be compromised, resulting in increases in decision time.

Research of operational failures or outages indicate a majority of outages are due to human manual operations, and that "80%[2] happen at the client-specific application and database layer" where the operation procedures are complex and where small errors have large effects.

Autonomic Computing proposes that complexity can be minimized by partitioning certain network functions into semi-autonomous subsystems that are self-managed, and mitigate the need for complex operations. The shift of responsibility for dealing with this complexity in the autonomic concept moves away from the user and toward the network Figure 3-6.

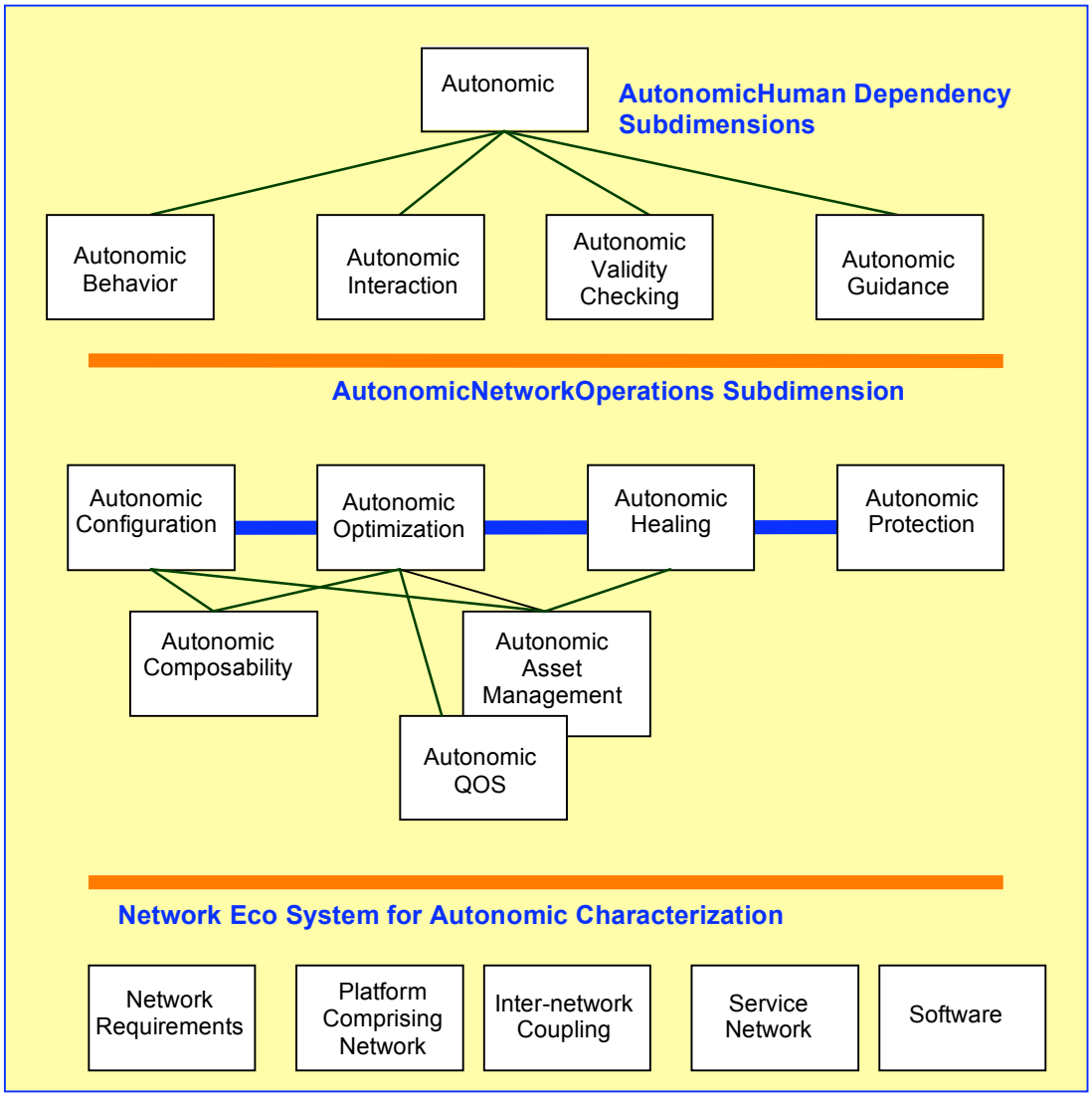
To accomplish this, systems may be autonomic, not requiring "conscious" intervention, which in this sense means not requiring human intervention. The question remains of determining the appropriate balance between autonomic network control and human control. In most cases, it is desirable that the human operator has an overriding capability, a capability to guide the autonomous behavior through establishing rule-based policies.

Other determining factors for this balance determination may be based on the cognitive complexity of making a decision for a complex problem that requires fast response, and where the decisions do not have much variability or are deterministic according to some set of rules or algorithm. Further, it is

proposed that the interactions between the autonomic system and the operator now evolves to a higher level of semantic interaction more focused on the policies and business rules that guide the activities of the self-managed system, thus further reducing the complexity of human interactions with the network.

Autonomic Dimension Model

This



section defines the Autonomic dimension as an information model that can be used to characterize the level of autonomic self management that solutions have and their ability to reduce the negative effects due to increasing complexity (Figure 3-8).

Figure 3-8. Autonomic Dimension Characterization Model

Network Eco System—Elements to Evaluate Autonomic Capabilities

The Network Ecosystem refers to elements of a network that can be evaluated for their ability to support the autonomic dimensions identified in Figure 3-8 and described in the subsections below. The list is provided as a guide, and is not exhaustive. An autonomic analysis should be accomplished at all levels of an architecture, and for all significant involved network elements. The dependency

relationships between the network elements to support autonomic capability should be identified so that a model can be made for each autonomic subdimension. Examples of network areas that could be evaluated for autonomic capabilities include:

1. Network Requirements – evaluation of the requirements specifications for a network to determine description of autonomic dimensions capabilities.
2. Network Platforms – any network element with a specific set of functions and interfaces that could be evaluated for ability to support autonomic dimensions.
3. Inter-Network Coupling – the identification and characterization of autonomic capability to support any reduction of the complexity of dependency coupling between networks.
4. Service Networks – the characterization of the service network according to the autonomic dimension evaluations.
5. Software – the characterization of the software as an entity with its own complexity of management according to the autonomic set of operations.

This sample is oriented toward technology, but other human elements for self-management are also important in an NCO environment, including the ability of organizations and units to minimize the complexity of management through concepts of self-management and self-organization. All of the autonomic dimensions are also applicable to human organizational structures to determine their complexity of operations and how the organization can reduce the complexity of interactions with it by other organizations, especially to their "autonomous" capabilities for self-optimization, self-healing, self-configuration, and other self-management aspects.

3.1.2.1 Autonomic Human Operator Dependency – 1.2.1

The following autonomic human dependency subdimensions are useful for characterizing all autonomic operations:

3.1.2.1.1 Autonomic Behavior – 1.2.1.1

Behavior represents the balance of responsibility for control of operations between a human and the network. Low-level autonomic solutions would require very detailed understanding by the operator of the possible interactions between the network and its underlying platforms for an overall operations activity. The operator would be intensely involved in many different operations activities, from configuration to performance management, failure recovery to protection settings and diagnostics. In low-level autonomic solutions, the user may not have an indication of the need for an operations activity; the user would have to follow an externally defined preventive policy or take an operations activity as a result of some network observable event. High-level autonomic solutions would mitigate the need for operations interactions by automating the response to the network event, or take preventive measures based on preestablished high-level policy guidance. This autonomic solution would have the capability of recognizing different classes of network events and be able to classify and determine thresholds for autonomous operations tasks. Moderate autonomic solutions may involve the operator, but provide significant relevant guidance and high-level information to simplify the operations interactions.

3.1.2.1.2 Autonomic Level of Interaction – 1.2.1.2

Low autonomic solutions would be characterized by very detailed and low-level interactions where the operator would be required to have detailed knowledge and training to determine how to interact with the network for an operations task. The user has to adapt to the system interface, which could be as complex as a common line interface. High autonomic solutions would provide significantly higher levels of interactions, possibly only at a goal or policy level, and the exchanged information would be at a contextual level of the operation and not at the technology level of the system. The interaction uses a standard semantic definition for the operations activity regardless of the technology or product. The network in this case would bear the complexity burden of constructing or selecting the appropriate sequence of atomic interactions and interpreting them at a user level for simpler user interaction.

3.1.2.1.3 Autonomic Validity Checking – 1.2.1.3

In low-level autonomic interactions, the user bears the burden of ensuring that the interactions, responses, and commands are valid for the purpose of the operations activity. The network does not check the validity of the atomic action or the information provided to the network. In high autonomic cases, the network will validate not only the low level action but also the sequence of the action within an overall operations context.

3.1.2.1.4 Autonomic Guidance – 1.2.1.4

Low-level autonomic guidance would require the user to access external information sources to understand how to interact with the network and its underlying platforms. Each platform may require separate external guidance and usage documents. All branching decision points must be understood by the operator after consulting external information sources. High autonomic solutions would provide context-based guidance at a high semantic level, would enable personalization of the guidance based on the role of the operator, and would adapt to a standardized set of semantic information models, regardless of the underlying platforms.

3.1.2.1.5 Autonomic Human Dependency Subdimension Values - 1.2.1.5

Table 3-7. Autonomic Human Dependency Subdimension Values

Human Dependency Subdimension	Subdimension Values		
	Low Autonomicity	Moderate Autonomicity	High Autonomicity
Autonomic behavior	High operator burden	Shared burden	Low operator burden
Autonomic interaction	Detailed interactions (script or command line)	Moderately detailed (GUI interfaces)	Policy-based interactions
Autonomic validity checking	No network operations validity checking	Some network operations validity checking	Full network operations validity checking
Autonomic guidance	No online guidance	Some online guidance	Full context-based guidance

3.1.2.2 Autonomic Network Operations – 1.2.2

The autonomic computing initiative[3] has identified four top-level areas where autonomic concepts can be applied, and where it might be useful to have separate autonomous subsystems defined. A similar

[3] Wikipedia Reference on Autonomic Computing: http://en.wikipedia.org/wiki/Autonomic_Computing

problem of complexity exists for networks and again the autonomic approach is useful for designing and autonomic network[4]. Each of these could be characterized by the four top-level autonomic generic characteristics described above.

3.1.2.2.1 Autonomic Configuration – 1.2.2.1

Configuration operations in a network-centric sense involves those interactions between people and a network, or subnetwork, necessary to bring a set of network elements to a normalized state such that their functionality and interactions with each other will enable desired overall network behavior. These interactions by the operator to configure the network have become more complex as the variety has increased for types of network elements, types of services offered by a network, and as the inter-dependencies between network elements have become more varied and complex. Autonomic configuration solves this problem by:

1. Enabling networks to be self-configuring, thereby obviating the need for some network configuration operations.
2. Reducing the complexity of the operations interaction by redefining it in high level goal-specific terms for configuration, and leaving the detailed interpretation to the network.

3.1.2.2.2 Autonomic Healing – 1.2.2.2

"Healing" refers to the set of operations interactions between the operator and the network that are associated with the reestablishment of a network, subnetwork, or network element from a failed state to a normal state, or at least a useful state. Typically these fault management operations involve:

1. Fault detection
2. Tracing and identifying faults
3. Executing diagnostic tests
4. Correcting faults
5. Reporting and gathering fault statistics and other data
6. Managing the availability state of the failed network element

This particular activity is even more complex than configuration as it involves determination of the cause of a fault, which in an ever increasingly complex network of interdependent elements, results in correspondingly complex set of operations interactions to detect, analyze, and correct faults. Again, the autonomic approach is to mitigate the need for the operation, or to simplify the operations interactions.

3.1.2.2.3 Autonomic Optimization or Performance Management – 1.2.2.3

"Optimization" refers to the set of operations interactions between the operator and the network that are associated with managing the performance of the network, or subnetwork to a desired state. The performance of the network necessarily depends on the real-time demands placed on it by the users of the network, and the relationship between the use of network resources or network utilization. The demand may be quite complex and not easily discernible as to how to reconfigure certain parameters of network behavior to optimize its performance. The increasing varied demands placed on a network and

[4] Wikipedia Reference for Autonomic Networking: http://en.wikipedia.org/wiki/Autonomic_Network

the variety of resources that can be used all increase the complexity of determining the optimized parameter and configuration for expected demand.

Different types of networks have different sets of operations to achieve higher performance, but all have similar goals and similar processes to achieve them. These different types of networks include: Network Performance Management, Applications Performance Management, Business Performance Management, Operational Performance Management, and the newer SOA Service Performance Management.

A similar approach of mitigation and simplification is used by an autonomic approach to the performance management complexity problem. A significant difference here, however, is the need to resolve the problems on different time scales associated with patterns of demand.

3.1.2.2.4 Autonomic Protection – 1.2.2.4

"Protection" refers to the set of operations interactions between the operator and the network, which are associated with detecting, diagnosing, and resolving attacks on the network or its use. With the advent of the Internet and the WWW, the variety and sophistication of attacks on the network and user devices has increased. Typically, the more open the network, the greater the opportunity for attack and thus network technology constantly evolves to meet these threats.

3.1.2.2.5 Autonomic Composability – 1.2.2.5

As illustrated in Figure 3-8, "composability" refers to the ability to organize the network in support of autonomic reconfiguration for different causal situations: reconfigure on failure, reconfigure for performance optimization, reconfigure for network growth, reconfigure for adding new services, etc. The autonomic interest is to determine to what level of complexity a human operator has to interact with the network to compose the structure of the network, and to what extent are the composing interactions mitigated or simplified.

3.1.2.2.6 Autonomic Asset Management – 1.2.2.6

"Autonomic asset management" refers to the ability to manage and allocate assets in support of autonomic optimization or healing operations. The autonomic ability to mitigate or simplify the management of assets would reduce the complexity for asset management operations.

3.1.2.2.7 Autonomic QoS – 1.2.2.7

Autonomic QoS refers to the ability of the network to mitigate or simplify the human-network operations necessary to establish the network level of performance in a specific way. Ideally a fully autonomic network would incorporate the capability for network elements to share information about QoS capabilities, current state, and requirements through a common set of interactions that support QoS negotiation and commitments. With this autonomic capability the human operator can specific policies for QoS using high level business rules defined within the context of the operator's goals, rather than in technical terms of the network.

3.1.2.2.8 Autonomic Networking Subdimension Values – 1.2.2.8

Table 3-8 provides a map of definitions at the intersections of the generic autonomic level values and four major autonomic focus application areas: configuration, optimization, healing, and protection.

Table 3-8. Generic Autonomic Dependency Dimension Values Definitions

Autonomic Operational Dependency Dimension Value	Description
No autonomic—No autonomous behavior, no guidance to user, no checking of valid actions, low-level actions	Only low-level actions are supported by network and complexity of invocation and use if there are any sequence dependencies are burden of user. No data is provided by network itself about how to use service or network capability. For operations activities involving multiple low-level interactions with network, user bears burden of reading and understanding manuals and any state-dependent decisions for determining interactions with network within an overall operations task. These can be normal use of network capabilities or tasks associated with managing network. Information provided by network in response to interactions is at very low detail level and will require training or reading of operations guidelines to understand. User bears the cognitive burden fully. In addition, network may not have any self-checking capability to determine if action request is valid or can be accomplished in its current state, or if user requested action out of sequence from what user guide described.
Some guidance—No autonomous behavior, some guidance to user, no checking of valid actions, low-level actions	Same low-level set of interactions are required, but in this case, there may be ancillary information provided by network that will guide user about what information to give network required by action. May have some help support. User still bears burden of understanding any decision points within sequence of interactions. No checking of valid actions or sequences.
Moderate guidance—No autonomous behavior, some guidance to user, some checking of valid actions, low level actions	Same low-level set of interactions is required, but in this case, there is significant guidance about what actions to take with respect to an operation task or high-level need. Information provided by network in support of this providing guidance on allowed actions at various decision points.
Intelligent adaptive and personalized guidance—No autonomous behavior, full guidance to user, checking of valid actions, still low-level actions	Network provides a high-level interaction for user selection of tasks or desired results, and network will provide a step-by-step guidance for every action and required data to support it as well as checking validity of action request, and enabling user to save previous sequences with user choices retained to be used again to reduce complexity, once a known sequence works or is required by user many time.
Some autonomic behaviors and policy-based high-level control— Partial autonomous behavior, high level guidance to user, checking of valid interactions, high-level actions	Network provides high-level interaction with user and supports setting of policies that will determine networks actions at various decision points. User will be provided high-level response in context of task, and network will automate to some extent detailed actions for task and only interact with user at decision points where there are no policies.
Autonomic behaviors and policy-based high-level control— Autonomous behavior with very little need for interactions at all, high-level guidance to user when required for status checking, checking of valid interactions, high-level actions	Network capability is almost completely autonomic, it takes actions based on perceived environmental state, and predefined policies. User requests are not required to trigger autonomous network behaviors, but they can be. Network will provide information about its autonomous behaviors and conditions before and after behavior for network observation and management purposes. Need for operation activity is almost completely mitigated with this autonomic capability. Only in rare circumstances will user need to interact with autonomic subsystem, and then only to establish operational policies that autonomic system will interpret into its sets of actions.
Intelligent autonomic behaviors and policy-based high-level control— Learning and evolving Autonomous behavior with very little need for	Autonomic subsystem has capability to not only be guided by policies, and not require human triggering or guidance at interaction level, but it now has capability to adapt its actions to its experience with satisfying these policies with different sets of actions and parameter values.

Autonomic Operational Dependency Dimension Value	Description
interactions at all, high-level guidance to user when required for status checking, checking of valid interactions, high level actions	Autonomic subsystem requires very little human interaction, but may provide observation statistics and data to measure its own autonomic performance and quality.

3.1.2.2.9 Autonomic Network Dimension Values Descriptions – 1.2.2.9

Each autonomic network dependency subdimension is further characterized by selecting a set of values (Table 3-8) indicating the level of guidance or dependency on human operations. Each value defined in this table can be selected appropriately for each of the autonomic subdimensions as indicated in Table 3-9.

Table 3-9. Autonomic Network Operations Subdimensions and Values

	Autonomic Values						
Autonomic Subdimension	**No Autonomic**	**Some Guidance**	**Moderate Guidance**	**Intelligent Adaptive and Personalized Guidance**	**Some Autonomic Behaviors and Policy-Based High-Level Control**	**Autonomic Behaviors and Policy-Based High-Level Control**	**Intelligent Autonomic Behaviors and Policy-Based High-Level Control**
Configuration	Manual, low-level, complex actions required to configure network assets, services. External documentation required for guidance. No self-checking by network for invalid actions.	Same set of low-level configuration actions required, but external management system may have some online help support. Still requires training and understanding by operator as to what actions to take to configure network assets. Still no validity checking.	Network interacts with external system to provide significant guidance based on platform types and common information models for platforms, and managed objects, for operations task, through online manuals, and may provide help at various decision points.	Network now uses role-based information of operator to determine what configuration phase and interactions to offer for specific configuration task, and also may provide different levels of guidance personalized to operator training level. User will be walked through what must be accomplished at each step. Still low-level actions and possibly platform-specific interactions.	Network uses operator-specified policies to determine some autonomous behaviors to reduce complexity of interaction. It will translate high-level interactions for a configuration operation into platform-specific low-level interactions and hide these from user. Each high-level step is validated and confirmed by network low-level action results.	Many configuration operations are fully automated and only require request form operator. Confirmation is provided and view of configured results is also provided. Some reconfigurations for optimization or failures do not require an operator event, but will happen automatically in concert with optimization or healing.	Network now has capability to perform many different types of configuration operations across many different types of assets for different purposes. It in fact uses autonomic configuration to perform self-optimization of asset usage and has learning capability to determine optimal configurations for predicted network states based on past experience.
Healing	Low-level alarms, little support for diagnostics, separate management system required, no guidance for diagnostics, any redundancy requires manual control actions, no validity checking of actions. Most respond to failure cause alarms and require human	Significantly complex failure management environment, and little autonomous response by system to failure events, though detected. Network will provide some guidance on causes for failure, and recommended actions that are platform specific.	Much online guidance provided, but requires user to observe network state parameters and values to determine state of failed asset, and what category of actions to take based on analysis.	Network provides specific guidance for the specific detected failure event, and guides user actions for enabling healing. Role of user is taken into account with respect to guidance given, and level of validity checking for action completion.	Network responds automatically to failure event and provides a high-level interface to operator to enable simplified healing operations activities. Certain sequences of failure management operations activities are now automated and are guided by policies with	Network attempts to maintain network viability and performance after a failure by automatically diagnosing failure type and possible actions, and either repairing failed element, or reconfiguring network assets to maintain valid operational state.	Network improves its failure autonomous management through self-learning and adaptive behaviors to results observed with past experience.

	Autonomic Values						
Autonomic Subdimension	**No Autonomic**	**Some Guidance**	**Moderate Guidance**	**Intelligent Adaptive and Personalized Guidance**	**Some Autonomic Behaviors and Policy-Based High-Level Control**	**Autonomic Behaviors and Policy-Based High-Level Control**	**Intelligent Autonomic Behaviors and Policy-Based High-Level Control**
	intervention.	No specific online guidance during operations activity.			respect to any branching decision points.	Guided by policy-based rules and infrequent user high-level interactions in situations where self-knowledge of system cannot conclusively diagnose problem or cannot heal itself with computer operations.	
Optimization	Statistics provided to management system, but interpretation left to knowledge of operator to understand statistics, and what actions to take to optimize performance, and allocation of assets. Little flexibility if any for specifying how system allocates resources effectively, any optimization hidden from operator.	Network provides sufficient information and interaction to provide some guidance with respect to what is possible to modify to achieve optimization of resources or other network quality of service characteristics. Interactions are platform specific, and require extensive training and knowledge of external specifications. Some help provided but minimal.	Extensive online guidance provided for optimization operations, but user has to determine when to accomplish operations for optimization. Performance measures and information may be provided to assist with this decision. Online guides may be provided. Some indications of what areas require optimization operations may be indicated to operator.	Significant guidance provided and personalized for role of user, and operations tasks. Provides guidance at each step as to necessary actions across platforms, takes network level perspective of optimization activity. Still low level actions but context guidance simplifies error prone activity. Network will verify that action is appropriate within overall sequence and validate conclusion of action.	Optimization responsibility starts to shift to network, provides common semantic high-level interface across platforms for optimization activities, and automates some of lower level action sequences, provides feedback on completion, and enables user to set high-level policies that will guide any autonomous behavior and modify guidance on activities and settings.	Optimization responsibility fully shifted to network with only infrequent need for operator interactions for setting policies and high-level parameters, and for tracking performance. Network will notify user of optimization anomaly and need for operations activity that cannot be solved through software applications or services.	Autonomic subsystems have intelligent learning capabilities to optimize their autonomic optimization behavior to achieve better results for specific environments and configurations.
Protection	No observation capability for identifying potential security problems. Few actions available for setting security levels for users, asset use, or system response to protect itself against malicious threats.	Some online help provided for knowledgeable operator regarding protection capabilities and properties of network.	Significant online guidance and manuals for knowledgeable operator. Some protection threat detection capability with statistics to aid diagnosis of threat.	Event-triggered support for identifying needed protection operations, and guidance about what actions to take.	Network has some autonomous behaviors that attempt to detect, isolate, and remedy threats. User can set policies for each of these operations aspects for protection.	Many critical elements of network are protected by multiple self-protection capabilities. Operators can establish overall protection business rules and settings for triggering autonomous behaviors. High-level information is provided with drill-down capability to simplify protection observations and results of autonomous actions.	Protection system learns from experience various patterns of threats and is able to select appropriate measures for each threat pattern. Files are updated periodically to update algorithms business rules, and threat patterns. Machine learning improves threat classification and detection.

	Autonomic Values						
Autonomic Subdimension	**No Autonomic**	**Some Guidance**	**Moderate Guidance**	**Intelligent Adaptive and Personalized Guidance**	**Some Autonomic Behaviors and Policy-Based High-Level Control**	**Autonomic Behaviors and Policy-Based High-Level Control**	**Intelligent Autonomic Behaviors and Policy-Based High-Level Control**
QoS	Statistics provided to observe end- to-end QoS performance, but interpretation left to knowledge of operator to understand QoS statistics, and what actions to take to modify QoS performance, and allocation of assets. Little flexibility if any for specifying how system allocates resources effectively to achieve QoS.	Network provides sufficient information and interaction to provide some guidance with respect to what is possible to modify resource allocation to achieve desired network QoS characteristics. Interactions are platform specific, and require extensive training and knowledge of external specifications to understand how node parameters and protocols effect end-to- end QoS. Some help provided but minimal.	Extensive online guidance provided for QoS management operations, but user has to determine when to accomplish operations for QoS management. QoS performance measures and information may be provided to assist with this decision. Online guides may be provided. Some indications of what areas require adjustment for QoS goals through operations task may be indicated to operator.	Significant guidance provided and personalized for role of user, and QoS operations tasks. Provides guidance at each step as to necessary actions across platforms, takes network level perspective of QoS. Still low level actions but context guidance simplifies error prone activity. Network will verify that action is appropriate within an overall sequence and validate conclusion of action.	QoS management responsibility starts to shift to network, provides a common semantic high-level interface across platforms for QoS operations activities, and automates some of lower level action sequences, provides feedback on completion, and enables user to set high level policies that will guide any autonomous QoS behavior and modify guidance on activities and settings.	QoS management responsibility fully shifted to network with only infrequent need for operator interactions for setting QoS policies and high level parameters, and for tracking QoS performance. Network will notify user of QoS anomaly and need for operations activity that cannot be solved through software applications or services.	Autonomic subsystems have intelligent learning capabilities to optimize their autonomic QoS behavior to achieve better results for specific environments and configurations.

Rationale for Order of Autonomic Characterization Concepts

Although it is possible to create different ordered lists of the increasing values for the characterization, the approach considered here moderates the evolution of autonomic solutions from one of providing better guidance early on to one of autonomous behaviors later on. The first phases will most likely add intelligence to provide increasing guidance and relevant high-level interactions to reduce the complexity of operations, while later phases will increasingly shift the burden of the operation itself to the network with semi-autonomous to autonomous behaviors ,while still maintaining high-level user policy control over the intent of the operation.

Based on this model for autonomic and the SCOPE model theory document concepts, we will not define a top-level set of values or concepts for the Autonomic dimension, but rather define the concepts for the subdimensions. There are, however, generic concepts that illustrate the relative responsibility between the user and the network. These definitions of increasing autonomy are defined in the following, and are related to the concept illustrated in Figure 3-8.

3.1.3 Information Assurance Capability – 1.3

This dimension is intended to measure the degrees of information assurance mechanisms that are required to effect the desired protection levels among systems supporting a specific net-centric capability or system of systems. Alternatively, it can be used to measure the existing mechanisms in a given set of systems, independent of any specific protection goals. Protection goals are driven by the nature of the operating capability (e.g., safety criticality or use of lethal force), and the openness and

degree of objective alignment among participants in a given capability , system of systems, program or enterprise, as measured by the appropriate Capability Scope dimensions in Section 3.2, primarily, Section 3.2.4.

The focus here is not on protection mechanisms typically used inside a particular platform or system boundary, but rather on the protection mechanisms applied to interactions over the network. Thus, emphasis is placed on network-based identity management and authentication, user permission services, transport encryption and digital signature mechanisms, transaction audit trails, and the like. That's not to say that platform/system mechanisms are un-important or that they might not impact the trustworthiness of network security mechanisms. However, other forums and information assurance approaches are already addressing platform security mechanisms.

The SCOPE WG has developed an initial model for measuring this dimension, but there is as yet little pragmatic experience with utilizing this dimension in net-centric assessments of actual systems or capabilities. As a consequence, the initial description of this dimension is included under Section 4, Emerging Dimensions, as Section 4.1.1. The SCOPE WG intends to evolve this description in future versions of the SCOPE model and welcomes contributions to improving the definition of this dimension to facilitate its use in net-centric assessments.

3.1.4 Semantic Interoperability – 1.4

Semantic Interoperability has two major components or aspects, based on their grounding in the institutional or cultural domains that need to be spanned to achieve some network centric capability, or in the technical mechanisms required to convey the different institutional semantics among systems over the network. This dimension addresses the latter aspects of semantic interoperability. Because the other aspect of semantic interoperability is grounded in the nature and scope of the institutions that wish to interoperate, rather than in technical/system mechanisms, it is addressed by a separate set of dimensions under the category of Capability-Independent Scope Dimensions (see Section 3.2).

The semantic interoperability mechanisms dimension is intended to measure the degree to which systems make their semantics and information models explicit and discoverable/accessible over the network. It is also intended to measure the degree to which systems use these mechanisms to find, understand, and access other system's services and associated information and to use it correctly. Typically, this involves use of technologies such as metadata repositories and explicit use of semantic technologies such as ontologies and ontology mapping tools and services. It may also involve language translation services and namespace mapping services, especially in multi-national contexts.

The SCOPE WG has developed an initial model for measuring this dimension, but there is as yet little pragmatic experience with utilizing this dimension in net-centric assessments of actual systems or capabilities. In addition, this model was developed before the advent of the NCOIC Semantic Interoperability Framework (SIF) Working Group (WG) under the Specialized Frameworks Functional Team. In retrospect, some of the material included in the model might better fit into a description of semantic interoperability patterns. As a consequence, the initial description of this dimension is included under Section 4, Emerging Dimensions, as Section 4.1.2. The SCOPE WG intends to evolve this description in future versions of the SCOPE model, in cooperation with the SIF WG, and welcomes contributions to improving the definition of this dimension to facilitate its use in net-centric assessments.

3.2 Capability/Domain-Independent Scope Dimensions – 2.0

The Capability/Domain-Independent and Capability/Domain-Specific Scope dimensions are the major focus of operational functional capability boards (FCBs) or similar governance or doctrinal bodies, and reflect the specific system interface attributes that are needed to support the full scope of the desired functional/mission capabilities.

The Capability/Domain-Independent Scope dimensions measure the overall scope of the capability that institutions are planning to implement, while the Capability/Domain-Specific Scope dimensions characterize the key measures of functional/operational performance inherent in a desired capability. While the latter measures are fairly well established in the capability development process and attract a lot of attention from governance bodies, the Capability/Domain Scope measures are potentially more significant from a system architecture and implementation cost perspective.

The Capability/Domain-Independent Scope dimensions are intended to help make more explicit the capability scope decisions that influence system interface requirements and cost. It is important to note that the Capability/Domain Scope dimensions have major impact on domain (e.g., battlespace) object representation and naming conventions, authorities, and standards on the network. This impacts the service interface specifications at the capability level as well as at the enterprise and enterprise services (e.g., GIG/NCES) level. These dimensions also drive the need for information broker services, both within a capability service set and across capability sets.

3.2.1 Enterprise Type – 2.1

This dimension offers a general categorization of the type of enterprise being examined. The distinctions among the Enterprise Type values are based on the fundamental guiding principles or missions underlying the enterprise. There are many ways to categorize enterprises, and this dimension may incorporate additional categorization schemes in the future.

Some of the other Capability/Domain-Independent Scope dimensions described below may vary depending on the Enterprise Type. The Enterprise Type can be used as an index or selector for determining which value set (and possibly which subdimensions) apply in these other dimensions. The Enterprise Breadth dimension is an excellent candidate for such an approach. More complete articulations of the SCOPE model in the future will more fully address such contingencies.

Value Type

Enumerated set

Value Set

Table 3-10. Enterprise Type Value Set

Value	Description
Government or legally-oriented	Enterprise whose primary mission is to govern and whose operations are guided by rule of law.
Defense-oriented	Enterprise whose primary mission is national or regional defense and whose operations are military in nature.
Profit-oriented	Enterprise whose primary mission is to earn for its owners/shareholders.
Nongovernment, nonprofit	Enterprise whose primary mission is to perform some needed function, but without government affiliation and without seeking to earn profit.
Ad hoc	Enterprise that does not cleanly fit into one of above categories. It may

	have a mission that is not crisply defined, that changes over time, or that is not same for all participants/stakeholders. Ad hoc enterprises are often short-lived and may be composed of diverse stakeholders with diverse interests.

3.2.2 Overall Scope – 2.2

The Overall Scope dimension is intended to measure the overall scope of a capability and the enterprise in which it will be employed. The larger the overall scope, the more individual systems are likely to be involved in implementing the capability. This increases the challenge of defining the service interfaces and gaining consensus across all participating systems. In some cases, service interfaces may be segmented into different security and functional domains as the overall scope increases, further complicating service interoperability to achieve the desired capability.

Overall scope is characterized in terms of three subdimensions:

- Enterprise Scale
- Enterprise Heterogeneity
- Enterprise Cohesiveness

3.2.2.1 Enterprise Scale - 2.2.1

This dimension offers a general measure of how large or encompassing an enterprise is. The scale of an enterprise can range from a single thing, such as a person or device, to the known world, and potentially beyond.

Value Type

Conceptual continuum

Value Set

Table 3-11 contains a representative set of values within this continuum in the form of named values and descriptions of those values. The descriptions are augmented with some numerical ranges of enterprise participants that typically might be associated with enterprises whose scale is characterized by that value. The scale tries to capture in simple terms the fact that enterprise scale isn't just based on numbers of people or entities managed by the enterprise, but also its inherent heterogeneity/diversity as well as geospatial and socio-political extent. There is also a close association between the values and the notion of span of control and organizational levels. Future versions of the SCOPE model might consider refinement by decomposing this dimension into multiple subdimensions.

Table 3-11. Enterprise Scale Value Set

Value	Description
Individual person/thing/device	Enterprise consists of an individual entity such as a person or device. *[1 individual]*
Atomic group/organization (project, unit)	Enterprise is a group or organization that consists of a relatively small number of individual entities. *[Single-purpose organization, e.g., 2-400 individuals]*
Larger-scale group/organization composed	Enterprise is a group or organization that is relatively large and consists of multiple smaller groups, collections, or organizations. *[Multipurpose*

of multiple atomic groups/organizations (line of business, military division)	*organization, e.g., 2-40 single-purpose organizations]*
Entire company, agency, service, agency, small government, NGO	Enterprise is an entire large-scale organization such as a large company, small government, large government agency, military service, or nongovernmental organization (NGO). *[Broad-based organization, e.g., 2-50 multipurpose organizations]*
Entire business sector, government, nation, etc.	Enterprise is a very large-scale group or organization such as a large government/nation, an entire business sector, or a large affiliation of organizations. *[National-scale organization, e.g., tens to thousands of broad-based organizations]*
Multiple governments, nations, business sectors, etc.	Enterprise is a very large-scale collection of large-scale groups or organizations such as governments/nations, business sectors, affiliations. *[Multinational-scale organization, e.g., 2-20 national-scale organizations]*
Worldwide	Enterprise encompasses entire world (and beyond, to include space as an enterprise domain). *[Global organization, e.g., potentially all national and multinational organizations, bounded only by the reach and scope of humanity]*

3.2.2.2 Enterprise Heterogeneity – 2.2.2

This dimension characterizes how similar or different the components of the enterprise are. Enterprises can range from a highly homogeneous set of components to a highly heterogeneous set, and this can have implications for ease of interoperability within and outside the enterprise.

Value Type

Conceptual continuum

Value Set

Table 3-12. Enterprise Heterogeneity Value Set

Value	Description
All identical	All components in enterprise are identical
All similar	All components in enterprise are similar
Most similar	Most components in enterprise are similar
Some similar, some different	Some components in enterprise are similar, some are different
Most different	Most components in enterprise are different
All different	All components in enterprise are different

3.2.2.3 Enterprise Cohesiveness – 2.2.3

This dimension characterizes how cohesive the enterprise is. The enterprise can range from a set of independent components (that nevertheless have something in common to merit the term “enterprise”) to a fully integrated whole.

Value Type

Conceptual continuum

Value Set

Table 3-13. Enterprise Cohesiveness Value Set

Value	Description
Independent	Components of enterprise are fully independent and autonomous.
Loose	Components of enterprise are generally independent, but share some goals and processes.
Moderate	Substantial portions of enterprise operate in unison, but many components are independent.
Tight	Enterprise is mostly unified, with some independence.
Fully Integrated	Enterprise is unified whole, operating as a unit.

3.2.3 Enterprise Breadth – 2.3

The Enterprise Breadth dimensions characterize "how much" of an enterprise is "covered" by an entity (or entities) under examination (e.g., system, capability, operation, or program). That is, which aspects, elements, functions, organizations, etc., of the enterprise are addressed, impacted by, or relevant to the entities in question.

There are many possible ways of characterizing Enterprise Breadth. The subdimensions defined in the following offer several perspectives on this topic, and are designed to address a "defense-oriented" Enterprise Type (see the Enterprise Type dimension previously described). Even within this type of enterprise, other means of characterizing Enterprise Breadth are possible, and the set of subdimensions defined will evolve accordingly in the future. The current subdimensions can be viewed as a core set of defense-oriented Enterprise Breadth subdimensions, which can be tailored or extended by the SCOPE user as needed.

It is also possible to define Enterprise Breadth differently for different Enterprise Types because other Enterprise Types have different missions, structures, processes, etc., that can be used as a basis for characterizing breadth. As noted in the description of the Enterprise Type dimension, a future goal for the SCOPE model is to use Enterprise Type as a selector for identifying which value sets or subdimensions within a given dimension should be used in a SCOPE analysis. The SCOPE model, and the Enterprise Breadth dimension in particular, will evolve to define additional value sets and/or subdimensions addressing multiple Enterprise Types.

In several cases, the nominally "defense-oriented" subdimensions defined in the following can be interpreted and tailored to address other Enterprise Types to some degree, as needed and applicable.

3.2.3.1 Operating Concepts – 2.3.1

This dimension identifies which of the DOD Joint Operating Concepts (JOCs) [12] are addressed by the entity under examination in a SCOPE analysis. Multiple values may apply.

The DOD JOCs, Joint Functional Concepts (JFCs) [13], and Joint Integration Concepts (JICs) [14] comprise the set of Joint Future Concepts [15] derived from the DOD Capstone Concept for Joint Operations (CCJO) [16]. The CCJO is the overarching concept guiding development of future joint force capabilities. The CCJO broadly describes how the joint force is expected to operate in the mid-to-long term, reflects enduring national interests derived from strategic guidance, and identifies the key characteristics of the future joint force. It proposes a solution to meet challenges across the range of military operations to guide force development, organization, training, and employment.

JOCs are operational-level descriptions of how a joint force commander will accomplish a strategic mission through the conduct of operational-level military operations within a campaign. JOCs apply the CCJO solution and joint force characteristics to a more specific military problem. More specifically, JOCs identify challenges, key ideas for solving those challenges, effects to be generated to achieve objectives, essential capabilities likely needed to achieve objectives, and the relevant conditions in which the capabilities must be applied.

Value Type

Enumerated set

Value Set

Table 3-14. Operating Concepts Value Set

Value	Description
Major combat operations	Operations concept for future joint force engagement in combat operations with foreign adversaries. Captures most challenging likely adversaries and conditions U.S. may face in next decade against a regional competitor.
Stability operations	U.S. government and coalition partner response when war is thrust on them, and under circumstances including change in political arrangement of an opponent's government that precede, are concurrent with, and follow major combat operations.
Homeland security	Joint force operations to plan, prepare, deploy, employ, and sustain force in future to detect, deter, prevent, and defeat attacks against Homeland, provide military forces in support of civilian authority, and plan for emergencies.
Strategic deterrence	Joint force operations to plan, prepare, deploy, employ, and sustain force in future to contribute to a strategic deterrence strategy set forth by national leadership. Strategic deterrence is prevention of adversary aggression or coercion threatening vital interests and/or national survival.

3.2.3.2 Functional Concepts – 2.3.2

This dimension identifies which of the DOD JFCs are addressed by the entity under examination in a SCOPE analysis. Multiple values may apply.

JFCs describe how the future joint force will perform a particular military function across the full range of military operations. JFCs apply the CCJO solution and joint force characteristics to more specific military problems. They identify the required functional capabilities needed to generate the effects identified in JOCs and identify attributes needed to functionally support the Future Joint Force.

Value Type

Enumerated set

Value Set

Table 3-15. Functional Concepts Value Set

Value	Description
Net-centric operations	Principles, capabilities, and attributes required for joint force to function in fully connected framework for full human and technical connectivity and interoperability. Allows all DOD users and mission partners to share information they need, when they need it, in a form they can understand, to

Value	Description
	achieve information and decision superiority.
Force management	Principles, capabilities, and attributes (and associated policies, processes, and tools) required to integrate human and technical assets from across joint force to make right capabilities available at right time and place.
Force application	Overarching force application capabilities and associated attributes needed to meet future military challenges for major combat operations against large-scale enemy forces. Respond rapidly anywhere around globe, provide overwhelming force to meet any contingency, and be ready to operate in multinational and interagency environment.
Force protection	Means by which DOD agencies, unified combatant commands (COCOMS), and services should plan, integrate, and provide protection to support joint force at point-of-origin through deploying, employing, sustaining, and redeploying across range of military operations.
Command and Control	Seamless, deployable command and control capability, agile across range of military operations, enabled by robust, secure, integrated network and collaborative information environments.
Battlespace Awareness	Means by which commanders and force elements gain ability to make better decisions faster by enabling a more thorough understanding of environment in which they operate, relevant friendly force data, adversaries they face, and nonaligned actors that could aid in or detract from friendly force success.
Focused Logistics	Comprehensive, integrated approach for transforming DOD logistics capabilities and improving quality of logistics support. Transformed logistics capabilities must support future joint forces that are fully integrated, expeditionary, networked, decentralized, adaptable, capable of decision superiority, and increasingly lethal, continuous and distributed, across full range of military operations .
Training	Means for training joint force in agile, net-centric military environment of future.

3.2.3.3 Integrating Concepts – 2.3.2

This dimension identifies which of the DOD JICs are addressed by the entity under examination in a SCOPE analysis. Multiple values may apply.

JICs describe how a joint force commander performs operations or functions that are a subset of JOC and JFC capabilities. JICs have the narrowest focus of all Joint Future Concepts and describe capabilities and decompose them into task level detail.

Value Type

Enumerated set

Value Set

Table 3-16. Integrating Concepts Value Set

Value	Description
Global strike	Capabilities required for rapid global strike against highly challenging targets. Focuses on operations within first 10 days of major combat operations.
Seabasing	Capabilities required for rapid deployment, assembly, command, projection, reconstitution, and reemployment of joint combat power from a sea base across range of military operations.
Joint forcible entry	Operations conducted against armed opposition to gain entry into territory

	of adversary as rapidly as possible to enable conduct of follow-on operations or conduct singular operation.
Joint undersea superiority	Capabilities required for executing operations to establish battlespace dominance in undersea environment. This includes offensive and defensive submarine, antisubmarine, undersea vehicle, and mine warfare operations.
Integrated air and missile defense	Integration of capabilities and overlapping operations to defend homeland and national interests, protect joint force and enable freedom of action by negating adversaries' abilities to achieve adverse effects from air and missile capabilities.
Joint command and control	Command and control capabilities for agile, decisive, and integrated force employment in all phases of combat and supporting operations, addressing rapidly changing scenarios involving complex distributed, simultaneous or sequential operations, often with other agencies and nations
Joint logistics	Integrated, networked, end-to-end deployment and distribution capabilities possessing right capacity and scalability, agility, control, force projection, and time-assurance qualities to effectively support joint force commanders.

3.2.3.4 Swim Lanes – 2.3.4

This dimension identifies which of the major DOD swim lanes are addressed by the entity under examination in a SCOPE analysis. Multiple values may apply. Swim lanes are horizontal functional roles that may be involved in a variety of vertical processes. Typically systems are acquired to support a specific swim lane and are not designed to support the context of other swim lanes. Thus, a system designed for base operations or a system for inventory management has little, if any, interaction with systems in the Warfighting Swimlane, or with each other.

Value Type

Enumerated set

Value Set

Table 3-17 is a set of values representing high-level swim lanes. Although these are assumed to be in a military context, many translate directly or can be adapted to other domains. Furthermore, additional swim lanes can be defined to address a specific domain or context or to aim at a different level of granularity within an enterprise.

Table 3-17. Swim Lane Value Set

Value	Description
Warfighting	Roles and functions associated directly with combat operations.
Transportation	Roles and functions associated with transporting cargo and personnel in support of all enterprise operations.
System life cycle management	Roles and functions associated with conceptualization, acquisition, installation, operation, maintenance, and retirement of systems to support all enterprise operations.
Materiel supply and service management	Roles and functions associated with supply and service of materiel to support all enterprise operations.
Human resources management	Roles and functions associated with management and support of personnel and related resources within enterprise.
Financial management	Roles and functions associated with management and reporting of enterprise finances.
Real property and installation life cycle management	Roles and functions associated with acquisition, installation, operation, maintenance, and disposal of facilities and other real property to support all enterprise operations.

3.2.3.5 DOTMLPF – 2.3.5

This dimension identifies which of the capabilities identified in the DOD Doctrine, Organization, Training, Materiel, Learning and Education, Personnel, and Facilities (DOTMLPF) [17] construct are addressed by the entity under examination in a SCOPE analysis. Multiple values may apply.

DOTMLPF is a framework used as part of the DOD Joint Capabilities Integration and Development System (JCIDS). JCIDS defines processes and mechanisms for identifying capability needs and specifying capabilities to be developed to satisfy those needs. The DOTMLPF framework is used to help analyze functional areas and identify capability gaps and needs in the specific areas that are encompassed by the various DOTMLPF elements.

Value Type

Enumerated set

Value Set

Table 3-18. DOTMLPF Value Set

Value	Description
Doctrine	Fundamental principles that guide employment of U.S. military forces in coordinated action toward a common objective.
Organization	Unit or element with varied functions enabled by structure through which individuals cooperate systematically to accomplish common mission and directly provide or support warfighting capabilities.
Training	Military training based on doctrine or tactics, techniques, and procedures to prepare forces and/or staffs to respond to strategic and operational requirements deemed necessary by combatant commanders to execute their assigned missions.
Materiel	All items (including weapon systems and related spares, repair parts and support equipment, but excluding real property, installations, and utilities) necessary to equip, operate, maintain, and support military activities without distinction as to its application for administrative or combat purposes.
Leadership and education	Professional development via a learning continuum comprising training, experience, education, and self-improvement.
Personnel	Synchronized efforts of joint force commanders and service components to optimize personnel support to joint force to ensure success of ongoing peacetime, contingency, and wartime operations.
Facilities	Real property consisting of one or more of following: building, structure, utility system, pavement and underlying land.

3.2.3.6 PMESII – 2.3.6

This dimension identifies which of the battlespace elements identified in the DOD Political, Military, Economic, Social, Infrastructure, and Information (PMESII) [18] framework are addressed by the entity under examination in a SCOPE analysis. Multiple values may apply.

In modern effects-based operations (EBO) [19] approaches to military planning and assessment, a system-of-systems analysis (SoSA) is performed to assist in understanding the battlespace. The PMESII elements define the functional categories or dimensions used in the SoSA process to characterize vulnerabilities of the adversary or in the operational environment of the focus area.

Value Type

Enumerated set

Value Set

Table 3-19. PMESII Value Set

Value	Description
Political	Functions, capabilities, and issues associated with governing of anation or organization, administration and control of that entity's internal and external affairs, and/or intrigue or maneuvering to gain control or power.
Military	Functions, capabilities, and issues associated with armed forces and application of such forces in offensive or defensive operations.
Economic	Functions, capabilities, and issues associated with production, development, and management of goods and services to establish, sustain, and enhance standards of living and produce material wealth.
Social	Functions, capabilities, and issues associated with human relationships, interaction patterns, modes of organization, mores and customs within society.
Infrastructure	Functions, capabilities, and issues associated with basic facilitating services of society, including water, electricity, sewers, transportation, , and communications.
Information	Functions, capabilities, and issues associated with acquisition, processing, management, manipulation, and dissemination of information

3.2.4 Enterprise Depth – 2.4

The Enterprise Depth dimension focuses on the levels of operational granularity that a given capability service set must support. The set of services interacting with each other to provide the desired capability typically represent operational entities at different levels of abstraction or aggregation. These might be called echelons or organizational tiers in various operational contexts. The key issue that this dimension measures is the number of aggregation and disaggregation transformations that the service set must support. Note that services can be designed to support multiple levels of operational granularity within their interfaces, or a designer could create specialized gateway or mapping services that translate between one level of aggregation and another when requested by other functional services. Hence the use of the term "service set" as the collection of services that exhibit the ability to manage some given range of "enterprise depth".

Mapping enterprise entity representations across aggregation and disaggregation levels create interoperability challenges. Mission application services in one system might characterize the business environment at a level of granularity/detail that is different from the way another application service in another system characterizes the analogous environment. In addition, the same business/battlespace entities may be aggregated in different ways by different enterprise perspectives; for example, the weapon system supporter's perspective (e.g., individual weapon system end items) vice the force employer's perspective (complete weapons systems).

The absolute level of detail/granularity in any one system is less important for net-centric interoperability assessment than is the number of different levels that have to be traversed or mapped between sets of systems supporting a desired capability or enterprise. The greater the number of different levels, the greater is the challenge of representing aggregation mapping rules and the more likely it is that one system will not have the level of detail necessary for another system to act on the

information. Conversely, net-centric assessments of a given collection of systems that have more than one level of detail need to ensure that the mapping rules are visible/accessible on the network, and that the service interaction between the systems is achievable given the aggregation mapping rules.

Value Type

Integer (0 .. n)

Value Set

The total number of organizational/operational representation aggregation levels that must be crossed between systems/services over the network to achieve the capability.

Example

A common approach to representing situational awareness information, especially in naval or air operations contexts is the so-called "track" database, usually fed in some way by some form of sensor/intelligence data sources and a correlator/data fusion function of some type. Such a "common operational picture" is usually focused on representing the battlespace at the individual platform/vehicle level, such as a ship, an airplane, a missile, or an armored vehicle or truck. This is fine for systems operating at the tactical engagement level, usually focused on intercepting and destroying specific vehicle/platform or fixed-site targets.

However, if the desired capability involves situational awareness at the operational force structure level, the problem of mapping individual platform tracks to force structure elements, and vice versa arises. Force structure elements can themselves be represented hierarchically (e.g., platoon/company/battalion/task force/brigade/etc.), in other forms of aggregation (e.g., by country, by type of force such as armor or logistics) or at different levels of time granularity (e.g., daily, hourly, second by second). Typically the systems that capture and represent "track" data are many, forward deployed and tied to near-real time location reporting means, while the systems that capture force structure information are fewer and usually not as current or precise as to location of force elements. More importantly, the latter systems don't typically capture or represent status or location data at the individual vehicle/platform level or operate at the same time granularity. It therefore becomes a real challenge to use the track-oriented systems to feed a situational awareness capability for operational level staff.

For example, how does such a situational awareness service determine the location of brigade X, given the current track data for all the vehicles belonging to brigade X? The aggregation rules for membership of vehicle A in brigade X might be obtainable from some order of battle capability. This becomes more problematic if we want to represent a "task force alpha" to which vehicle A might belong in an operational context, but not in a static force structure context. Another problem in this example that would need to be addressed is deciding how to represent the location of brigade X, given the location of the individual vehicle track data, and how often to update the location of brigade X. The more different levels/types of aggregation that must be supported to create such a situational awareness capability, the more complex this example problem becomes, and the more explicit the aggregation mapping information that will be required on the network to support net-centric agility among the systems involved.

A set of systems that only involve individual track data exchange at generally the same level of time granularity would be scored a "1" on the value scale for this dimension. A capability that requires individual track data to be aggregated into nonhierarchical force elements would be scored a 2, with

multiple force level hierarchies resulting in a concomitant higher score. Decreasing/increasing the time granularity of reporting between the systems from seconds to minutes or minutes to hours would add another level to the score.

3.2.5 Semantic Interoperability – 2.5

Semantic Interoperability entails the networking concept for mutually consistent semantic interpretation of intention and shared knowledge within a situational and purposeful context. This results from a semantic interaction, where *intention, context, and knowledge* are explicitly represented and expressed in some language of discourse or are implied by convention and use. Semantic Interoperability characterizes the compatibility of the descriptive elements of the semantic interaction expression and the corollary representations and models that the interacting agents use to semantically interpret the interaction. The former characterizes semantic interactions, while the latter characterizes the compatibility or congruence of the semantic model that are used to interpret the semantic interactions.

"Semantics" is the meaning of an expression, and its interpretation with respect to a world model, while semantic interoperability is the possibility for mutual understanding when exchanging expressions from each other's world model and background knowledge. Consistent semantics indicates that the meanings of exchanged expressions refer to the same set of real world concepts and objects in equivalent world models for networked agents. Different world models entail different semantics. The problem of semantic understanding of expressions is a real problem among individuals for whom the knowledge of how to interpret exchanged expressions is different. This section on semantic interoperability describes a set of concepts (dimensions) and a set of unique descriptive values for each concept that can illuminate the inherent capabilities for mutually consistent interpretation of interactions for networked agents, where agents could be either human or technological entities. One of the more significant problems inhibiting agreement is the ability to understand one another in a deep semantic sense, as observed throughout history with respect to the axioms that people have in their world models for political, religious, legal, and cultural domains. A parallel situation exists when attempting to interconnect different systems in a network where each system has different assumptions about interactions across their interfaces and where the interpretations of the interactions are not mutually compatible.

The set of Capability/Domain-Independent Scope semantic interoperability dimensions are illustrated in Figure 3-9.

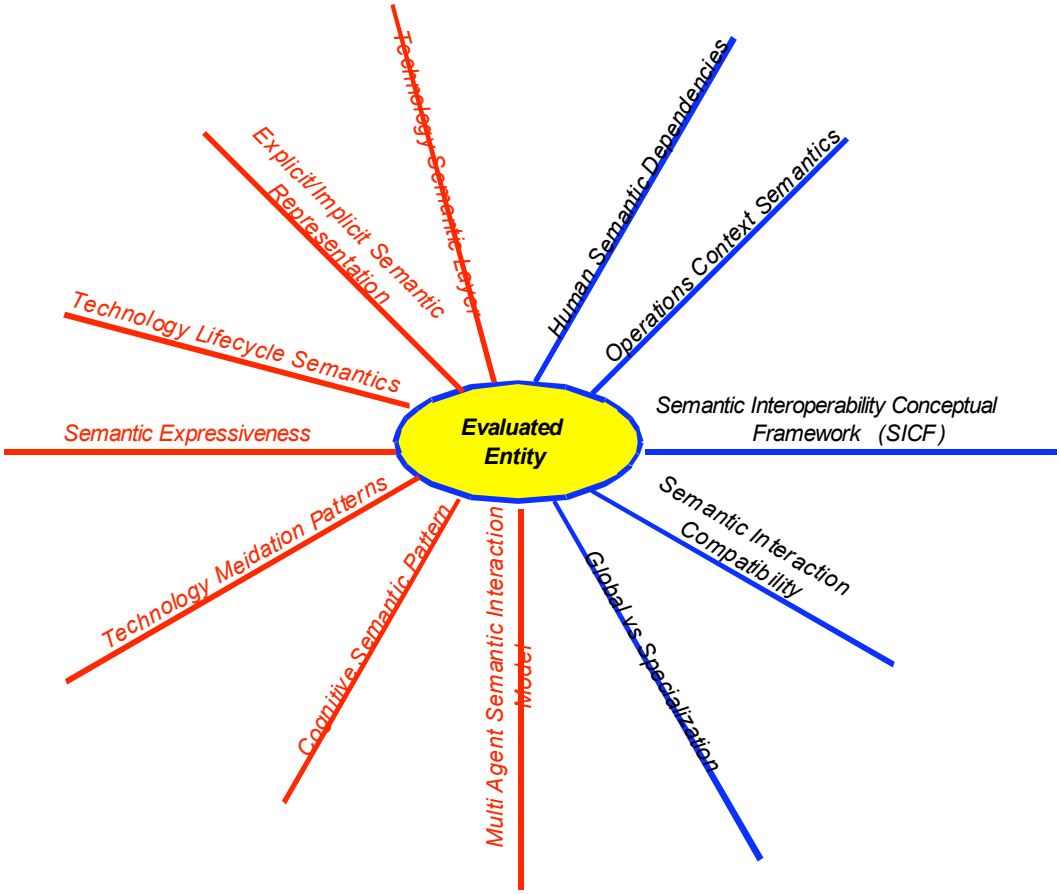


Figure 3-9. Semantic Interoperability Dimensions

Human Semantics of Knowledge and Interactions

Human communicative interactions are constrained by the semantic knowledge defined by human communities within a context, and by the experiential and learned concepts an individual develops through his life (Figure 3-10). Whether the existence of concepts is contained in any agent's world model, the language used to represent knowledge provides an additional obstacle to reach mutual understanding. Other social and context constraints may include additional assumptions about knowledge of the world that would bias the interpretation of any expression.

Language and Cultural Semantic Dependencies

Certain concepts are not readily expressed in the languages of all cultures, requiring the concepts to be expressed as complex expressions in the native language. Even in the case where different languages may have similar concepts, they may have different levels of semantic granularity in a taxonomic sense, or different social constraints as to use and purpose.

This whole area of interpretation of the words or concepts and mappings to the real world has limitations of vagueness[5], where the referent objects or concepts with respect to a word are different, even in the same language and culture by individual members. An example from Quine, "*Insofar as it*

[5] "Word & Object", by Willard Van Orman Quine, The MIT Press, ISBN 0-262-67001-1

is left unsettled how far from the summit of Mount Rainier one can be and still count as "Mount Rainier" is vague."

Semantics, or interpretations of expressions in a language, thus depend on culture, social context, nationality, language, education, experience, and individual inherent capabilities such as intellect, physical attributes for speech, hearing, seeing, smelling, touching, etc.

The human agent exists in both a physical real world and a social conceptual world where the human agent integrates these concepts into meaning. In the literature, this experiential knowledge is referred to as "deep background knowledge," and consists of concepts that most people would understand at some level, and other concepts that are unique to a culture, social environment, and individual experience.

Though human language is almost infinite in its ability to describe concepts of complexity, it may take some time for people to reach a common understanding of the meaning of their expressions. If the domain of discourse is constrained, and most participants have similar experiences in that domain, then a common understanding should be feasible.

We should be able to reach an understanding of what is being expressed with respect to the interpretation of the expression and mapping to real-world objects or social concepts in some world model. What is not necessarily agreed to is the accuracy of the world model with respect to the real world. In most cases people assume that others each have the same world model as their own, which is a source of error. Subtle differences in world models result in similar expressions having potentially different meanings.

The semantic interaction between people will undergo multiple interpretations at different semantic levels as illustrated in Figure 3-10, syntax and grammar of expressions, interpretation to world models of individuals, interpretation to the physical world and human conceptual world. The former real world is mediated by the concepts of science, while the latter are mediated by cultural, political, religious, and personal education and experience.

An approach recommended for characterizing human semantic interactions is to describe the archetypical semantic elements <Context, Domain, and Intent> used by the expressions in the set of semantic interactions being analyzed and the set of world models used by agents to which they refer. Context could be defined by identifying the COI, while Domain could be associated with the different areas of knowledge for a specific role or task within the COI, and the Intent could be associated with the type of semantic interpretations typical for the set of semantic interactions being analyzed, e.g., Request Action, Share Knowledge, etc.

Semantic Interoperability Model Navigation

It is possible to characterize the semantic compatibility of network interactions using a model consisting of multiple dimensions and characterization values for each dimension. A taxonomic view of the Capability/Domain Scope Semantic Interoperability dimensions is provided here.

- Semantic Interoperability Taxonomic Subdimension Structure
    - Semantic Interoperability Conceptual Framework (SICF) (Section 3.2.5.1)
        - Common Environment Knowledge—Physical

        - Common Environment Knowledge—Social
        - Agent Classification
        - Agent Context
        - Agent Domain Knowledge
        - Agent Intentions—Speech Acts
    - Semantic Interaction Compatibility (Section 3.2.5.2)
        - Semantic Interaction Model Compatibility (<*context, domain knowledge, intention*>)
        - Language Compatibility
        - Intentions Compatibility
        - Domain Knowledge Compatibility
        - Context Compatibility
        - Collaboration Compatibility
    - Human Semantic Dependencies (Section 3.2.5.3)
        - Individual Role Dependency
        - Social/Cultural Background Knowledge Dependency
        - Organizational Mission/Focus Dependency
        - Language Dependency
        - Domain Knowledge Dependency
        - Situational Context Dependency
    - Operational Context Subdimensions (Section 3.2.5.4)
        - Time Context
        - Object Situational Context
            - Geospatial Location Context
            - Domain Classification Context
        - Multi-agent Context (Shared Situational Context Knowledge)
            - Collaborative Relationship
            - Complementary Roles
            - Agent Capability Representation
            - Agent Capability Representation
            - Agreement/Commitment Protocol
            - Environment Domain Knowledge
    - Global Versus Specialization of Domain Knowledge for Communities of Interest (Section 3.2.5.5)

The following framework is recommended as a basis for describing the semantic interoperability of networked entities, regardless of the type of entity. The interacting entities could be human or technical, e.g., clients of web services, application software, data exchange protocols, message protocols, overall system interfaces, network protocols, data structure specifications, human organizations, communities of interest, individuals with a specific role, etc.

3.2.5.1 Semantic Interoperability Conceptual Framework Dimension – 2.5.1

A SICF (Figure 3-10) describes a model that can be used to understand and characterize key aspects of semantic interoperability between interacting entities and their ability to achieve mutually consistent interpretation of the meaning and intent of the knowledge shared between them.

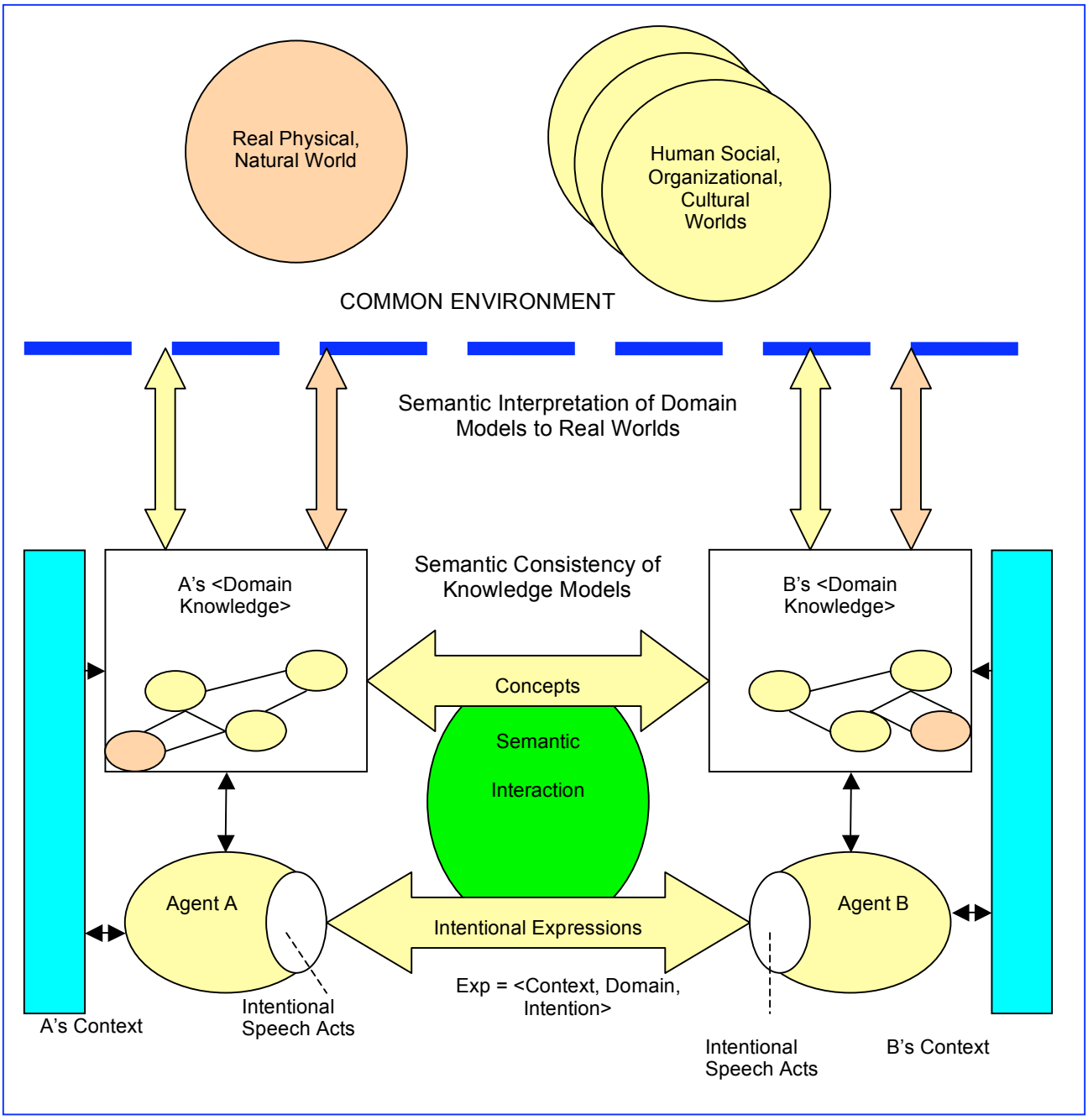


Figure 3-10. Semantic Interoperability Conceptual Framework

The model incorporates concepts derived from cognitive science, knowledge engineering, logic, multi-agent systems, pragmatics, set theory, and the recent World Wide Web Consortium (W3C) efforts to standardize on languages that can be used to add semantics to the WWW. It is the premise of the SICF framework that semantic interoperability always involves an agent's use of definitions and representations of intention, context and domain knowledge when interacting with other agents, and that these semantic definitions are always created by humans, though their final form may be either explicitly transparent, or implicitly discoverable.

This evaluative dimension uses the overall SICF model to characterize the knowledge about the overall Semantic Interoperability situation at a high level. It ensures that the SICF elements referenced in

(Figure 3-10) are identified prior to detailed Semantic Interoperability evaluations. It thus provides an evaluation of the necessary knowledge about overall semantic situation according to the SICF model, as to its completeness, uncertainty, and level of detail. It should be accomplished first in the data gathering stage to enable more detailed subsequent semantic interaction evaluations.

SICF Dimension—Concepts

Fundamental to SICF are the following concepts:

Agents are the peer elements involved in a semantic interaction, and are classified as either human or technology types. The boundaries of agent definitions should incorporate its relationships to domain knowledge, context, and intention definitions and forms of representation. The role of the agent in an interaction may also be prescribed according to communities of practice or more general social constraints, etc. The key is to identify the agent's capabilities and role in the interactions.

All semantics are defined by a human capability to represent and understand knowledge about the world that is expressed in a language that other humans can understand.

Each human agent develops his/her own knowledge about the physical and social world that is constantly evolving as a result of her/his interactions with the real physical world and the social world of other humans

Technology agents have their semantics derived from human engineering processes, and the final semantic representations in current systems may have a form that goes through multiple translations into a technical form that is not easily traceable back to the original human semantic definitions. They may have their semantic definitions explicitly available as integral elements of their implementation instances, or implicitly traceable through their engineering information artifacts used to develop the technology agent, e.g., requirements, architecture, design models, etc. Recent trends indicate the need to represent semantic knowledge in a form that is understood easily by humans and is computable. This semantic definitional approach embodies the human level of explicit semantic representation in the technology result, rather than having a transitive set of translations from human to technology implementation.

Knowledge can be partitioned into domains where the concepts of that domain are represented with their own models and vocabulary

Context provides an overall agent's situational perspective on the use and relevancy of domain knowledge for an intentional purpose, e.g., it may include knowledge about the goals of the agent, role of the agent, the capabilities of the agent, the key aspects of the current situation, the weather, etc. Context also determines the level of detail necessary for the purpose, defines an agent's intention with respect to use of the domain knowledge, and defines knowledge of an agent's physical and social current situation that influences its interpretation of domain knowledge. Shared context between agents can facilitate the joint interpretation of shared knowledge that can create a mutually consistent and integrated picture of real-world objects and social concepts.

Semantic interoperability between agents can be characterized by a definition of a semantic interaction comprised of three basic elements: <context, domain knowledge, intention>. An analysis of the assumptions about these three semantic elements for each interacting agent or entity will characterize the nature, scope, and success or failure of the semantic interaction.

Technology agents can be further classified as cognitive or reactive. Each type has different inherent capabilities to process these three semantic elements, based on whether the representation and interpretation of these elements is explicitly visible in the agent (cognitive), or whether they are implicit in the implementation of the agent (reactive) where the semantic definitions are not easily determined from the implementation .

3.2.5.1.1 SICF Subdimensions

The SICF subdimensions of Figure 3-10 are as follows:

Common Environment Knowledge—Physical: Comprised of the knowledge about the real physical world. The physical environment in which the interacting agents are situated and have knowledge of.

Common Environment Knowledge—Social: Comprised of the knowledge about the social world of humans. The social environment in which the interacting agents are situated and have knowledge of.

Agent Classification: These are classified as human or technology types, which interact with each other for intentional purposes of collaboration. Interactions can be defined for different combination of these types. Each agent interaction involves the sharing of knowledge about context, domain knowledge and intent, e.g., request and commit to actions, etc. The success of their interactions is premised on their mutually consistent understanding (semantics) of knowledge necessary for their intentional purposes within a specified context.

Agent Context: Context defines an environmental situation and agent perspective state, which unifies required knowledge about common environment and specific domain knowledge necessary for a purpose and the set of intentions that that can be used to achieve that purpose. Context can define the role of an agent, its capabilities, the necessary subset of domain knowledge to decide what actions to take or to make appropriate inferences, and any social or other real environmental constraints on the logical inferences that can be made. It may also include such environmental knowledge as agent location, time, movement vector, etc. Other agent context definitions may also determine the perspective, granularity, and relevancy of domain knowledge with respect to the specific context, e.g., a social context, an organizational context, an environment context, a physical context, a task context, a role context, a space-time situational context, etc.

Agent Domain Knowledge: Each agent or entity has an explicit or implicit subset of knowledge about the real physical and social world, as well as specific knowledge related to a domain of interest related to the purpose and intention of the agent, herein called domain knowledge. Successful multi-agent communication is heavily dependent on a mutually consistent semantic interpretation of the exchanged expressions, e.g., the concepts represented by an expression must have some compatibility with interacting agents' knowledge, e.g., context, domain, and intention. Domain knowledge defines the concepts representing a specific subset of world knowledge that an agent understands and can share through semantic communicative interactions. The domain knowledge shared within the interaction has additional semantic characteristics such as language, concept model and taxonomy, real world referent interpretation assumptions, properties, grammar and a logic constraining semantically correct inferences. Examples of domain knowledge models for emergency response context could include concepts and models for medical supplies, first responder capabilities, current medical emergency type, locations of medical treatment centers, geographical area of medical impact, etc.

Agent Intentions—Speech Acts: The agent's intention is expressed through the set of speech acts it uses and by its nature defines one type of context that provides an indication of how to semantically interpret the expression of the interaction, e.g., to share knowledge about the environment, to request a service, to commit to a collaborative action, to signal completion of an action, to state a belief, etc. Each communicative semantic interaction (speech act[6]) has an intention definition, which specifies locutory, illocutory, and perlocutory components. The locutory component is the actual material form of the communicative speech act, e.g., sound waves, radio waves, text messages, symbols, etc. The illocutory component identifies the type of illocutory force the speaker applies to the content or proposition of the speech expression or locutory component. The perlocutory component identifies the effects of the illocutory act on the state of the recipient, e.g., convincing, persuading, etc.

Typically in computer systems types of speech acts[7] have been used in systems where it is necessary to signal the recipient how to interpret the content of the message involved in the interaction, an illocutionary act with intent. "*An **illocutionary act** is any speech act that amounts to stating, questioning, commanding, promising, and so on. It is an act performed in saying something, as contrasted with a locutionary act, the act of saying something, the locution and also contrasted with a perlocutionary act, an act performed by saying something."*

In many systems, protocols have been used for communications, where the messages and their headers are classified into types, where each type has an illocutory force and purpose; in effect an illocutionary speech act. In Wikipedia, illocutionary force is defined as ". . . roughly the speaker's intention behind the production of an illocutionary act, including its communicative point, attitudes involved, and presuppositions."

Thus to enable correct semantic interpretation of an expression in an interaction between agents of whatever type, it is also necessary for the agents to have knowledge of the illocutionary force of the speech act, represented in a message.

3.2.5.1.2 SICF Subdimension Values

Table 3-20 defines the set of assessment values for each SICF Model element, thus providing an overview of the uncertainty or certainty of the knowledge available consistent with the intent of the SICF model framework, e.g., to be able to assess the overall Semantic Interoperability of the networked entities and their semantic interactions and assumptions. No specific formula is provided to define what is adequate at this level of evaluation, but certainly more than a few uncertain values would infer that the consequent detailed dimensional analysis of Semantic Interoperability is somewhat uncertain due to a lack of knowledge. The intent is to encourage the use of the SICF model to gain an understanding of the elements involved in semantic networked interactions, to provide a guide to those requiring knowledge about the elements affecting the level and certainty of the congruence of semantic networked interactions, and to provide an objective bias on the subsequent Semantic Interoperability analysis results.

[6] "Multi-Agent Systems – An Introduction to Distributed Artificial Intelligence," Jacques Ferber, ADDISON-WESLEY, ISBN 0-201-36048-9

[7] http://en.wikipedia.org/wiki/Illocutionary_act

Table 3-20. SICF Dimension—Model Knowledge Assessment Values

SICF Subdimensions	SICF Subdimension Values
Common Physical Environment Knowledge—Physical situation that interactions are constrained by or any assumptions influencing semantics of interactions.	• Known and fixed • Known and variable • Uncertain
Common Social Environment Knowledge—Social situation that interactions are constrained by or any assumptions influencing semantics of interactions. Example would include identification of organizational policies, communities of practice, operations practices, types of missions, training dependencies, etc.	• Known and fixed • Known and variable • Uncertain
Agent Classification—Entities involved in the semantic interactions. Specifically their roles in the semantic interactions and whether they are human or technological.	• Known and fixed • Known and variable • Uncertain
Agent Context—Identification of physical and social and purposeful situations associated with semantic situations.	• Known and fixed • Known and variable • Uncertain
Agent Domain Knowledge—What domain knowledge is required or understood by the agent types to successfully and consistently interpret the meaning and intent of the interactions with respect to their roles. How is the information represented, is it explicit or implicit, are the representations based on standards for interoperability, is there a defined community of practice associated with the domain knowledge, is there general agreement on the purpose of the domain knowledge, do the interactions utilize the concepts from the domain knowledge representation and are these relationship clearly defined.	• Known and fixed • Known and variable • Uncertain
Agent Intentions and Speech Acts in communications supporting Interactions. Are the messages for communications defined using standards, doe the content of the messages relate to domain knowledge, does each message have a clear intention, are the set of message defined for collaborative interactions.	• Known and fixed • Known and variable • Uncertain

3.2.5.2 SICF Semantic Interaction Compatibility – 2.5.2

Semantic Model Assumption: As illustrated in Figure 3-10, all human or computer communications supporting a purposeful type of interaction require semantic definitions relating the referents of the intentional expressions to the concepts in the agent's domain model, where the intention definition further refines the intent and meaning of a signal, a command, a request, a shared data element, a communications response, an expression or proposition describing the world or a situation, etc. The ability for agents to realize mutually consistent understanding of exchanged expressions in a semantic interaction is increasing due to globalization, while a counter force to this is the emergence of specialization by communities defining their own specialized concepts and languages for specialized knowledge. Though specialization provides quick semantic shortcuts for those trained in the specialized domain, it has the effect of decreasing ubiquitous understanding of shared knowledge. Specialization has the effect of trading off global understanding for local optimization of semantic interactions. Invariably, semantic interactions using knowledge from specialized domains will have to rely on the specialists to translate their meaning into a common language that is more ubiquitous for wider understanding.

Semantics Compatibility Assumption for Technology Interactions: All technology elements that interact with each other have an original set of predefined human semantic definitions covering the

intent of the interaction, the meaning of the information exchanged, and the context of the overall interaction. Technology based semantic interactions have an additional problem where the predefined semantic definitions made during the engineering development life cycle processes are in many cases lost. If not lost, the definitions prove complex, in relating the final implementation implied semantics to these original explicit semantic definitions. For example, the engineering artifacts, with their semantic definitions, are not usually inherent in the technology itself, e.g., design models. This makes recovery of the original interaction semantic definitions problematic, and results in possible misinterpretation errors that can cause non-interoperability when engineers attempt to extend the definition of the semantics for an existing technology interaction, or attempt to add new systems that can interact with the original system community.

This problem of technology interaction semantics has been compounded by the proliferation of various types of technology, and their specialized assumptions about the knowledge for the interactions when using this technology. In most cases, engineers must read documentation and other models to be able to develop semantically consistent interactions with this technology since the semantic interaction definitions are not embedded within the technology itself.

Newer concepts such as envisioned by the Semantic Web provide the means for online access to semantic definitions of shared knowledge and service, and in some cases, these semantic definitions are also embedded as part of the final implementation. Different levels of semantic expressiveness may be used, with the highest being expressed as ontologies and exchanged metadata, which enable the original semantic definitions to become part of the technology implementation and thus reduce possible errors of semantic interpretation by engineers when extending the scope of the original interaction set.

3.2.5.2.1 SICF Semantic Interaction Compatibility Subdimensions and Values

Each interaction should entail the following concepts and for each concept a determination can be made regarding level of mutual consistency. The values are organized per Table 3-21.

SICF Semantic Interaction Subdimensions

1. **Semantic Interaction Model Compatibility (<*context, domain knowledge, intention*>)**
2. **Language Compatibility**—all agents use compatible languages in the expressions of exchanges involved in the interaction.
3. **Intentions Compatibility**—compatible and defined **intentions** of the expression in the interaction,
4. **Domain Knowledge Compatibility**—have similar definitions in **domain knowledge** models for meaning with respect to the world or similar human defined concepts in order to reach mutually consistent understanding of the propositions exchange with each other,
5. **Context Compatibility**—compatible joint knowledge of the **context** of the semantic interaction, which will also influence how the domain knowledge referred to by the expression will be interpreted. This is especially true of indexical concepts, where the referent real world object changes due to different contextual interpretations, e.g., the domain concept location may refer to either the city that a person is in or it may refer to a longitude/latitude/height tuple. The necessary accuracy of the location information depends on the context of the use of the information, e.g., a census count might require only zip code accuracy, while an emergency response would require a street address or equivalent.

6. **Collaboration Compatibility** of roles and capabilities—this is especially important in open collaborative networks, where each agent has certain capabilities in roles in accomplishing joint collaborative activities to achieve some overall goal

Mutual consistency definitions concepts within the SICF Interaction model

These values defined here imply that the semantic interpretation of the exchanged expression and its concepts, will find corollaries in each interacting agent's domain model, and that inferences made based on this shared knowledge will also be consistent across the agent domain models.

There could be different levels of mutual consistency:

- *Disjoint domain concepts*—participants in the interaction are operating with different elements of knowledge in potentially different domains.
- *Overlap domain concepts*—where the recipient and originator concepts overlap in meaning but are not exactly equivalent, or some concepts are unique to each participant and some are common.
- *Subset domain concept*—where either the recipient or originator domain concept is a subset of the other's set of concepts, $C1 \subset C2$.
- *Equivalent* domain concept—where the originator and recipient concepts are equal $C1 \equiv C2$.
- *Model inconsistent*—concepts are compatible but the logical inferences between concepts are not consistent, or the meaning of the concepts are incompatible, or the concepts imply different sets of truths.
- *Model consistent*—concepts are compatible and the inferences of other concepts in the model are consistent between originator and recipient.

We can now join these into Table 3-21 to make assessments about the potential for semantic interaction consistency.

Table 3-21. Semantic Interaction Dimension Compatibility Values

Semantic Interaction Subdimension	Subdimension Value	
Compatible Semantic Interaction Model <*context, domain knowledge, intention*>	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent
Mutually compatible languages used in the interaction message and data	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent
Mutually compatible intentions of the interaction	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent
Mutually compatible domain knowledge models	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent

Mutually compatible context of the interaction	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent
Compatible collaboration roles, similar purpose for exchanging messages and interactions	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent

3.2.5.3 Human Semantic Dependencies Subdimensions and Values – 2.5.3

In addition this characterization of the semantic layers involved in the interaction, the following dimension values identify the semantic dependencies among different human agent contexts, e.g., their role. The potential agent context roles are listed in the following in addition to a description of the context (Table 3-22). Do the collaborating agent types match, do they have similar domains and context? The subdimensions that should be compared to establish possible semantic dependencies between human collaborators are the following:

- Individual role.
- Social or other cultural assumptions about background knowledge.
- Organizational mission/focus.
- Language used for communication and sharing of information.
- Domain knowledge.
- Context of the situation among collaborating agents.

Table 3-22. Human Semantic Dependencies and Relationships to Network Perspective

Human Semantic Dependency Subdimensions	Human Semantic Dependency Subdimensions Values	
Individual role dependency	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent
Social or cultural background knowledge dependency	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent
Operational mission/focus dependency	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent
Language dependency	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent
Domain knowledge dependency	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent

Situational context dependency	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent
Any value actually is allowed at a subdimension layer, but this could result in lack of semantic understanding between different agent types due to background knowledge differences in their respective domains.		

3.2.5.4 Operational Context Flexibility Semantic Compatibility – 2.5.4

This dimension identifies two important semantic interoperability subdimensions from an operational perspective within a context model. All other context semantic characterizations described in the Semantic Interoperability dimension could also consider these context subdimensions, e.g., Time Context and Situational Context, as part of their SCOPE evaluations.

These context subdimensions provide guidance on how to interpret domain knowledge from the perspective of a context. Typically, one takes an agent's perspective in an operational sense with regard to time and the overall situation of the agent, as well as the overall environment situation across agents and environment objects.

Time context provides guidance with respect to interpreting and selecting instances of information that are associated with facts in a knowledge base. It is well known that facts that satisfy a semantic expression in a domain model depend on the contextual time with which that fact is associated. A location fact in a domain knowledge base will also be associated with a time context to enable queries and reasoning about facts satisfying certain locations of objects or agents at various times. Time can also be associated with collaborative states to enable mission tracking and process management. Both time and situational knowledge provide organizing relationships across facts in a knowledge base, thus enabling reasoning across time interval and at points in time.

Time context has three potential purposes:

1. To enable facts in a knowledge base to be structure around time instants.
2. To associate an agent's perspective with a particular time instant.
3. To associate an overall situation state with a time fact.

Situational context is typically associated with such concepts as physical location, multi-agent collaborative states and agent roles, current agent devices and capabilities, current communications availability and capabilities, etc. Situational context is an umbrella concept that could contain many different subconcepts defining a situation in time and space. The concepts of relevance are related to the situation definition in use from an operations perspective.

The approach used is to characterize context interoperability compatibility, which entails determining the compatibility with respect to semantic representation, granularity, range, and international standards. For example, the Time Context should be evaluated from the assumptions about the time relationships for shared domain knowledge base facts, intentional collaborative interactions, and for relationship to the Situational Context.

Each of the following Operational Context subdimensions will be evaluated for compatibility.

Context Values

Each context dimension should be evaluated with respect to the following areas for compatibility:

- Representation of context (explicit or implicit)
- Level of semantic representation (structured, unstructured)
- Granularity relationship (equivalent, finer)
- Range (equivalent, covers, disjoint)
- International standard (none, exists)

Operational Context Subdimensions

- Time context
- Object situational context
  - Geospatial location context
  - Domain classification context
- Multi-agent context (shared situational context knowledge)
  - Collaborative relationship
  - Complementary roles
  - Agent capability representation
  - Agreement/commitment protocol
  - Environment domain knowledge

3.2.5.4.1 Operational Context Dimension Values

Table 3-23 is an example of the table to be filled out by the evaluator.

Operational Context Subdimensions	Dimension Values			
	Representation of Context (Explicit/Implicit)	Level of Semantic Representation (Structured, Unstructured)	Granularity Relationship (Equivalent, Finer)	International Standard (None, Exists)
Time Context				
Object Situational Context				
Geospatial Location Context				
Domain Classification Context				
Multi-agent Context (Shared Situational Context Knowledge)				
Collaborative Relationship				
Complementary Roles				
Agent Capability Representation				
Agreement/Commitment Protocol				
Environment Domain Knowledge				

3.2.5.5 Global Versus Specialization of Domain Knowledge for Communities of Interest – 2.5.5

This dimension is intended to capture the degree to which a given system or the COIs it supports or is sponsored by represents knowledge that is widely known/understood or uses vocabulary and language that has very specialized meanings known only to that community. Conceptually, this is similar to organizational openness, but is not generally a deliberate result of the community attempting to isolate itself. Usually this is simply due to the specialized nature of the work that the community concerns itself with, and the vocabulary and semantics it has developed, often as a profession, to facilitate research, development, and operations in a specific field or specialty. Of course, this directly affects the discoverability and understandability and usability of information and services provided by that community to other systems and users on the network. A high degree of such specialization typically suggest that some information broker services, possibly operated by some COI-level organization, might be needed to make the information/services it offers more generally understandable and usable by other network users.

The SCOPE WG has developed an initial model for measuring this dimension, but there is as yet little pragmatic experience with utilizing this dimension in net-centric assessments of actual systems or capabilities. In addition, this model was developed before the advent of the NCOIC SIF WG under the Specialized Frameworks Functional Team. In retrospect, some of the material included in the model might better fit into a description of semantic interoperability patterns. As a consequence, the initial description of this dimension is included under Section 4, Emerging Dimensions, as Section 4.2.3. The SCOPE WG intends to evolve this description in future versions of the SCOPE model, in cooperation with the SIF WG, and welcomes contributions to improving the definition of this dimension to facilitate its use in net-centric assessments.

3.2.6 Organizational Business Model and Culture – 2.6

This dimension is aimed at capturing the degree of openness and cultural diversity in the organizations that are attempting to implement a network-centric capability with each other. Their willingness to interact and share information will drive the degree of network centricity that they specify/allow in the systems and services they sponsor. A key aspect of this willingness is the organizational policies they enforce regarding information sharing and the risk they are willing to assume regarding the levels of protection (i.e., security mechanisms) required to share information and services with other organizations and their systems over a network. The dimension also measures the degree to which organizations backup their stated policies with commensurate resource allocations and incentive structures for open, net-centric behaviors. Note that this is not a judgmental factor by itself. For example, the mission of an organization may have high inherent operational risks if it shares information with others, and such an organization should not be expected to share information freely.

The SCOPE WG has developed an initial model for measuring this dimension, but there is as yet little pragmatic experience with utilizing this dimension in net-centric assessments of actual systems or capabilities. As a consequence, the initial description of this dimension is included under Section 4, Emerging Dimensions, as Section 4.1.2. The SCOPE WG intends to evolve this description in future versions of the SCOPE model, especially in cooperation with the SIF WG and the operationally oriented NCOIC IPTs, and welcomes contributions to improving the definition of this dimension to facilitate its use in net-centric assessments.

3.2.7 Life Cycle Control – 2.7

The Life-Cycle Control dimension is intended to capture two primary attributes of a network-centric set of systems and services. These are the degree to which a single executive has budgetary and operational control over the systems and services involved and the degree to which the systems' are aligned in their development and sustainment life cycle. In other words, are the systems/services all being built by the same sponsor and implementing organization, and are they being built on a synchronized schedule and expected to have identical life spans. In most network-centric capabilities, the systems comprising the service set needed for the capability typically have some diversity in ownership/sponsorship organizations with different objectives, and have different development and sustainment timelines. This creates decisional dynamics that impact which systems adapt to and take advantage of which other systems in the network-centric ecosystem to achieve some capability. For example, existing systems and systems nearing their expected end-of-life are less likely to be modified to accommodate some new system, even if that might otherwise be the most cost-effective implementation approach for a capability in the near term.

The SCOPE WG has developed an initial model for measuring this dimension, but there is as yet little pragmatic experience with utilizing this dimension in net-centric assessments of actual systems or capabilities. As a consequence, the initial description of this dimension is included under Section 4, Emerging Dimensions, as Section 4.2.2. The SCOPE WG intends to evolve this description in future versions of the SCOPE model, and welcomes contributions to improving the definition of this dimension to facilitate its use in net-centric assessments.

3.3 Capability/Domain-Specific Scope Dimensions – 3.0

While the Capability/Domain-Independent Scope dimensions are applicable to just about any capability or functional domain, the Capability/Domain-Specific Scope dimensions are capability/domain-specific. They are thus sensitive to the functional domain being considered, and it is not practical to attempt to define a comprehensive set of dimensions in the model *a priori*. For any given assessment, an appropriate set of dimensions will need to be defined, although over time a substantial body of reusable material may be developed to facilitate this process.

Figure 3-11 illustrates a representative sample of Capability/Domain-Specific Scope dimensions. The Capability/Domain-Specific Scope dimensions are the responsibility of the operational architecture proponents, with heavy emphasis on measures of effectiveness (MOEs) appropriate to a capability. It is reasonable to expect that the FCBs and the JCIDS process will drive the selection of the Capability/Domain-Specific Scope dimensions and values associated with a given set of capabilities.

Value / Example Dimensions	Less Capability			More Capability
Time to Target Engagement	1 Hour	30 Minutes	10 Minutes	1 Minute
Stryker Bde Deploy Time	30 Days	7 Days	72 Hours	24 Hours
Total Lift Capacity	Single aircraft type	Multiple aircraft types	Multiple lift types	All lift types
Target Detection	Single sensor	Multiple sensor	Multiple sensor types	All source
ISR Management	Single Platform	Multiple Platforms	Multiple platform types	All platform types
Power Grid Denial	Single Substation	Single Plant	30% Power Capacity	100% Power Capacity

Figure 3-11. Example Capability/Domain-Specific Dimensions

3.4 Technical and Economic Feasibility Dimensions – 4.0

The Technical and Economic Feasibility dimensions assess the degree to which the interface services between systems/capabilities are technically and/or economically feasible given the technical architecture constraints and the resources available to affect the interface services.

In general, the dimensions below should be interpreted to indicate that technical/economic feasibility is increasingly more challenging as one moves from "left" to "right" across the dimensions (or "down" to "up" or whatever the appropriate metaphor is for each dimension). This corresponds to the increasing impact on program or capability implementation cost and risk as one moves across each dimension.

3.4.1 Inter-Element Time-Binding Sensitivity – 4.1

A key technical feasibility measure for network-centric capabilities is the degree of time binding needed between systems to execute a capability. In this context, time binding is measured in terms of the amount of time required for a service requestor and provider to bind to each other and interact to provide an operational capability. The tighter the time binding required to effectively support a capability, the greater the demand on the technical architecture elements used to implement the interface service. Often the technical architecture elements make it infeasible to meet the time binding needed to support a capability such as time-critical targeting or ballistic missile defense. For example, the speed of light latency and fairly large spatial distribution of network nodes may make some operational capabilities infeasible to implement using a network-centric approach.

Value Type

Named ranges (hard real time, near real time, transactional, tactical, operational, accounting, strategic). These can also be converted to numbers for scoring purposes.

Value Set

The ranges included in this version of the document represent a somewhat logarithmic scale and reflect the kind of inter-system time binding criteria found in existing classes/types of systems in operational use today. The boundaries between the values may overlap, so in conducting an assessment, some judgment about the importance of the time binding aspect of a given interface to operational success will need to be applied. It should be emphasized that either end of this scale suggests that a network centric solution might not be appropriate. At the hard real-time end of the scale, it may not be technically and economically feasible to achieve the low inter-element network latencies needed. At the other end of the scale, there is essentially no operational benefit to being network centric, because transferring the information between systems can take too long with little impact on the usefulness of the resulting product/capability.

The scale presented in Table 3-24 includes seven levels or ranges, but additional ranges could easily be added. The values can be converted to numbers using a simple 1-7 range, but in inverse order so that the more net-centric value is the higher value. Thus, strategic time binding requirements, which are measured in months and years, would result in a score of 1, while hard-real-time binding would merit a 7. A different mapping of values to numbers could be used with more weighting on the hard-real-time end of the spectrum if one wants to emphasize a more "real-time" enterprise/capability as the goal.

Table 3-24. Inter-Element Time-Binding Sensitivity Value Set

Value	Description
Strategic	Months to years, e.g., deliberate planning functions, annual budget reviews.
Accounting	Daily/weekly/monthly/end-of-year processing, typical of financial reporting systems.
Operational	Hours/days, required for inventory management, force element employment planning/replanning, monitoring.
Tactical	Minutes to hours, required for planning and execution monitoring of tactical operations, targeting, vehicle routing/scheduling, and similar capabilities.
Transactional	Subsecond to several seconds. This is typical of credit card transactions, target engagement decisions, search/queries, and some application user interface response times.
Near Real Time	Milliseconds to subseconds. This is typical of vehicle/weapon control systems and some aspects of user interfaces (e.g., mouse responsiveness).
Hard Real Time	Millisecond and below response times. This is typical of high-speed weapon systems control loops such as in missiles guidance and homing systems and the like.

Example

Examples of each value set element are given in the table with each value description. For any given capability, the nature of the capability often dictates the time-binding range among systems that would lead to operationally effective performance, resulting in a single value for this scale for all the systems involved. However, there may also be cases where some systems will require different levels of time binding for different aspects of the capability. For example, target selection or weapon/target pairing may operate at the tactical or transactional level, while weapon engagement with the target via sensors might operate at the near-real-time or hard-real-time level. Therefore, a system might want to behave more net-centrically in the target selection portion of mission space, and more tightly coupled in the weapon/target engagement portion of mission space. Of course, the actual data collected by the engagement sensor might still be made discoverable and visible on the network (e.g., for postoperation analysis), but that doesn't have to be done in real time as part of the target engagement process.

Applying this dimension of SCOPE on some capability requires some sensitivity to this general issue of what time binding levels apply to what part of the overall capability domain space. If there is more

than one time-binding value that is appropriate to a given capability, the capability/system should be decomposed into elements and the time binding appropriate for each element interaction or element interaction type should be captured. This avoids making blanket response time decisions that could increase the cost of a capability needlessly, or force a network centric approach on a portion of the capability where hard real-time interactions are called for.

3.4.2 Transport Capacity Needed – 4.2

The dimension set described in the following, addressing the transport capacity needed to effect desired capabilities, may grow in time to encompass a range of network performance issues. Key among these is network latency.

3.4.2.1 Network Latency – 4.2.1

An important technical feasibility measure for network-centric capabilities is the network latency that can be measured in a network round trip between given (or representative) source and destination nodes. This is the total time it takes for a packet to traverse the network between these nodes. Round-trip latency excludes the amount of time that the destination system spends processing the packet. Thus, the "ping" utility/service commonly found on most computing platforms (which does no significant packet processing) is a rough, but handy means of measuring network latency.

To be more precise, one-way latency can be defined as the time from the start of packet transmission to the start of packet reception. This definition is independent of the link's throughput and the size of the packet, and is the minimum delay possible with that link. This is a simplistic view, however, because in a typical network environment, a packet is forwarded over many links through many gateways, and each of these does not start to forward the packet until it has been completely received. In such networks, the minimum latency is the sum of the minimum latency of each link, plus the transmission delay of each link except the final one, plus the forwarding latency of each gateway.

Network latency can be a critical factor for net-centric systems and applications that require rapid and consistent responsiveness from network services. Thus, the Network Latency dimension and the Inter-Element Time-Binding Sensitivity dimension are closely (and inversely) related. The greater the time binding sensitivity is (i.e., the shorter the binding time must be), the smaller the network latency must be to satisfy the time binding requirements.

As noted, network latency consistency can be as important as the magnitude of the latency in many circumstances. This dimension does not directly address that at this time, but it is a factor that should be considered.

Value Type

Named numeric ranges: negligible, minimal, significant, considerable, large, prohibitive

Value Set

The ranges offered in the value set defined in Table 3-25 are somewhat arbitrary and can be tailored by SCOPE users to address their domains, systems, and applications of interest appropriately. For example, in one domain, a latency of 100 ms might be prohibitive, while in another, such latency might be negligible. If greater precision is needed, another way that the value set can be tailored is to define additional, more fine-grained ranges.

If desired, the values can be converted to any desired integer scale for scoring purposes. The obvious such conversion would be to map them to the numbers 1-6, but if we consider the principle that less network latency facilitates more net-centricity, values should be mapped to the integer scale in inverse order so that the more net-centric value is the higher value.

Table 3-25. Network Latency Value Set

Value	Description
Negligible	Less than 10 ms—Typical LAN or very fast WAN response.
Minimal	10-60 ms—Typical fast Internet response.
Significant	60-300 ms—Typical moderate Internet response.
Considerable	300-1000 ms—Slow or delayed Internet response.
Large	1-5 sec—Problematic internet response, indicating major congestion, rerouting, outages.
Prohibitive	Greater than 5 sec—Major delays, possibly indicating serious network problems.

Example

Network latency can be a key consideration in determining which elements of a system to make more distributed, loosely coupled, and network aware. If the latency is low, and the time-binding sensitivity of system elements is generally not severe, many of those elements can be distributed across the network in a loosely coupled, net-centric manner. If the latency is high, this will tend to restrict the set of system elements that can be net-centric, because they will need to be closer on the network to the elements they communicate with and will tolerate less network traffic to facilitate net-centric qualities such as network visibility, discoverability, and manageability.

Different elements of a system are likely to have different sensitivity to network latency. For example, weapon communication with sensors during target engagement will generally be highly sensitive, whereas weapons status communication will be less so. Thus, it is good practice to evaluate network latency impacts on various system elements individually (keeping system interdependencies in mind, of course) to determine which of them may be candidates for more net-centric implementations.

3.4.3 Run-Time Computing Resources Needed – 4.3

The use of run-time computing resources is a key technical feasibility measure of net-centricity. It addresses the amount of resources needed by applications or services versus what is assumed to be available. Run-time computing resources include all the resources an application or service needs to operate. They include processor resources, storage resources, and network resources. Network resources are involved when an application accesses resources through a network. Network resources may include processors, storage, and interaction with other applications. Performance affects are the direct result of resource management or allocations and if characterization metrics are defined in terms of QoS-type parameters, then they can provide both a target and measurement of performance achieved. The technical feasibility of co-opting extensive run-time resources depends to a large extent

on where the resources are required. It is greatest on the host machine and less likely on networked resources unless special provisions are put in place ahead of time.

3.4.3.1 Storage Utilization – 4.3.1

The storage utilization dimension is a measure of the degree to which interactions among systems over the network are constrained or facilitated by available information storage capacities of host nodes on the network. Network data exchange volume per unit time and data retention requirements are the primary drivers of storage requirements on host nodes. These factors are in turn driven by the nature and volume of the operational information that must be exchanged among systems over the network and the enveloping or representational overhead added by the network and application protocols used to transfer the information. The retention period is also driven potentially by regulatory and enterprise policies regarding audit trails and record retention.

Storage capacity of host nodes on the network can constrain the types, frequencies, and volumes of information exchanges over the network, as we often experience in our everyday lives when we get notified that our email inboxes are "full," and we can't send anymore email until we clean out our inboxes, or may even have older inbox messages deleted by "the system." As in the case of the processor utilization dimension, the greater the percentage of the available storage capacity that is devoted to supporting and recording interactions with other systems on the network, the more net-centric the systems are. The storage utilization dimension is therefore best characterized as the percentage of overall host network node storage capacity set aside for supporting interactions with other systems over the network.

The type of storage available and used (e.g., magnetic disk, DVD, tape, RAM, flash, etc.) may also be important at a more detailed level. For example, web service applications may employ persistent storage on the server machine, but are not usually allowed to create or access persistent storage on the user's client machine without special preparation. The use of random access memory is usually allowed, but the creation of temporary files during execution may depend on the operating system and organization security policy. However, this aspect is left as a possible future addition of new subdimensions to the SCOPE model.

Value Type

Percentage range of available storage capacity devoted to storage of information related to exchanges of information with other systems or users over the network.

Value Set

- Negligible storage utilization (less than 1% of available host node capacity).
- Minor storage utilization (1-10% of available host node capacity).
- Significant storage utilization (10-30% of available host node capacity).
- Major storage utilization (30-100% of available host node capacity).
- Capacity constrained utilization (more than 100% of available host node storage capacity).

Example

An asset management system may have a large percentage of its storage capacity devoted to storing descriptive information about a large number of assets owned by an organization. The nature of the

assets is such that this information is rarely changed and only infrequently interrogated by other systems. Such interrogations might be very simple and return only a small percentage of the information retained in the asset descriptions. In such cases, storage utilization values would likely be negligible or minor. By contrast, an "in-transit visibility" system is likely to have relatively little information about the assets themselves, but be updated frequently by sensor or other data-input systems regarding assets being inserted into (i.e., sent) or removed from (i.e., delivered) some transportation flow, as well as intermediate movement updates. Such systems would likely be characterized as having major storage utilization for network traffic, since the asset data itself is a small fraction of the overall transactional data flow volume over the network.

Changes to reporting frequency, number of sources on the network, or level of detail in status reporting are likely to have a big impact on storage capacity in the latter case, but not in the former. Systems of the latter type are more likely to reach capacity-constrained utilization levels, and thereby can constrain the operational capabilities of the organizations with respect to what might otherwise be achievable through net-centric interaction.

3.4.3.2 Application Interaction Frequency/Pattern – 4.3.2

For an application the feasibility of NCO may depend on the application interaction frequency and/or pattern and may have a direct impact on bandwidth requirements and, hence, QoS provided to this and other applications. This depends to a great extent on the amount and type of data interaction required. Possible types of interactions include low latency, low frequency, high frequency, low data, and high data in various combinations.

Value Type

Frequency/pattern

Value Set

- Low frequency and low data requirements.
- High frequency, but low data requirements.
- Low frequency, but high data requirements.
- High frequency and high data requirements.

Example

Static or text-only web pages usually have low-frequency and low-data requirements, while data exchange services may have both high-frequency and high-data requirements. Taken alone, the loading of these types of applications are predictable, but in the aggregate they may produce a network loading pattern that is cyclic in nature. In addition, some applications also may have a loading pattern that is cyclic in nature. The network supporting application interactions must be sized to support these cycles or delays and interruptions will occur.

3.4.3.3 Processor Utilization – 4.3.3

This dimension is intended to measure the overall processing capacity of systems devoted to interacting with other systems over the network. Typically, this might best be measured as a percentage of network host node processing capacity instead of some absolute processing capacity

measure (e.g., Specmarks or MIPS). However, even a relative measure requires some agreement as to how to measure both processor capacity and the processor utilization entailed by some specific set of network interactions. In addition, processor utilization depends on both processor capacity, which thanks to Moore's Law, continues to grow dramatically over time, and on a subjective assessment of what constitutes acceptable performance over the network for a given operational activity. Arguably, systems that consume a majority of their processor resources on interactions with other systems over the network are more net-centric than those that do not. Conversely, available processor capacity will constrain interactions over the network if those interactions require sizable processing capacities.

The SCOPE WG has developed an initial model for measuring this dimension, but there is as yet little pragmatic experience with utilizing this dimension in net-centric assessments of actual systems or capabilities. As a consequence, the initial description of this dimension is included under Section 4, Emerging Dimensions, as Section 4.3.1.1. The SCOPE WG intends to evolve this description in future versions of the SCOPE model and welcomes contributions to improving the definition of this dimension to facilitate its use in net-centric assessments.

While processor utilization might be viewed purely as a CPU capacity measure, the intent here is to have it include other capacity and resource attributes of nodes providing services to others over the network. This would include such nodal processor capacity attributes such as available main memory resources, disk storage capacity, and disk access time and data transfer rate characteristics. At a more fine-grained level it may be desirable to make this explicit by creating separate subdimensions for each of these possible resource constraint types at some future date.

3.4.3.4 Nodal Quality of Service – 4.3.4

Like Processor Utilization, Nodal Quality of Service is another major driver of a network node's ability to interact with service providers and consumers over the network. This dimension is intended to measure the degree to which relevant network nodes' ability to deliver and consume services over the network are constrained by their immediate network media connections. This is not normally an issue with so-called "back-office" services, which typically have local network access connections that are faster than the "long haul" or global network capacity. However, in a network-centric environment, information that is often most critically needed to conduct operations is first made available over the network by service provider nodes that are not well-connected to the long haul network. Essentially, this is a version of the "last-mile" network bandwidth problem often raised for homeowner access to the Internet, only in reverse. The requestor of the data service may be well-connected to the network in such cases, but the data service provider is not. In contrast to transport QoS considerations over the long haul, common user network, the local service provider may have specific knowledge of the operational situation being supported and the needs of different service requestors, and make QoS decisions based on this information rather than on the basis of general network traffic types and service categories.

The SCOPE WG has developed an initial model for measuring this dimension, but there is as yet little pragmatic experience with utilizing this dimension in net-centric assessments of actual systems or capabilities. As a consequence, the initial description of this dimension is included under Section 4, Emerging Dimensions, as Section 4.3.1.2. The SCOPE WG intends to evolve this description in future versions of the SCOPE model and welcomes contributions to improving the definition of this dimension to facilitate its use in net-centric assessments.

3.4.4 Enterprise Service Management Feasibility Dimension – 4.4

The capability to manage network services over one or more enterprises is an important aspect of net-centricity. Technical feasibility diminishes as the need for enterprise service management increases. Enterprise service management encompasses dynamic service composition, dynamic resource configuration, information assurance considerations, QoS, and network management. Network management and dynamic resource configuration employ the same methodologies, the difference being one of degree. Dynamic resource configuration goes beyond network management in that network management usually limits itself to standard infrastructure elements while dynamic resource composition extends to servers, processors and other configurable elements.

3.4.4.1 Dynamic Service Composition Feasibility – 4.4.1

Dynamic service composition involves interfacing with one or more network services to accomplish a task. The existence of fairly stable network services is a requirement for dynamic service composition. For the purpose of this discussion, the discovery of a service or services is assumed, and this aspect is covered elsewhere. Dynamic service composition spans the range from interfacing with a single service to synchronizing the outputs and inputs of a number of services, each of which may be used to accomplish part of a task purpose. Dynamic service composition may include stateful or non-stateful services. Stateful services are those services that provide indications of state. Stateful services are used when a multistep service is required.

Value Type

Types of interfaces

Value Set

- Interface with one service.
- Interface with multiple services.
- Interface with stateful services.
- Synchronization of the input/output of multiple services.

Example

The use of dynamic service composition depends on the stability and responsiveness of network services. For example, if one service provider creates maps on demand and accepts feature data to display on the maps, and another service provider creates feature data for maps based on periodic inputs from users, then by using the two services, a third service can be provided that includes maps with special feature data. The resultant service may also provide indications of the currency (state) of the resultant maps in which case it is a stateful service.

3.4.4.2 Dynamic Resource Configuration and Network Management Feasibility – 4.4.2

Both dynamic resource configuration and network management involve configuring one or more resources to accomplish a purpose. Resources may be local to a single machine or distributed on a network. Resources may include application resources, e.g., processors, servers, etc., configured by a network or the network infrastructure resources themselves, e.g., routers, switches, etc. Resources may be configured either out-of-band, e.g., RS232, or in-band using IP, e.g., Simple Network Management

Protocol (SNMP). Dynamic resource configuration requires that the operating systems on the devices to be configured to accept configuration commands.

Network management needs characterize interoperability-relevant aspects between systems and/or services on a network in terms of the extent of network management required for NCO. Networks can be classified as local intranet, enterprise, public, and *ad hoc*. At its fullest extent network management includes the functions of design, configuration, monitoring, and maintenance. At the local intranet level, it is assumed that all functions are required. At the network level special configurations may be necessary to provide the required QoS. Special configurations may require periodic configuration monitoring and adjustment. At the ad hoc network level autonomic network management may be required due to the lack of a centralized monitor and control system.

Value Type

Type of resource

Value Set

- Configuration/use of a local resource.
- Configuration/use of multiple resources on intranet.
- Configuration/use of multiple resources on enterprise.
- Configuration/use on a heterogeneous network.

Example

Dynamic resource configuration and network management both involve sending commands to network infrastructure elements. These elements may include routers, switches, and other nodal resources. The ability to accomplish this will dictate the degree of net-centricity of applications or services that require this capability. If routing protocols must be put in place to allow all routers on a network to handle specific classes of traffic in a specific manner, configuration commands are sent to the routers through IP. The configuration of intelligent devices is usually accomplished by SNMP, but the MIB may first need to be set using SNMP, and the agent that monitors the MIB may also need to be modified.

3.4.4.3 Information Assurance Feasibility – 4.4.3

Information assurance resources include elements needed for an application to communicate securely. The basic requirement is implementation of user access control to a single machine or networked machines. Access control includes authentication and authorization. Once a user application gains access to the network, it must also gain access to the resources required, e.g., files, databases, etc. Access to resources may also include methods for secure interaction and accounting for access to and use of any resources. Information assurance resources needed may be delineated as follows.

Value Type

Degree of access

Value Set

- Implementation of access control including authentication and authorization for services.

- Provision of special permissions for resource use.
- Secure interaction.
- Security accounting for access and resources used.

Example

Net-centricity may involve interfacing with and information assurance resources. Before a user can access a machine, the operating system permissions must be set to allow user access. Before a user can access a dedicated resource, e.g., server, database, etc., special permissions may also be required. Examples are operating system permissions, passwords, encryption/decryption devices, etc.

3.4.4.4 Quality-of-Service Feasibility – 4.4.4

QoS feasibility may be a determining factor for net-centricity. This dimension is concerned both with the resource requirements for QoS and the feasibility of meeting those requirements. It is intended to measure the degree to which there exists adequate network resources (mainly, but not exclusively, bandwidth) to support the network overhead required to implement QoS monitoring and management services, since these consume bandwidth over and above the organizational mission services that the network supports. If network resources are relatively constrained, organizations may wish to minimize the use of QoS mechanisms in exchange for greater mission throughput, but at the risk of having high priority applications encounter sporadic congestion induced delays. Conversely, organizations may sacrifice some overall network capacity in exchange for higher probability that critical applications will get the network resources they need.

As overall network capacity increases with respect to aggregate demand, it generally becomes more feasible to devote increasing network resource levels to QoS management services. Aggregate network throughput becomes less of a concern as available bandwidth increases. Organizations are then generally more willing to devote network resources to minimize the chances that high priority services are impacted by network congestion caused by random fluctuations in network loading or by specific network resource failures.

While QoS service management is usually easier to implement on an intranet than on the open Internet (or other shared, federated, network-centric infrastructure), it still depends on the availability of adequate resources, e.g., bandwidth. In addition, the possible quality of service that is achievable may vary over time due to variations in the load on the network. The feasibility of achieving a particular level QoS service management depends on the design of a network, the degree to which it collects network utilization level data and its ability to allocate network resources appropriately in response to QoS requests.

Value Type

QoS required

Value Set

- None—No QoS Support—No specified dynamic QoS request capability either in protocols, service interfaces, APIs, or other interface types.
- Static QoS Classes—No Variability—A fixed number and type of static QoS classes offered as a result of a request, but no dynamic capability to tailor a class to requestor specific needs.

- Hybrid Static and Dynamic QoS Classes—A fixed number and type of Static QoS classes with a set of dynamically tailorable QOS classes.
- Fully Dynamic—All QoS classes tailorable to requestor requirements.

Example

Voice over IP requires low latency transmission to avoid gaps in the information. It is not usually bandwidth intensive, so may be sent via best effort QoS, with reasonable intelligibility. Streaming video sent over IP, however, may require both low latency and a high bandwidth allocation to achieve reasonable performance.

3.4.5 Interface Development Complexity Dimension – 4.5

Technical feasibility generally diminishes as the complexity and difficulty of implementing the interfaces for a system or capability increase. This Interface Development Complexity dimension is more an acquisition-time concern than a run-time concern (although it may also impact the ease of change/evolution over the life cycle of a system/interface). This dimension is best measured relative to the overall scope of the envisioned system or capability. When an interface needed to affect a capability begins to dominate the development size of the system being acquired, sponsors may well consider it to be technically infeasible. They may also question whether the technical architecture assumptions might need to be reexamined and newer/alternative technologies and technology standards considered.

While the bulk of the development effort in the area of network interfaces for network-centric systems is usually focused on application level interfaces such as discovering, invoking, and parsing or utilizing mission services accessible over the network, some leading-edge system-level and communications infrastructure type services may also require custom development in certain cases. This dimension is intended to capture all development effort devoted to achieving a capability by making systems interact with other systems and the network infrastructure services over the network, whether at the mission application level or at the infrastructure level. Until they evolve to the point of being commodity capabilities, complex network interfaces and their associated development costs and risks are a significant inhibitor of network centricity in capability implementation.

Value Type

Real (0 .. 100)

Value Set

The value is the percentage of the overall system/capability size, however, size is measured for a specific set of system developments/acquisitions that is taken up by the implementation of the interfaces. This can be measured in terms such as total lines of code by system component, program budget allocation by WBS, or some other means. The important point here is the relative effort or budget devoted to achieving network-centric interaction with other systems as opposed to implementing the functions the system or capability is responsible for. For example, if half the cost of a system is devoted to implementing a client workstation user interface (or a web browser interface), and to building mission planning software, while the remainder of the development is devoted to supporting unanticipated network users (not part of the local user account management services), making mission planning data available through the network and obtaining mission resource data from

other systems on the network, the value for this dimension would be 50%—a fairly network-centric measurement.

As systems increasingly adopt a services architectural paradigm and rely more on services provided by other systems and network/enterprise service providers, scores upward of 90% may be achievable on this dimension. This is in sharp contrast to traditional system development ratios where interaction over the network typically amounted to less than 10% of development costs. It should also be noted that the cost per unit capability should go down as the level of network centricity on this dimension goes up. More effort may be devoted to interaction over the network, but the cost of such interaction is usually significantly less than the cost of developing/providing the service being accessed over the network.

3.4.6 Technology Readiness Level for System Connections Dimension – 4.6

The technology readiness level (TRL) for System Connections dimension applies in the case where an operational capability requires technical architecture elements that are not yet part of established baselines for mature standards and products. In other words, the capability is feasible if we relax the technical maturity level below production level. However, lower TRLs imply added development cost and technical/schedule risks and thereby impact the feasibility of implementing a particular capability. The primary purpose of TRLs is to help management make decisions concerning the development, transitioning, and application of technology. Advantages include the following:

- Provides a common understanding of technology status.
- Supports risk management.
- Aids decisions concerning technology funding.
- Aids decisions concerning transition of technology.

Value Type

Integer (1 .. 9)

Value Set

The TRL framework summarized in Table 3-26 is taken from the U.S. National Aeronautics and Space Administration (NASA). NASA initiated TRL framework development in the late 1980s, evolved it in the 1990s, and has applied it widely within its organization.

Although the framework is couched in terms that are specific to the NASA mission, it is straightforward to adapt it to different contexts, and it has, in fact, been widely adapted for use in other organizations. In the early 2000s, the U.S. DOD adapted the NASA TRL framework to support technology management efforts within the military. The DOD TRL framework is described in the *DOD Technology Readiness Assessment Deskbook*. It is defined to apply to all technology, hardware and software. The deskbook also includes a refinement of the TRLs defined specifically for software technology. Furthermore, the U.S .Air Force has developed a TRL assessment tool called the Technology Readiness Level Calculator that is readily available.

Table 3-26. Technology Readiness Levels (NASA)

Technology Readiness Level	Description

Technology Readiness Level	Description
1. Basic principles observed and reported	The lowest "level" of technology maturation. At this level, scientific research begins to be translated into applied research and development.
2. Technology concept and/or application formulated	Once basic physical principles are observed, then at next level of maturation, practical applications of those characteristics can be "invented" or identified. At this level, application is still speculative: there is not experimental proof or detailed analysis to support conjecture.
3. Analytical and experimental critical function and/or characteristic proof of concept	At this step in maturation process, active research and development (R&D) is initiated. This must include both analytical studies to set technology into an appropriate context and laboratory-based studies to physically validate that nalytical predictions are correct. These studies and experiments should constitute "proof-of-concept" validation of applications/concepts formulated at TRL 2.
4. Component and/or breadboard validation in laboratory environment	Following successful "proof-of-concept" work, basic technological elements must be integrated to establish that "pieces" will work together to achieve concept-enabling levels of performance for component and/or breadboard. This validation must be devised to support concept that was formulated earlier, and should also be consistent with requirements of potential system applications. Validation is relatively "low-fidelity" compared to eventual system: it could be composed of ad hoc discrete components in laboratory.
5. Component and/or breadboard validation in relevant environment	At this level, fidelity of component and/or breadboard being tested has to increase significantly. Basic technological elements must be integrated with reasonably realistic supporting elements so that total applications (component-level, subsystem level, or system-level) can be tested in "simulated" or somewhat realistic environment.
6. System/subsystem model or prototype demonstration in a relevant environment (ground or space)	Major step in level of fidelity of technology demonstration follows completion of TRL 5. At TRL 6, a representative model or prototype system or system—which would go well beyond ad hoc, 'patch-cord' or discrete component level breadboarding—would be tested in relevant environment. At this level, if only "relevant environment" is environment of space, then model/prototype must be demonstrated in space.
7. System prototype demonstration in a space environment	TRL 7 is a significant step beyond TRL 6, requiring actual system prototype demonstration in a space environment. Prototype should be near or at scale of planned operational system and demonstration must take place in space.
8. Actual system completed and 'flight qualified' through test and demonstration (ground or space)	In almost all cases, this level is end of true "system development" for most technology elements. This might include integration of new technology into existing system.
9. Actual system "flight proven" through successful mission operations	In almost all cases, end of last "bug fixing" aspects of true "system development." This might include integration of new technology into an existing system. This TRL does not include planned product improvement of ongoing or reusable systems.

NASA has augmented these basic levels with a layer of additional information that defines a set of TRL ranges that could be viewed as technology life-cycle phases. These are depicted in Figure 3-12.

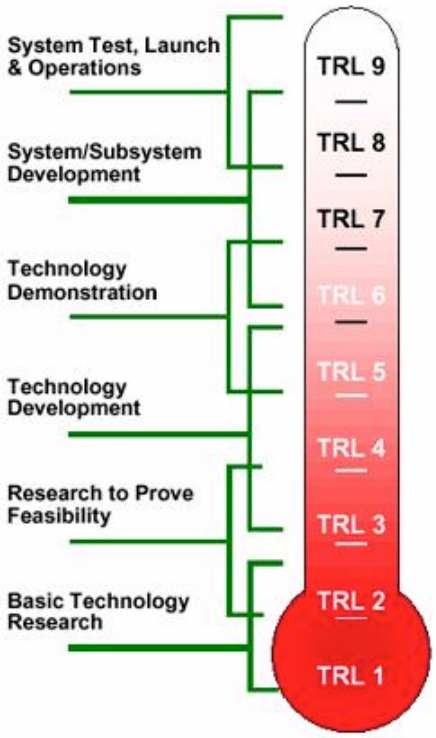


Figure 3-12. NASA Mapping of TRLs into Technology Life-Cycle Phases

The value set for any given application of the TRL dimension could include just the nine TRLs themselves (as described previously or adapted to the given context), or could be augmented to include a definition of TRL ranges with prescribed meanings as in Figure 3-12.

In either case, the integer value scale in this dimension is inverted from the norm (i.e., moves from high numbers to low) to indicate that higher TRLs more readily support interface development and integration (i.e., support greater feasibility and lower risk) than lower TRLs.

Example

A new concept for command and control systems is developed and an R&D project is initiated to explore the viability and applicability of this concept. Some prototype web services are developed to support implementation of this concept. These web services are applied within a military exercise alongside existing capabilities to demonstrate the new approach and evaluate its efficacy. The TRL of these web services is likely level 5, and the technology is still in the development and demonstration stage.

4. Emerging Dimensions

As described in Section 3, a number of SCOPE model dimensions have been conceptualized, but as yet lack sufficient practical application experience to establish confidence in the WG membership that the subdimensions and value sets are useful in conducting actual network centric assessments of real systems or capabilities. Nonetheless, the WG membership felt that the work done to date by the WG is valuable enough to share with the broader community as emerging dimensions. This work is also presented here in part to further engage the community and attract participation in evolving future versions of the SCOPE model. The section structure mirrors that of Section 3, and the section headings include the original section numbers the dimension in question has in Section 3. Typically, Section 3 included a short description of the intent or rationale for the dimension and pointed the reader to the corresponding subsection in Section 4.

4.1 Emerging Net-Readiness Dimensions

4.1.1 Information Assurance Capability Dimension - 1.3

This dimension addresses the Information Assurance (IA) capabilities and mechanisms that are specific to services and information exchanges across the network. The IA dimension is more critical than ever in net-readiness, since services and stored information may be exposed to anyone with network access. Interdependence among systems requires assurance that needed services and information are available at all times and that they accurately reflect the operational situation, or other situational awareness need for the application domain, such as Emergency Disaster Response coordination situational awareness. Balancing between information protection and information access in a net-centric environment implies greater information assurance awareness and enforcement at all levels and for all assets involved in the network including interfaces, services, and for information assets, not just at the "perimeter" of the network. Conversely, net-centric enterprise services must provide information and services that allow implementations to understand access rights that apply and enforce/enable them.

Some definitions of IA include the following:

The U.S. government's definition[8] of IA is as follows:

"*…Measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and nonrepudiation. These measures include providing for restoration of information systems by incorporating protection, detection, and reaction capabilities."*

[8] http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf

United States Intelligence Community [9]

"***Information assurance***-*information operations (IO) that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and nonrepudiation. This includes providing for restoration of information systems by incorporating protection, detection, and reaction capabilities*."

Wikipedia[10] Information Assurance Definition

"***Information Assurance*** *(IA) is the practice of managing information-related risks. More specifically, IA practitioners seek to protect the confidentiality, integrity, and availability of data and their delivery systems, in addition to ensuring adequate authentication and nonrepudiation. These goals are relevant whether the data are in storage, processing, or transit, and whether threatened by malice or accident.*

Information Assurance is closely related to information security and the terms are sometimes used interchangeably. However, IA's broader connotation also includes reliability and emphasizes strategic risk management over tools and tactics. In addition to defending against malicious hackers and code (e.g., viruses), IA includes other corporate governance issues such as privacy, compliance, audits, business continuity, and disaster recovery. Further, while information security draws primarily from computer science, IA is interdisciplinary and draws from multiple fields, including fraud examination, forensic science, military science, management science, systems engineering, security engineering, and criminology, in addition to computer science. Therefore, IA is best thought of as a superset of information security."

If IA depends on trusting a platform connected to the network, there also needs to be a way to verify that the platform is still trustworthy via the network. NCOIC is not focused on IA inside the platform boundary.

The emerging GIG IA architecture organizes IA capabilities in terms of four key IA functional areas: Assured Information Sharing, Highly Available Enterprise, Cyber Situational Awareness and Network Defense, and Assured Enterprise Management and Control. The following value set discussion is structured in accordance with these four areas, but potential values in some areas are possible in future versions of SCOPE. Further subdimensions will emerge as part of this process.

The Net-Readiness IA dimension is, of course, closely coupled with the Trust and IA Policy-Driven Constraints dimensions. The organizational context and associated policies will drive the technology mechanisms employed to achieve IA for net-readiness.

[9] http://www.intelligence.gov/0-glossary.shtml
[10] http://en.wikipedia.org/wiki/Information_assurance

Value Set

- Assured Information Sharing
  - Identification and Authentication
    - None.
    - Distinct authentication with each service provider based on system infrastructure identity.
    - Distinct authentication with each service provider based on network identity.
    - Single sign-on support based on network identity.
    - Enterprise-wide PKI support.
    - Cross-domain PKI support.
  - Labeling
    - Assured binding of labels to all required objects.
    - Label granularity supported.
  - Authorization
    - Risk-adaptive access control mechanisms.
      - Consider mission need, people trustworthiness, info policy, environmental risk factors.
    - Discretionary access control mechanisms.
      - Person-based.
      - Role-based.
    - Mandatory access control mechanisms.
      - MSL.
      - MLS.
      - Compartmented.
    - Cross-domain.
    - Authorization granularity supported.
- Highly Available Enterprise
  - Transport partitioning for IA (e.g., by COI).
  - IA policy-based routing mechanisms.
  - Protection of information in transit.
    - None.
    - Link encryption.
    - Network encryption.
    - Application encryption.

    - Virtual private networks.
    - Strength of encryption at each level.
- Cyber Situational Awareness and Network Defense
  - IA sensors and associated data management.
  - Vulnerability assessment and management.
  - Attack analysis and response.
  - Quarantining.
- Assured Enterprise Management and Control
  - Security management—identity, privilege, policy, key/cert.
  - Protection of management and control information.
- Audit mechanisms

4.1.2 Semantic Interoperability - 1.4

Semantic Interoperability entails the networking concept for mutually consistent semantic interpretation of intention and shared knowledge within a situational and purposeful context, as a result of a semantic interaction, where *intention, context, and knowledge* are explicitly represented and expressed in some language of discourse, or are implied by convention and use. Semantic Interoperability characterizes the compatibility of the descriptive elements (Figure 4-1) of the semantic interaction expression and the corollary representations and models that the interacting agents use to semantically interpret the interaction. Context provides the logic and additional information of how each network element should interpret the intention and shared knowledge of each interaction.

In this section, the focus is on characterizing the ability of any network and its elements to support Net Centric Tenets that enable consistent interpretation of each interaction from a semantic interoperability perspective. These new semantic concepts are associated with creating explicit representations of the semantics of interactions and models that can be used by network elements or agents to achieve this mutually consistent interpretation. These semantic concepts will influence all architectural layers of a network (Figure 4-2).

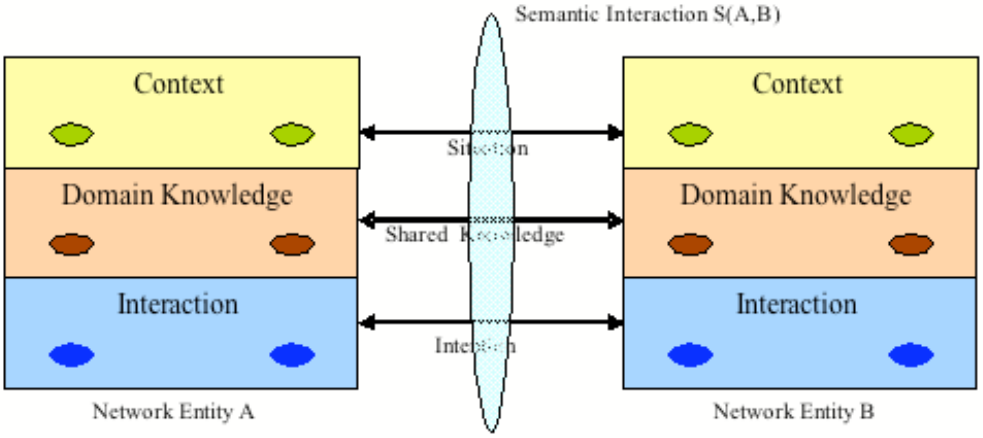


Figure 4-1. Semantic Interaction Elements

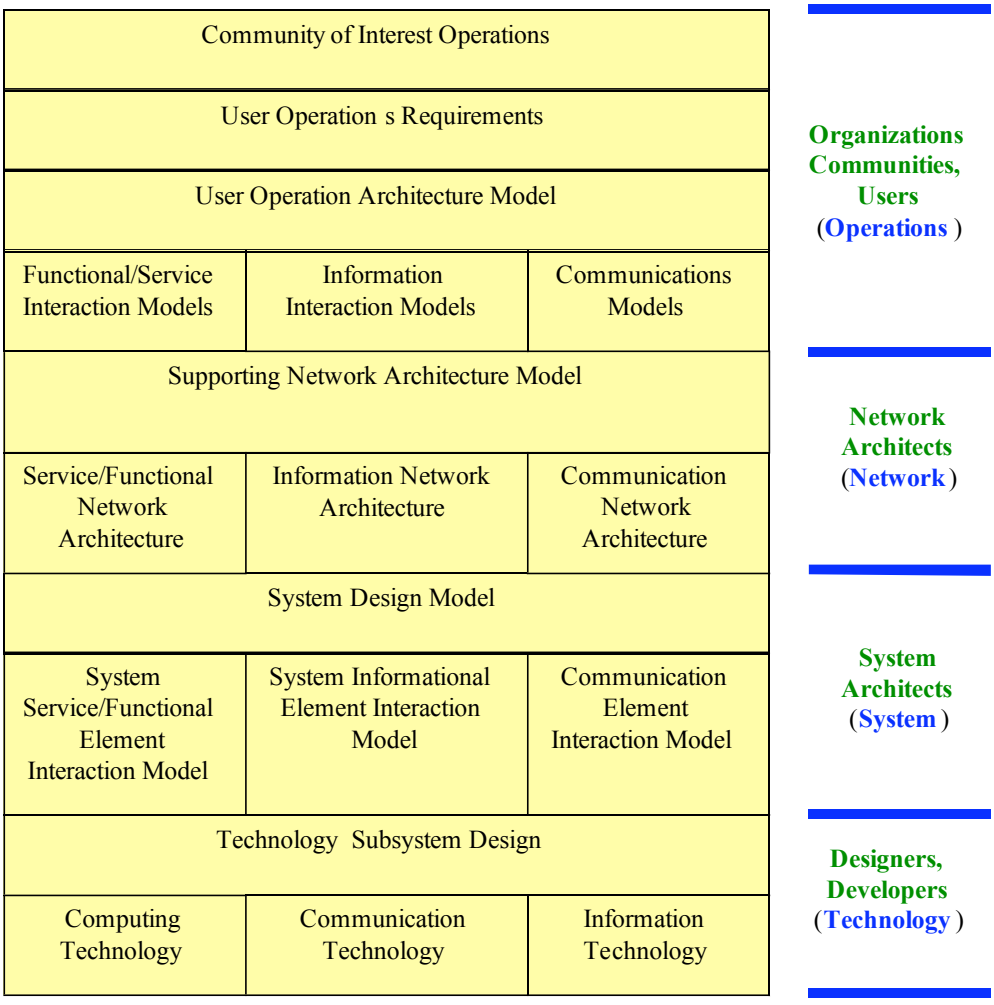


Figure 4-2. Semantic Interaction Layers in a Network and COIs That Use and Define Semantics

The set of semantic interoperability dimensions are illustrated in Figure 4-3 and organized in a taxonomic fashion.

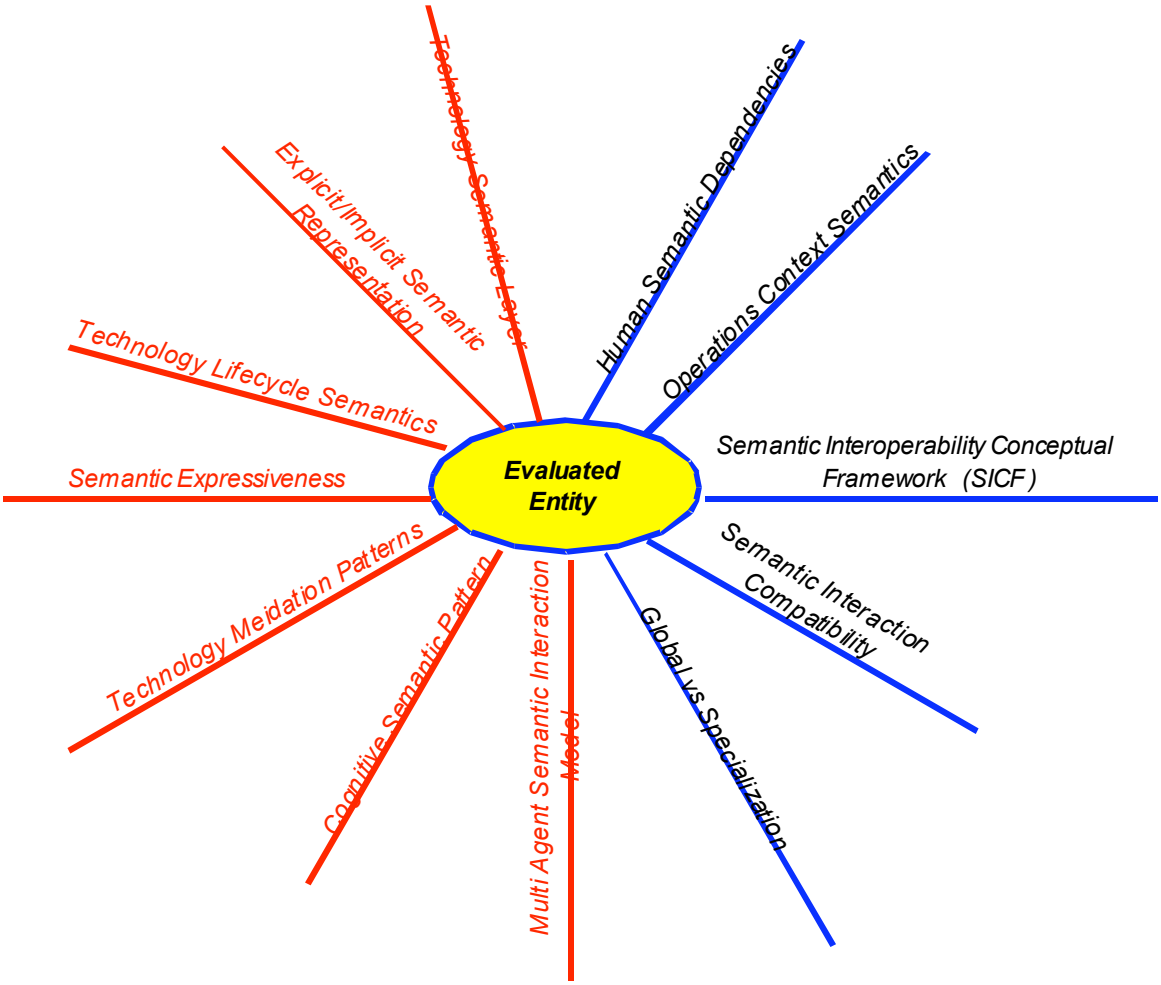


Figure 4-3. Semantic Interoperability Dimensions (Red Shows Net-Ready Dimensions, Blue Shows Capability Dimensions)

Taxonomic Overview of Semantic Interoperability Net-Ready Dimensions

- Semantic Technology Mediation Patterns
- Technology Mediated Semantic Network Layers
  - Operations
    - Community of Interest
    - User Operations Requirements
  - User Operational Architecture Model
    - Functional/Service Interaction Model
    - Information Interaction Model
    - Communications Model

  - Supporting Network Architecture
    - Service/Functional Network Architecture
    - Information Network Architecture
    - Communication Network Architecture
  - System Design
    - System Service/Functional Element Interaction Model
    - System Informational Element Interaction Model
    - Communication Element Interaction Model
  - Technology
    - Computer technology
    - Communication Technology
    - Information Technology
- Explicit/Implicit Semantics
  - Human Semantic Explicitness
  - Technology Semantic Explicitness
- Technology Life Cycle
  - Requirements
  - Architecture
  - Design
  - Development
  - Deployment
  - Operations
- Cognitive Semantic Pattern
  - Cognitive Agent Semantic Interactions Pattern (CA-CA)
    - Internal Agent World Model
    - Agent Cognitive Processing
  - Hybrid Agent Semantic Interactions Pattern (CA-RA)
    - Design Agent (DA) World Model
    - CA World Model
    - RA Stimulus – Response Processing
  - Reactive Agent Semantic Interactions Pattern (RA-RA)
    - Design Agent (DA) World Model
    - RA Agent Stimulus – Response Interactions
- Detailed Multi-agent Semantic Interaction Model

  - (CxC), (DxD), (IxI) Explicit and Interdependent Context, Domain and Intention Models and Interactions
  - (CxC), (IxI) Explicit and Interdependent Context and Intention Model, Implicit Independent Domain Model
  - (DxD) Implicit and Independent Context, Intention Model, Explicit and Interdependent Dependent Domain Model
  - (DxD), (IxI) Explicit and Interdependent Domain and Intention Models, Implicit Context
- Semantic Expressiveness

4.1.2.1 Semantic Technology Mediation Patterns - 1.4.1

The intent is to describe the salient concepts of semantics that enable a capacity for mutually consistent understanding of shared information in a technology mediated information and communication network environment. For this effort, NCOIC is not concerned with how the brain processes sensory information and relates it to various concepts, nor in the theory of linguistics and how we use language to represent these concepts in our communication and interactions with other humans; but rather how technology design assumptions and implementations can assist or inhibit groups of people to reach a reasonably consistent understanding of shared concepts expressed in their interactions. As a concrete example the human interface to most kinds of technology has evolved to become more flexible and powerful by enabling a context interpretable relationship between the display, the implied functional meaning of the icons or text lists within the display, and a soft selection button on the device for indicating the user's indication of which item is of interest for this interaction. Subsequent to the user's communicating the selection, the display will invariably change to indicate a new functional concept list for selection again by the user. In this way the technology can present a set of sequenced interactions with relevant information that enable the user to refine his intent, and the technology to refine a response based on the user's selection and supplied information. This example indicates semantic interaction and interoperability between a human and technology. Later in this document, it will be shown that the semantics of this interaction was defined by another human (a designer) at an earlier stage of technology development life cycle. Thus all human interactions with technology, require semantics to be defined by other humans.

Similar approaches can be used when a community of users are involved in some form of collaboration in which technology provides choices and relevant information to each user with respect to a predefined process of collaboration for some particular purpose and the role of each participant. eBay®'s auction services on the WWW is an excellent example of a technology mediated network human collaboration; the selling and buying of products within a predefined auction process. Again this auction process being used by the WWW community had its semantics of interaction previously defined by humans, a designer (Figure 4-4)

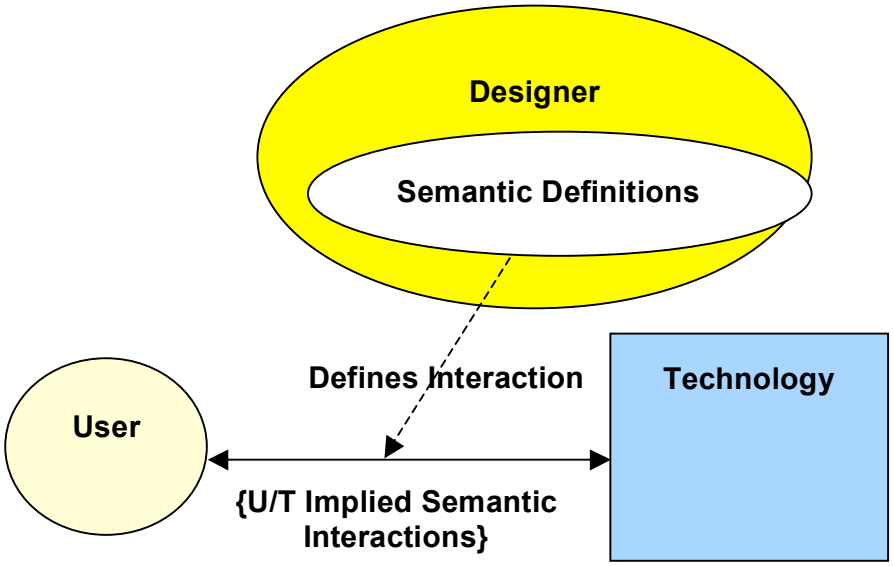
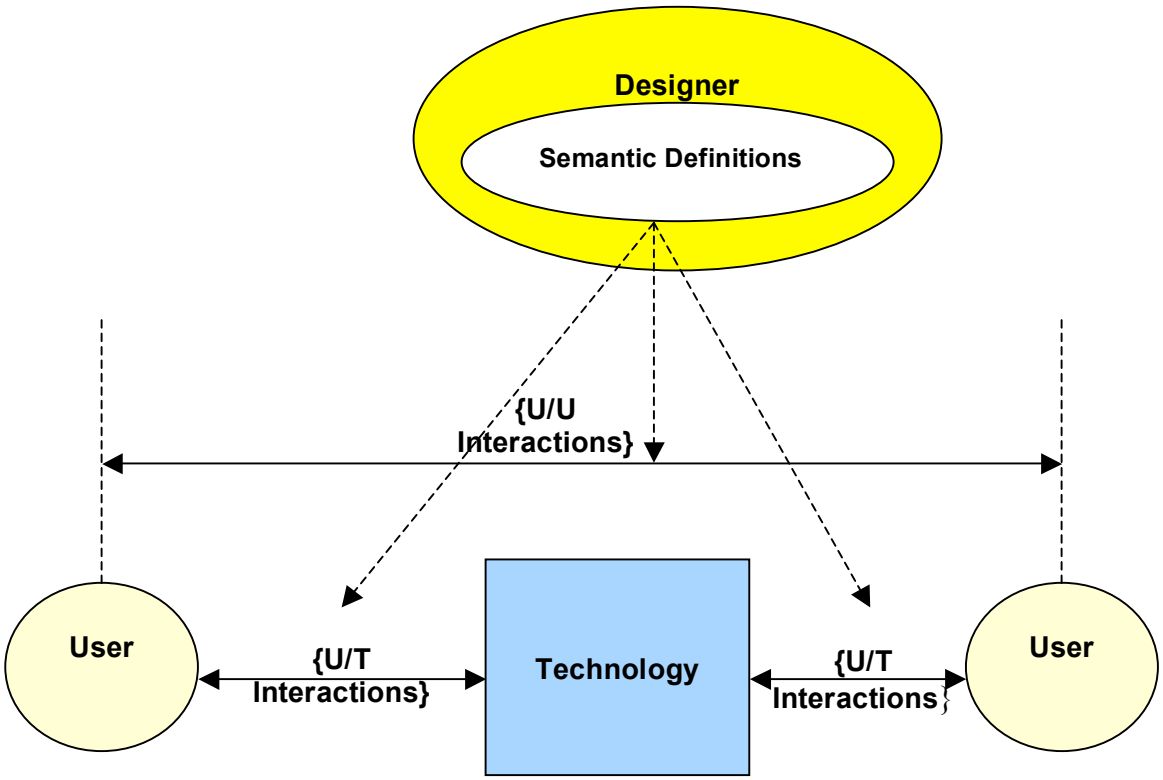


Figure 4-4. Human – Technology Mediated Semantic Interactions

The semantic interoperability model is expanded to include interactions not only between users and technology, but also interactions between users mediated by technology. We are interested in understanding and characterizing those aspects of semantics associated with technology mediated collaborations, though we will make some assumptions about using relational concepts for characterizing COIs and their domain knowledge relevant to their specific purposes.

In this model, each ordered triple (U-SI-T), (T-SI-T), (T-SI-U) has a defined semantic interaction, SI, between the interacting entities, U,T. The end–to-end semantics are thus a result of the concatenation of the semantic interactions and the mediating influences of each interacting entity. We always make an assumption that the end points have a human interpretation of the semantics of the interaction, even if the end point is a technology, T; we assume that the semantic interpretation implemented in T is defined by a human designer. When the end point is technology, the semantics of its interactions are implied through a reference to semantic definitions created by the designer. For current systems, all of the semantic interactions, SI, regardless of endpoint types (T,U) have an implicit reference to an external set of semantic definitions defined by designers. More recent semantic web technologies are representing some of these human semantic definitions into the technologies themselves, so that the semantic interactions, SI, will now have some explicitly and transparency within the implementation (Figure 4-5).



Technology Mediated U-T, U-U Semantic Interactions

Figure 4-5. Networked Users technology Mediated Semantic Interactions

4.1.2.1.1 Semantic Interaction Technology Mediation Patterns - Values

One of the four following values should be assigned to the type of interactions being evaluated. In general, different combinations of these can occur, e.g., multiple technologies intermediate between end-users. For example:

1. **U-SI-U**, 'not technology mediated '
2. **U-SI-T(D)**, ' implied designer semantics in the interface definition'
3. **U-SI-T-SI-U**, 'technology mediated'
4. **U-SI-T-SI …-U**, 'complex technology mediated form'

4.1.2.2 Technology Mediated Semantic Network Layers- 1.4.2

In technology mediated semantics, especially network examples, there are multiple layers of semantic abstractions, where each layer has semantic definitions appropriate to its purpose. The "user" and "designer" discussed previously are now refined according to their contextual role in using or creating a technology mediated network solution, and the relevant domain knowledge for their purposes. Figure 4-6 illustrates that technological-based networking solutions have a hierarchical model of semantic dependencies, where each layer abstracts to a higher level of abstraction the concepts of the lower level, and that networking agents and elements typically interact with concepts within their layer of concern.

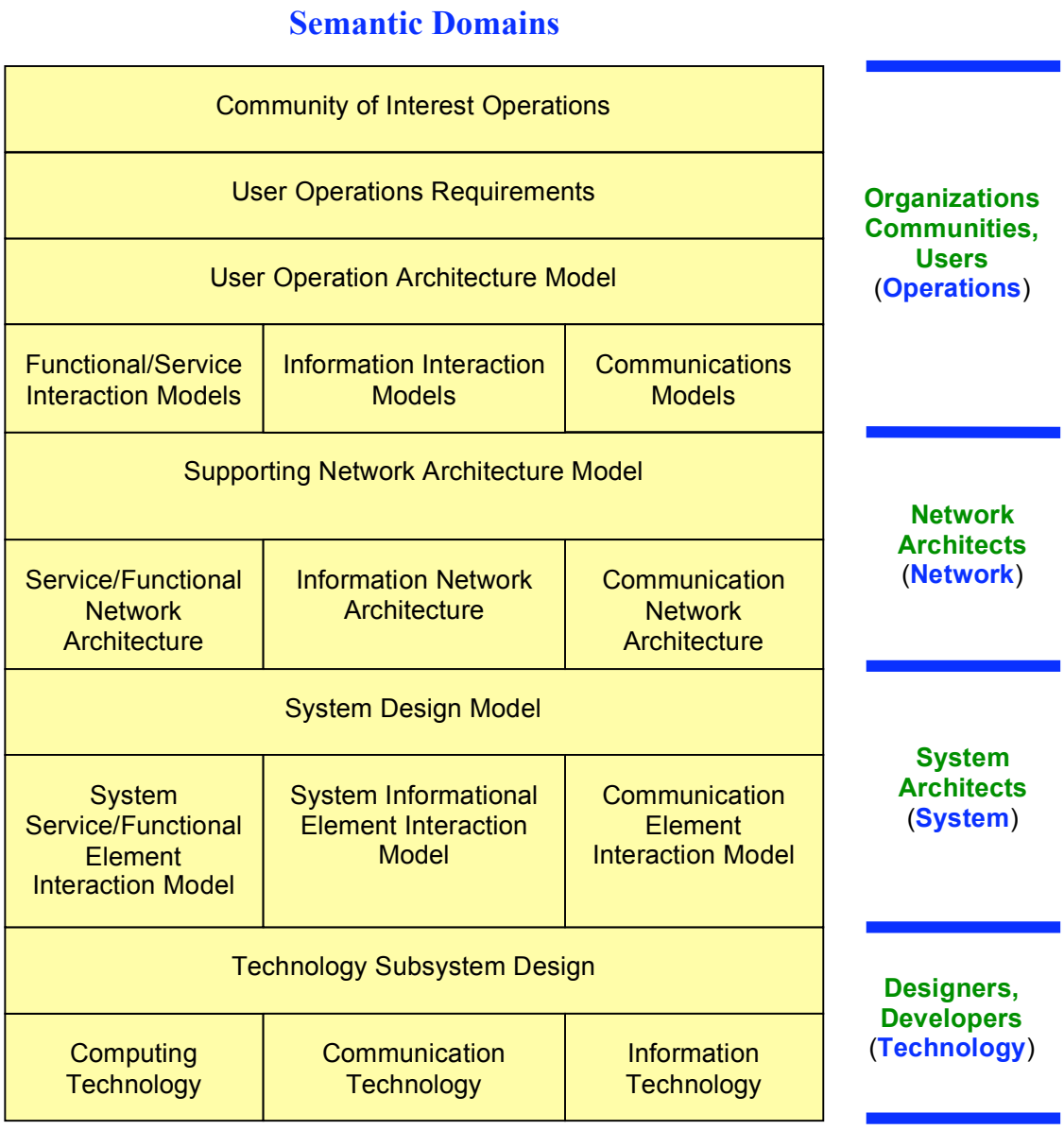


Figure 4-6. Hierarchical Semantic Abstraction Layers from a Network Technology Perspective

The subdimensions are taxonomic in structure and follow Figure 4-6. The aspect being characterized may be associated with one or more of the following conceptual layers, and one or more of the subdimensions should be selected to identify the architectural layers associated with the semantic interactions. We will use the set theoretic values to describe the compatibility relationships for all the interacting network elements for each appropriate subdimension.

For example if the evaluation is to determine Semantic Interoperability between organizations, then the Operations and User Operational Architecture Model would be appropriate selections.

If the evaluation is between systems, or systems of systems, then the top three layers will be important, where the operations and user operational architecture inform the need for system architectures supporting particular user interactions, while the network architecture identifies how the semantic of

these interactions are mediated and not disrupted or invalidated by the network services and systems functions.

1. Operations
    a. Community of Interest
    b. User Operations Requirements
2. User Operational Architecture Model
    a. Functional/Service Interaction Model
    b. Information Interaction Model
    c. Communications Model
3. Supporting Network Architecture
    a. Service/Functional Network Architecture
    b. Information Network Architecture
    c. Communication Network Architecture
4. System Design
    a. System Service/Functional Element Interaction Model
    b. System Informational Element Interaction Model
    c. Communication Element Interaction Model
5. Networking Technology
    a. Computer Networking technology
    b. Communication Networking Technology
    c. Information Networking Technology

4.1.2.2.1 Semantic Networking Layers Subdimensions and Values

Each of the subdimensions have the following set (Table 4-1) of compatibility values chosen as representative of the semantic compatibility of the interacting elements within that layer.

Table 4-1. Semantic Network Layers Subdimension Values

Semantic Interaction Subdimension		Subdimension Value	
Operations	Community of Interest	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent
	User Operations Requirements		
User Operational Architecture Model	Functional/Service Interaction Model	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent
	Information Interaction Model		
	Communications Model		
Supporting Network Architecture	Service/Functional Network Architecture	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent
	Information Network Architecture		
	Communication Network Architecture		
System Design	System Service/Functional Element Interaction Model	• Disjoint	• Inconsistent

	System Informational Element Interaction Model Communication Element Interaction Model	• Overlap • Subset • Equivalent	• Consistent
Networking Technology	Computer Networking technology Communication Networking Technology Information Networking Technology	• Disjoint • Overlap • Subset • Equivalent	• Inconsistent • Consistent

4.1.2.3 Explicit and Implicit Semantics for Human and Technology Interactions - 1.4.3

In addition to the effects of globalization and specialization and their impact on consistent semantic interaction, consistent semantic interactions are also affected by the explicit or implicit nature of the semantic representation and definitional relationships. We are thus interested in characterizing whether the semantics of interaction are explicitly or implicitly defined for both human and technology interaction.

The ability to achieve mutual and consistent semantic interpretation is biased by the level of explicitness of semantic definitions for the interaction. Implicit semantic definitions are the most problematic due to their greater potential for erroneous assumptions about the meaning of the exchanged expressions in the interaction. The following model (Figure 4-7) indicates that all technology semantic interactions are subsets of defined human semantic interaction definitions, and that they can either have explicit, hybrid, or implicit definitions.

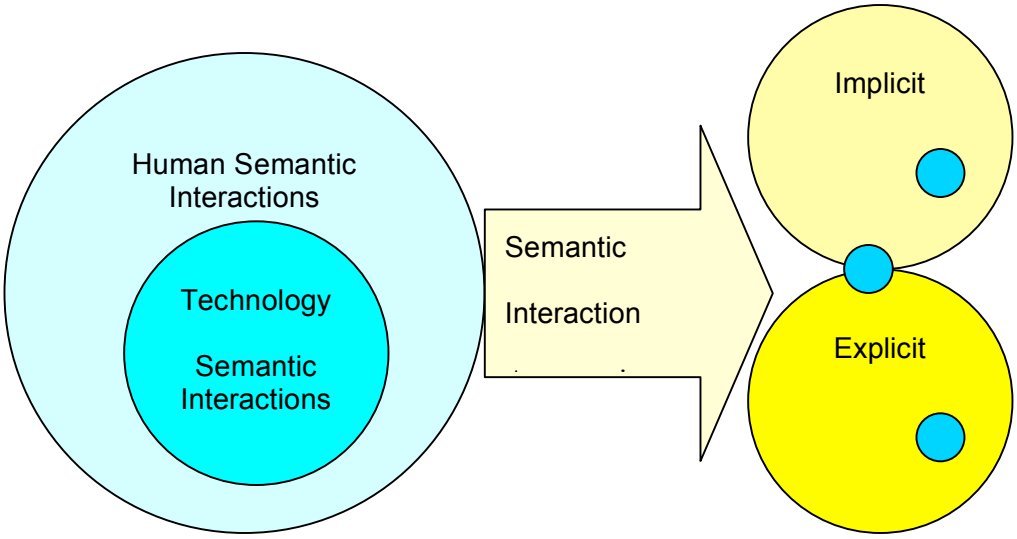


Figure 4-7. Explicit and Implicit Semantic Interactions

4.1.2.3.1 Explicit and Implicit Semantics for Human and Technology Interactions - SubDimensions and Values – 1.4.3.1

If we define human semantic interaction as depending on the explicit understanding and representation of context, domain knowledge, and the interaction intent, we can make the following axiomatic statements about the effects of explicit versus implicit semantic definitions. We differentiate between the problems of mutually consistent interpretation of interactions between humans and the

compatibility of interactions between networked technology elements (Figure4-8). We can state the following axiomatic assumptions about the technology and human semantic explicitness effects.

Human Semantic Explicitness

1. **(EH) Explicit—Human** semantic definitions **enable** mutually **consistent understanding** of **human interactions**
2. **(EH) Explicit—Human** semantic definitions **enable Explicit-Technology interactions** definitions
3. **(IH) Implicit—Human** semantic definitions **may enable** mutually consistent human understanding of interactions through shared background knowledge
4. **(IH) Implicit—Human** semantic definitions only enable **Implicit-Technology** interaction definitions

Technology Semantic Explicitness

1. **(ET) Explicit—Technology interactions** definitions **enables** mutually **consistent interpretation** of technology **interactions**
2. **(IT) Implicit** Semantic **Technology interactions** definitions **do not enable** mutually **consistent interpretation** of technology **interactions**

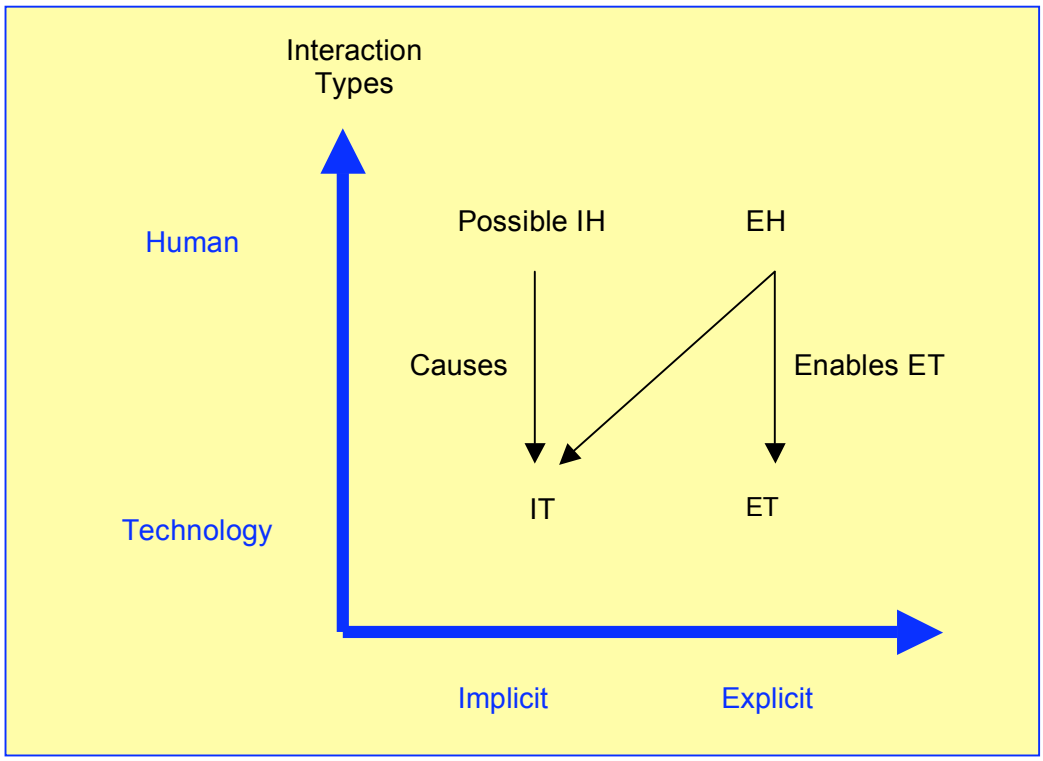


Figure 4-8. Explicit Versus Implicit Interaction Definitions and Effects on Mutual Understanding at Human Level and Compatibility at Technology Level

The key concept is that explicit human semantic interaction definitions and models will not only enable mutual understanding between networked humans, but also if applied to technology interactions definitions will enable technology interactions to be defined for compatibility.

When interoperability is important to a community, an attempt is made to reach agreement on the semantics to be used for interactions by establishing standards defining the metadata associated with data elements to be exchanged. Examples abound, such as the ANSI ASC X12e-business standards "*The EDI*[11] *standards are developed and maintained by the Accredited Standards Committee (ASC) X12. The standards are designed to work across industry and company boundaries. Changes and updates to the standards are made by consensus, reflecting the needs of the entire base of standards users, rather than those of a single organization or business sector. Today, more than 300,000 organizations use the 300+ EDI transaction sets to conduct business.* " The U.S. government is also active as exemplified by the Federal Enterprise Architecture (FEA)[12] Data Reference Model (DRM).

Regardless of how the interaction is described or how the data of exchange is represented there is a need to define the meaning of the interaction and the data exchanged so that others could design elements that could interact according to this definition. Typically, this level of semantic interaction definition is not completed sufficiently to achieve this goal.

Originally, it was assumed that simple metadata tagging of the data elements would provide sufficient specification to enable successful interoperation and mutually consistent interpretation by interacting elements within a community. It was assumed that the XML metadata element definitions for each data element, and extensive documents defining the intent of the use of the data, that the semantics would be consistently interpreted by each system and organization within the community. This has not proved to be the case for a variety of reasons, most notably the lack of semantic constraints and a semantic model for interpreting the exchanged data correctly. Only in the most simple of cases with small sets of metadata and data elements was success achieved.

4.1.2.4 Technology Life Cycle Semantic Dependencies - 1.4.4

Current and past technology development life cycles have hidden the definitional semantics as the technology life cycle progresses where normally explicit definitions are available in early requirements and design specifications, but are not normally available in later technology implementation or deployment or operational phases. Technology interaction semantics deals with how much of the human semantics is captured and inherent in the technologies themselves, e.g., computer languages, protocols, schemas, ontologies, metadata, web services, etc. Since all technology interactions were once designed with an original set of semantics, the problem for expansion or reuse is to recover the original semantics without error.

It is natural that technology-based systems and their interactions are derived from definitions created during the engineering process and that the semantics of any system interaction was originally defined with assumptions about context, domain knowledge and intention of the interaction.

Although technology implemented systems have this defined semantic model, it is typically not discernible in the final technology implementation, due to loss of the original semantics while creating translations to technological forms in the artifacts created during the processes associated with the different life cycle stages, e.g., requirements, design, development and deployment. This has had the effect of increasing the complexity, cost and risk for extending the scope of interoperability for systems.

[11] http://www.x12.org/x12org/about/faqs.cfm#a1
[12] http://xml.coverpages.org/ni2005-12-28-a.html

Technology Life Cycle Semantic Dependencies Subdimensions

1. Requirements
2. Architecture
3. Design
4. Development
5. Deployment
6. Operation

In computer systems, messages may be sent as part of a protocol definition and the interpretation of the message may be context dependent on the state of the receiver defined by some complex autonomous state machine. Without recourse to the document describing the state machine and familiarity with the protocol, a human would have a hard time determining the receiver's interpretation of the message. In this case, the semantics of the meaning of the message is context dependent on the state of the receiver and the state machine specification implemented in the technology of the receiver. Though the message may be observed, the state machine is not usually observable within the technology. It is usually hidden in the programmatic code for software implementations. In this case, the semantic interpretation is implicit where originally there was a human explicit definition, but the technical solution implies satisfaction to the human specification and only through testing is it determined whether the technology is consistent with the semantic definition of the state machine.

Another example of explicit semantics with a narrow interpretation (screen rendering of hypertext and navigation to other web sites) is the WWW with its syntactical Hypertext Markup Language (HTML), which has been very successful in enabling rendering of web site content by web browsers on computer screens and enabling navigation to other web sites. It has been noted that there were no semantic definitions describing the meaning of the web site content that could enable reasonable consistent interpretation by applications or web browsers. To assist in the achievement of mutual understanding of web content and services throughout the technology life cycle, the W3C has create a Semantic Web Initiative[13] and its first set of standards to add semantic metadata for web services and web content have been created in the form of XML, RDF, and OWL.

Other efforts have defined common metadata that could be used by organizations within a community of common interests. This has most notably taken the path of shared taxonomies and XML schemas, which provides explicit syntax and vocabulary, but relies on documents and participants in the community to reach agreements on the semantics. There is no explicit computer computable semantic metadata defined by XML, the meaning or semantics is implicit through shared definitions in documents or standards.

4.1.2.4.1 Technology Life Cycle Semantic Dependency - Values

Figure 4-9 illustrates the possible function of semantic definitional availability in an explicit manner as the technology life cycle phases occur. As newer systems are developed with increasing model-based architectures and design, and with increasing semantic representation, it is possible that the semantic

[13] http://www.w3.org/2001/sw/

availability improves through later phases of the technology life cycle. Therefore, the Table 4-2 provides the method to record the possible semantic definitional availability values for each technology life cycle phase.

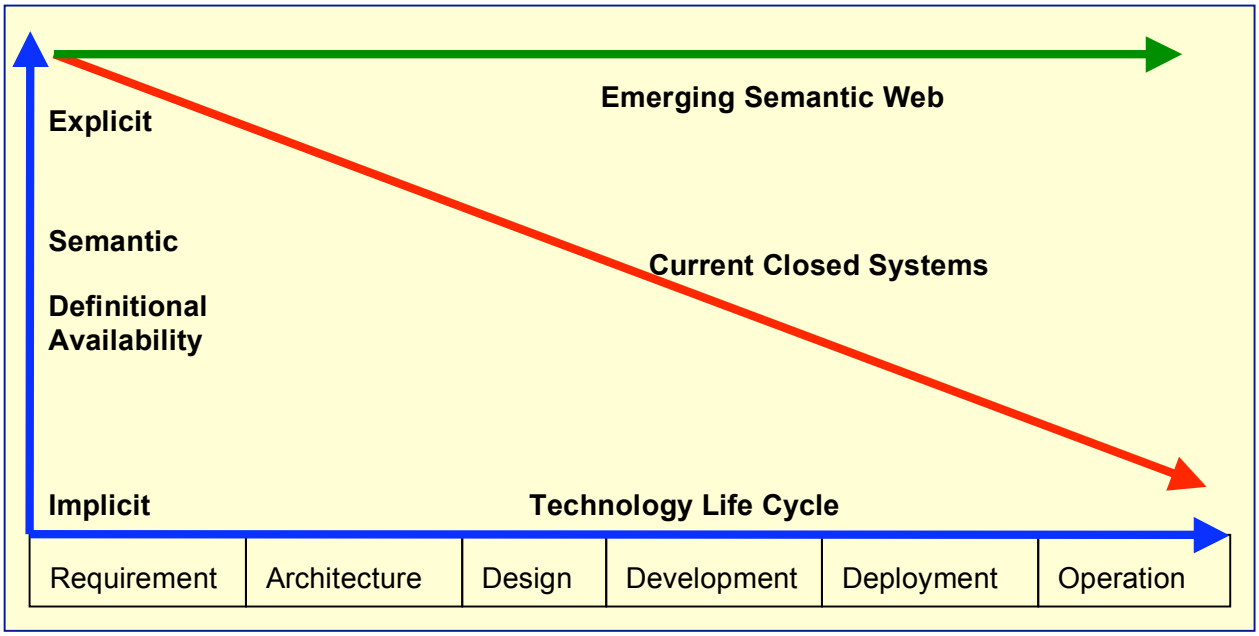


Figure 4-9. Semantic Definitional Availability During Technology Life Cycle

Table 4-2. Technology Life Cycle Semantic Dependency Values

Technology Life Cycle Subdimensions – Phase	Semantic Definitional Availability
Requirement	Explicit, implicit, not known
Architecture	Explicit, implicit, not known
Design	Explicit, implicit, not known
Development	Explicit, implicit, not known
Deployment	Explicit, implicit, not known
Operation	Explicit, implicit, not known

4.1.2.5 Cognitive Semantic Pattern Dimension (3.1.4.5)

In addition to the semantic assumptions of the interactions, the type of interacting agents should also be considered in their support a type of semantic interactions. Two types of agents[14] are considered for our analysis purposes, each having different semantic processing capabilities, e.g., cognitive agents[15] (CA), and reactive agents (RA). Cognitive agents have the ability to process expressions representing knowledge of the world against an internal world knowledge model, while reactive agents do not have an internal model representing knowledge about the world and thus cannot semantically interpret expressions about the world. The semantic interaction models described previously consisting of <context, domain knowledge intention> form the description of the type of interaction, while the agent

[14] The term agents are not necessarily meant to imply software agents, but only a conceptual model of a software system or design concept that either contains an internal knowledge model or not, cognitive or reactive designs

[15] "Multi-Agent Systems – An Introduction to Distributed Artificial Intelligence", Jacques Ferber, ADDISON-WESLEY, ISBN 0-201-36048-9

type combinations characterize their compatibility to interpret a semantic interaction type. Every interaction, whether by a cognitive agent or reactive agent design embodies these three semantic elements <context, domain knowledge, intention>. The form of the embodiment and semantic interpretation assumptions for each agent type are different. Three possible combinations of agent interactions have been identified: CA-CA, CA-RA, RA-RA, where the interactions have different semantic assumptions and characteristics.

The Cognitive Semantic Pattern subdimensions are as follows:

1. Cognitive Agent Semantic Interactions Pattern (CA-CA)
2. Hybrid Agent Semantic Interactions Pattern (CA-RA)
3. Reactive Agent Semantic Interactions Pattern (RA-RA)

4.1.2.5.1 Cognitive Agent Semantic Interactions Pattern (CA-CA) (3.1.4.5.1)

When cognitive agents interact, it is assumed that each agent has an internal knowledge model that can be used to guide all of its logical determinations about selection of actions and current state of affairs. Each exchanged expression as part of the interaction entails the <context, domain knowledge, intention> definitions and assumptions described previously and in most cases this is explicitly defined for each agent and transparent in the agent design. As long as the cognitive agent world models involved in the interaction are semantically compatible, and the expressions used at the interaction communication interface are translatable to the internal world model of each agent, then consistent semantic interactions and interoperability are possible. This compatibility is described in Figure 4-10 by using set relationships to determine the Venn intersections for semantic consistency in these three areas.

Subdimensions

1. Agent World Model
2. Agent Cognitive Processing

4.1.2.5.2 Hybrid Agent Semantic Interactions Pattern (CA-RA) (3.1.4.5.2)

With respect to hybrid situations where a cognitive agent interacts with a reactive agent, the cognitive agent has to bear the burden of semantically translating the sense/response interactions to the concepts contained in its internal knowledge model and consistent with the design agent model. Typically, the semantics of the sense/response reactive agent interactions are defined by a design agent, where these semantic definitions are usually in the form of documents. This situation is typically fraught with semantic interpretive errors of the sense/response interactions by design agents, especially if existing reactive agent implementation does not readily enable traceability to the design model (Figure 4-11).

Subdimensions

1. Design Agent (DA) World Model
2. CA World Model
3. RA Stimulus – Response Processing

4.1.2.5.3 Reactive Agent Semantic Interactions Pattern (RA-RA) (3.1.4.5.3)

In the case of interacting reactive agents, the only knowledge model defining semantics is intrinsic within the design agent creating the set of reactive agents. Thus, the semantics of the reactive interaction is no longer explicitly available from the interacting agents, but must be provided by the design agent to any other designer of a new reactive agent that it is desired to be compatible with. An analysis of the semantics of interacting reactive agents requires an understanding of the design model used to create each reactive agent and the behaviors associated with the set of sense/respond events for each reactive agent. To the extent that the sense/respond behaviors are similar for similar events, then the semantics of the reactive agent interaction are compatible. Communication network protocols are excellent examples of this kind of reactive agent interaction, where each reactive agent implements the same state machines defining the interacting protocols, e.g., their behaviors are the same for the same situation and internal state (Figure 4-12.

Subdimensions

1. Design Agent (DA) World Model
2. RA Agent Stimulus – Response Interactions

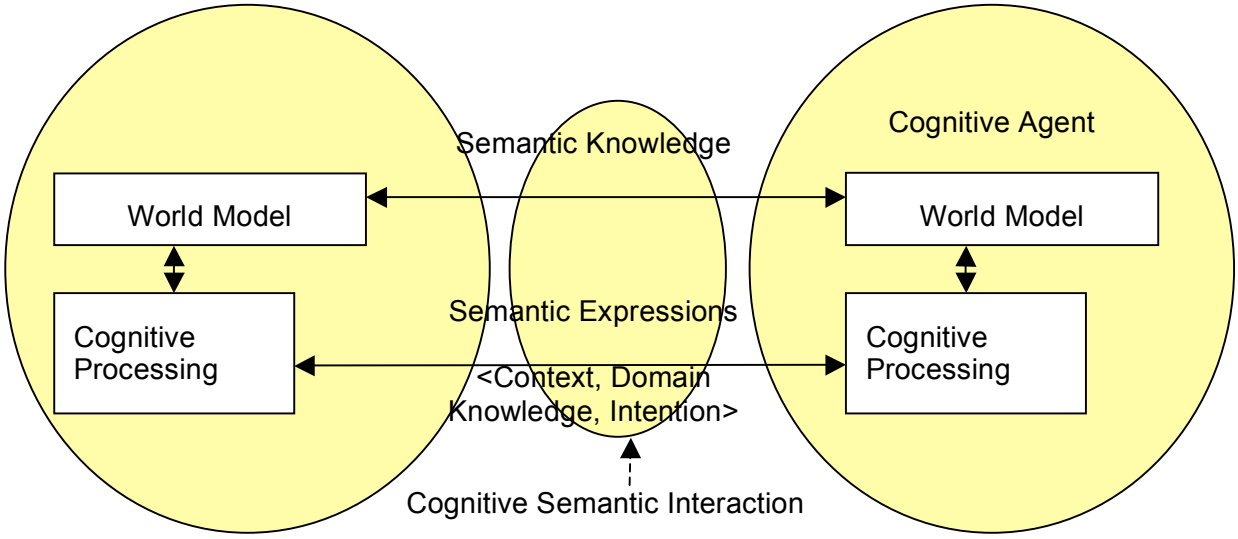


Figure 4-10 Semantic Interaction between Cognitive Agents (CA-CA)

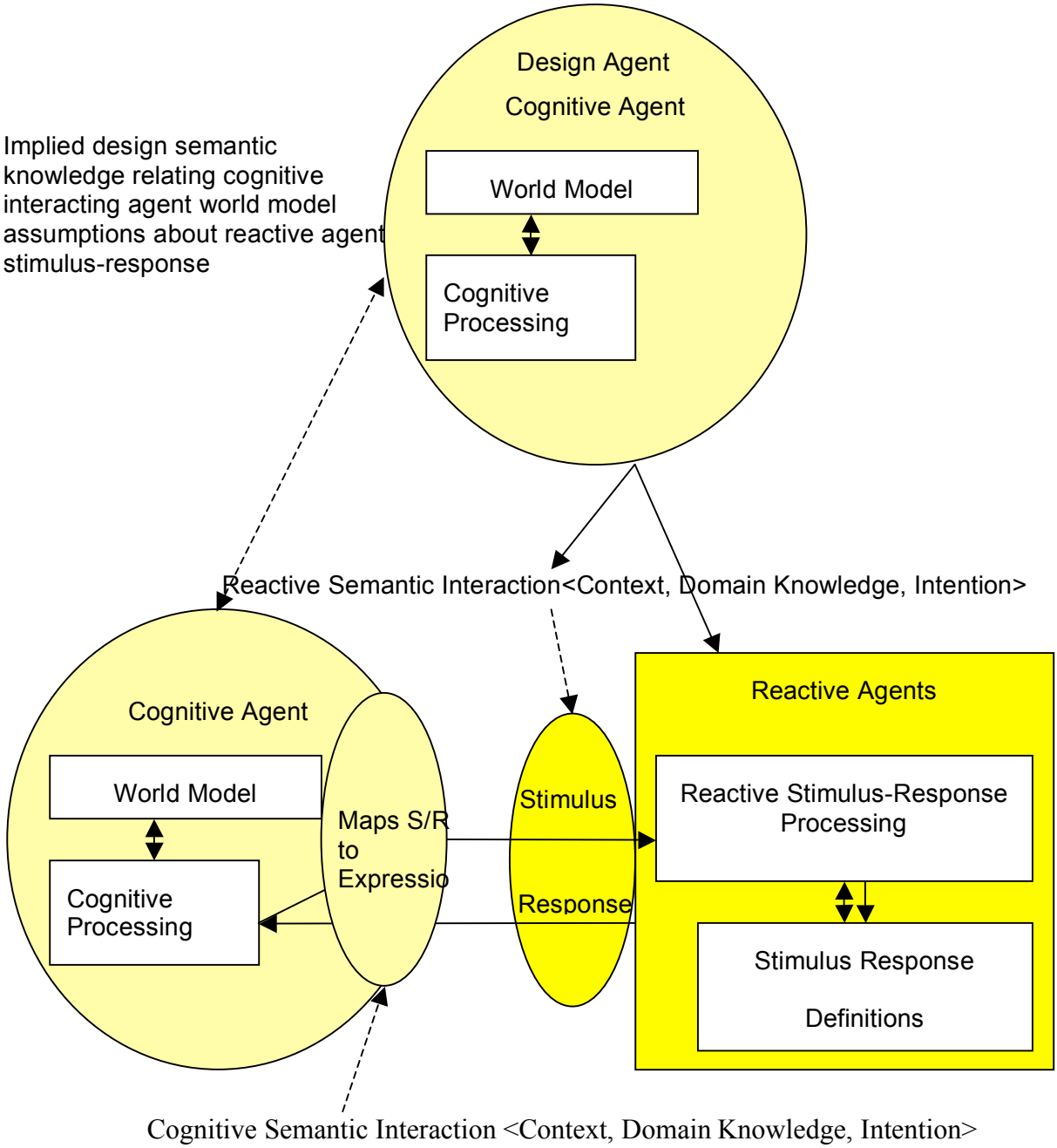


Figure 4-11. Hybrid Semantic Interactions between Cognitive and Reactive Agents (CA-RA)

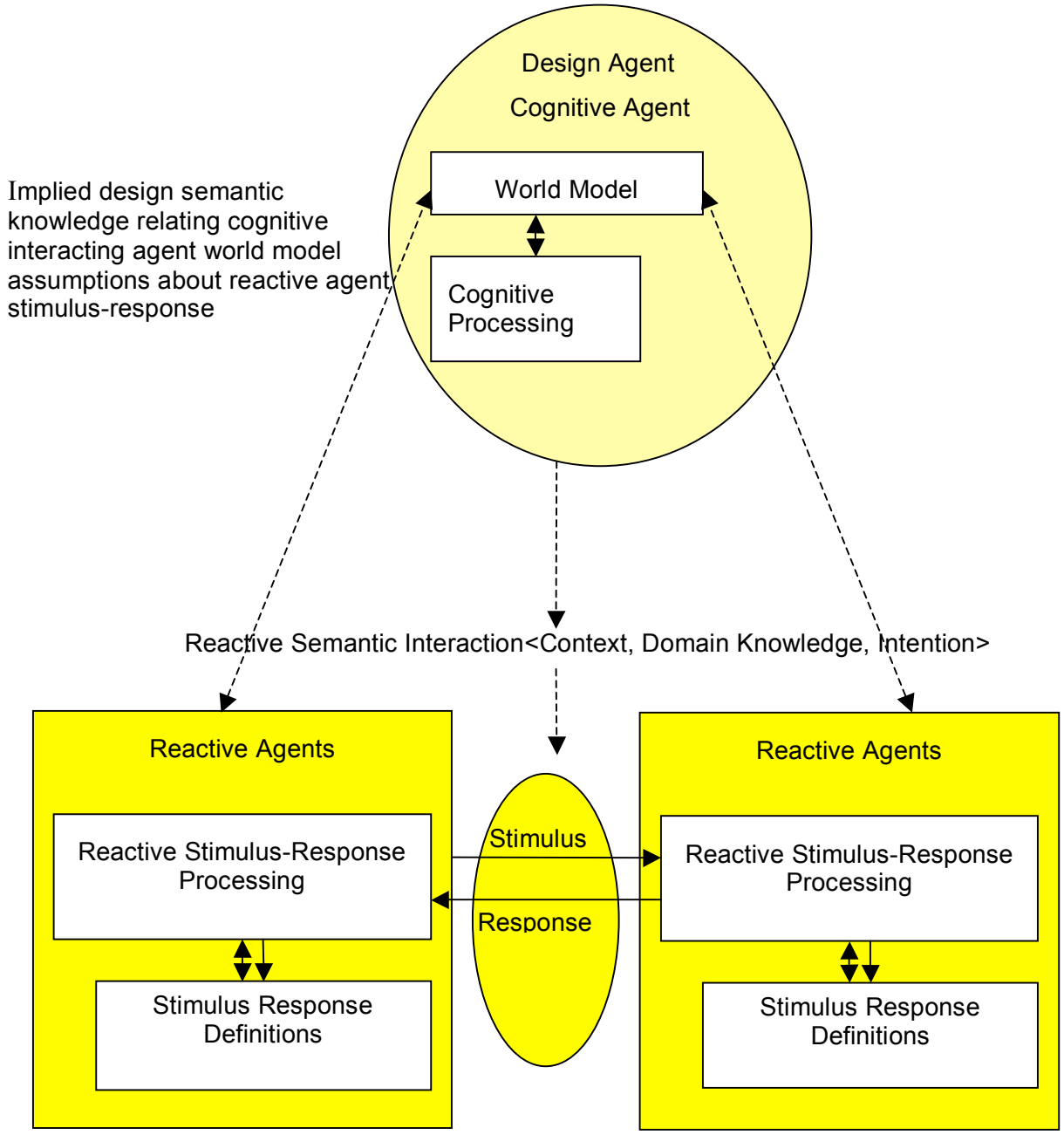


Figure 4-12. Reactive Agent to Reactive Agent Interactions (RA-RA)

4.1.2.5.4 Cognitive Semantic Dimension Pattern Values (3.1.4.5.4)

The evaluation should identify one of the three cognitive interaction patterns with the assumed semantic characteristics (Table 4-3).

Table 4-3. Cognitive Semantic Dimension Values—Cognitive and Reactive Designs

Cognitive Semantic Dimension Pattern Dimension	Subdimension	Dimension Values	Context	Domain Knowledge	Interaction Semantic Language and Interaction Intention
Cognitive with high level of semantics (CA-CA)	1. Internal Agent World Model	1. Compatible with other agents 2. Inconsistent with other interacting agents	Each agent may be in different COIs, same COI but with different roles, have different capabilities, have different assumptions about requirements for interaction, and may have other context assumptions constraining any interaction. For human agents, context is part of overall organizational and social model, while for technology agents, context may be implicit by design or explicit through context models for interpreting domain knowledge.	Domain Models in each agent contain overall knowledge that is used to guide behavior of agent and to interpret meaning of interactions. Its internal knowledge is typically updated through interactions with environment, including other agents.	Set of intentions supported by each agent may be somewhat different, but for those that are similarly supported then semantic interpretation of intent can be consistent within certain context. Language used for knowledge sharing interactions between cognitive agents are interpreted by each agent according to its internal knowledge model. Communications associated with interactions must have close semantic relationship to language used to represent knowledge in internal model. Language does not have to be same, but it should be semantically translatable to model.
	2. Agent Cognitive Processing	1. Consistent Exchanged Semantic Expressions 2. Inconsistent with exchanged semantic expressions			
Hybrid (CA-RA)	1. Design Agent (DA) World Model	1. DA World Model Consistent with RA Stimulus Response Implementation 2. RA Implementation inconsistent with DA World Model	Similar to CA-CA context except that RA will not have potential for explicit context models, RA designs will have implicit context by design agents.	CA has domain knowledge model and translates RA stimulus response communications to concepts contained within its knowledge model. Cognitive agent behavior is not dictated by knowledge model, but rather is used by agent for determining its actions. Conversely, RA has only simple set of definitions mapping interface stimulus messages to its response behavior model.	A cognitive agent translates stimulus/ language used by reactive agent to expressions consistent with its internal domain knowledge model, while reactive agent only understands language of the set of stimulus response messages.
	2. CA World Model	1. Consistent CA World model and cognitive processing mapping to RA Stimulus – Response interactions 2. Inconsistent CA World model and cognitive processing mapping to RA Stimulus – Response interactions.			

Cognitive Semantic Dimension Pattern Dimension	Subdimension	Dimension Values	Context	Domain Knowledge	Interaction Semantic Language and Interaction Intention
	RA Stimulus – Response Processing	1. S/R processing consistent with exchanged interactions. 2. S/R processing inconsistent with exchanged interactions.			
Reactive (RA-RA)	DA World Model	1. Explicit and consistent model defining multiple RA S/R Processing, Behaviors and Interactions	Context is completely defined implicitly by design assumptions.	Each RA can only respond to its finite set of signals and messages at interface. There is inherent knowledge other than behavior.	Interaction is limited to set of signals and messages that each RA can respond to.
		2. Inconsistent and nonexplicit model defining multiple RA S/R Processing, Behaviors and Interactions			
	RA Agent Stimulus – Response Interactions	1. Mutually Consistent Response to Stimulus messages			
		2. Inconsistent mutual response to stimulus messages			

4.1.2.6 Detailed Multi-agent Semantic Interaction Model Dimension (3.1.4.6)

A model defines a detailed description for semantic interaction based on the following concepts or subdimensions. An interaction is defined as a set of shared information exchanges with three separate domains of knowledge, e.g., information about the situation, information about shared domain knowledge, and information about the intentions of each interaction and expected response. Information about the situation is related to the mutually consistent perspective of the context among interacting agents. Agents also exchange information about domain knowledge and only those at mutual intersections in their respective knowledge models have a chance for mutually consistent interpretation. The messages comprising the interactions also have their own semantic representations and again these have to be mutually understood in a mutually consistent manner across all interacting agents.

Subdimensions

1. (CxC), (DxD), (IxI) **Explicit and Interdependent Context, Domain, and Intention Models and Interactions**
2. (CxC), (IxI) **Explicit and Interdependent Context and Intention Model, Implicit Independent Domain Model**
3. (DxD) **Implicit and Independent Context, Intention Model, Explicit and Interdependent Dependent Domain Model**

4. (DxD), (IxI) **Explicit and Interdependent Domain and Intention Models, Implicit Context**
5. **Implicit and Independent Context, Domain, and Intention Models**

Subdimension Elements

1. **Agent Context Model**—defines the relevant knowledge required for a specific situation and purpose from an interacting network entity's perspective and also defines the characteristics of the context itself.
2. **Agent Domain Knowledge**—defines a finite set of concepts, relationships, and properties within a domain of discourse, usually including the grammar, domain dependent and independent vocabulary, real world referents for domain vocabulary, rules of inference, and a unifying semantic model.
3. **Agent Intentional Model**—defines the intentional nature of the interaction and the conditions of satisfaction, where the conditions of satisfaction determine whether the interaction is satisfied by mapping knowledge to the described domain, or whether the interaction is satisfied by actions taken by the entities, etc.
4. **Situation Information Exchange Interactions**—the set of messages and dynamic processes across the interacting agents and related to the agent context models.
5. **Shared Knowledge Information Exchanges**—the set of messages related to the agent domain knowledge models.
6. **Shared Information about Intentions**—the set of messages related to the agent purpose for sending the message, also related to the agent cognitive processing.

It is expected that networked entities may have multiple types of semantic interactions where the domain of knowledge associated with the interaction, the particular situational context constraining the semantic interpretation, and the type of interaction vary (Figure 4-13). By this we mean that the elements of a semantic interaction, e.g., particular context, domain knowledge, and interaction type must be compatible for a successful semantic interaction between interacting entities. Equation 1 defines a successful semantic interaction between two interacting entities, A and B.

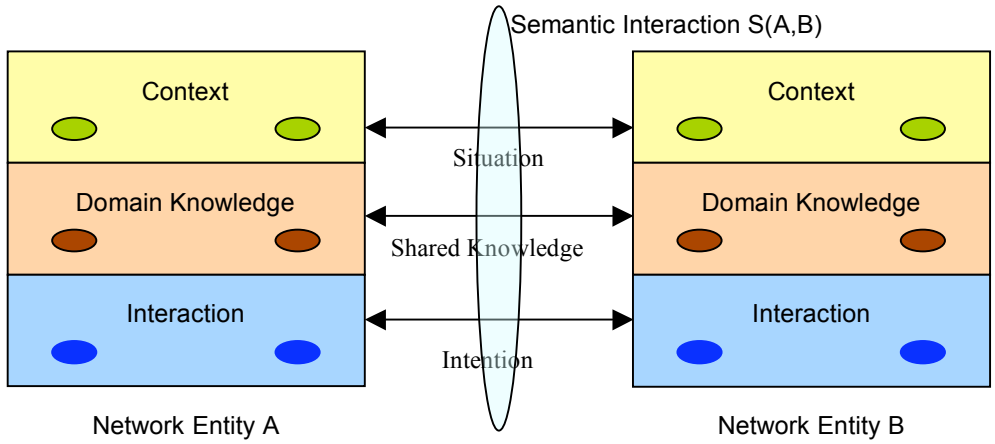


Figure 4-13. Semantic Interaction between Networked Entities

Eq 1. $S(A,B) = \{(ca,cb), (da,db), (ia,ib) \mid ca \cong cb, da \cong db, ia \cong ib, (ca, da, ia) \in A, (cb, db, ib) \in B$

This equation states that a semantic interaction between A and B can only have its conditions satisfied if it is comprised of compatible Context, Domain, and Interaction types by each networked entity involved in the interaction; in this case A and B entities. The context of A in the interactions must be compatible to the context of B in the interaction, the domain knowledge must be compatible, and the interaction type must be compatible.

If each interacting entity has a different semantic model for a particular semantic interaction, e.g., one of the semantic interaction elements is not compatible, then the interacting entities have little chance for achieving a mutually consistent understanding of shared data. In some cases, their semantic interaction models may overlap, providing an appearance of common understanding. This situation potentially has greater opportunity for confusion due to the possibility of reaching entirely different conclusions about the state of the world as a result of subtle differences in meaning about apparently similar concepts. To reach a shared and consistent understanding of concepts, each interacting entity would have to know the purpose of the shared information, the domain of discourse, have similar understanding of concepts in a domain of discourse, have similar models of what is true in a domain, have common rules of grammar, have common representations of information, and have the means to communicate all of the above either explicitly, or through implicit assumptions by design and implementation.

Domain Knowledge

Although the act of communications is the means by which entities share knowledge, it is usually the case that the entities refer to common definitions of a domain of knowledge when communicating to enable more concise and less verbose communications. With predefined meta definitions and models, entities need only communicate information using the same model to ensure consistent understanding. There are different levels of semantic expressivity in language and models and this bears on the level of semantic consistency attainable.

The shared knowledge comprising concepts within a domain of discourse between communicating entities will be expressed in some syntactical and semantic form, comprising rules of grammar, concepts and a lexicon for that domain, and a model of the possible statements or expressions using these concepts in that domain. The dimension “Semantic Expressiveness” will identify different levels of implicit and explicit semantic representation that can exist between communicating entities, whether they are human or machine representations.

Intention

There is usually an intentional aspect associated with all interactions, e.g., share knowledge about the world, coordinate collaborative activities, express the importance or relevancy of the communications, declare a state of affairs exists. The intentional component of the communications informs the responder on how to interpret the content of the communications act, and in multi-agent systems this is usually represented in the form

< intention> (<content>)

”Intention” is a key word identifying the type of communication act, providing a clue to how to interpret the content and the conditions of satisfaction for the communications act. Therefore a

successful semantic communications act will entail understanding of the conditions of satisfaction of the intentional aspect of communications, and a shared understanding of the semantics of the content

Some major classifications of interaction intentions associated with the communication interaction are shown in Table 4-4.

Table 4-4. Intention Speech Act Classifications

Communicative Intent Speech Act Type	Description
Assertive speech act	Agent shares knowledge about the world through informing act
Directive act	Agent requests some action by another agent
Commit speech act	Agent commits to perform some action
Expressive speech act	Agent expresses some internal state, e.g., thanking. Could be
Confirming	Agent confirms or denies a received proposition form another agent
Declaration	Effects some change in state of affairs, e.g., agent A is now offline

With respect to the semantic interaction, this classification of the different interaction intentions provides another context for interpreting the content exchanged in the interaction and its intended effect.

Context

It is understood that it is possible for interacting entities to have different perspectives on their relevant domain knowledge for a variety of reasons. This may lead to situations where the semantics of communication are not entirely successful, e.g., the concepts shared in communications may be similar but defined from an entirely different community of interest context. A UAV may have its operating instructions shared by a training organization, while the same platform type may have its maintenance instructions defined and managed by a different organization. Though there is common knowledge between these communities, much of the knowledge that could be shared is not directly related to the purpose of the other community. Each community would have to understand the knowledge from the other community's perspective and its set of definitions.

Semantic Interoperability Concept

We defined previously as satisfaction criteria for a semantic interaction between interacting networked entities, a particular set of compatible context, domain knowledge and interaction types. Different interacting entities at any moment will share domain knowledge relevant to their context and particular intention within a semantic interaction. It is important to understand the role of context and domain in ensuring that the knowledge being exchanged is relevant to each interacting networked entity. The context defines the knowledge necessary for a situation and entity's purpose, while the domain knowledge defines the concepts and semantic model within a universe of discourse. Thus context is a metalevel model that refers to relevant domain knowledge from its perspective. Interaction types further clarify the intent of the interaction, whether to share knowledge, request an action, seek proposals for collaboration, access a service, express a belief, etc. In multi-agent systems, such as Foundation for Intelligent Physical Agents (FIPA), these communicative intentional speech acts have been defined with semantic specifications clarifying their intent and how the intention and content is to be interpreted.

A context definition may require knowledge from more than one domain, and also that domain knowledge definitions may be used by more than one context definition. Over time, it is reasonable to

expect that additional contexts will be defined for an existing set of domain knowledge between interacting entities as their purposes and capabilities evolve, and that the domain knowledge will also expand to support the increasing capabilities of interacting networked entities. A semantic interaction according to this model as defined in Figure 4-14 has a semantic definition for a context, domain, and interaction type. Different intention interactions may exchange the same domain knowledge expression, e.g., assert the domain expression to be true, assert the domain expression to be false, assert the domain expression have high relevance to a context, etc. The actual semantic interpretation of any domain or context expressions will be modified by the type of intentional interaction type as suggested.

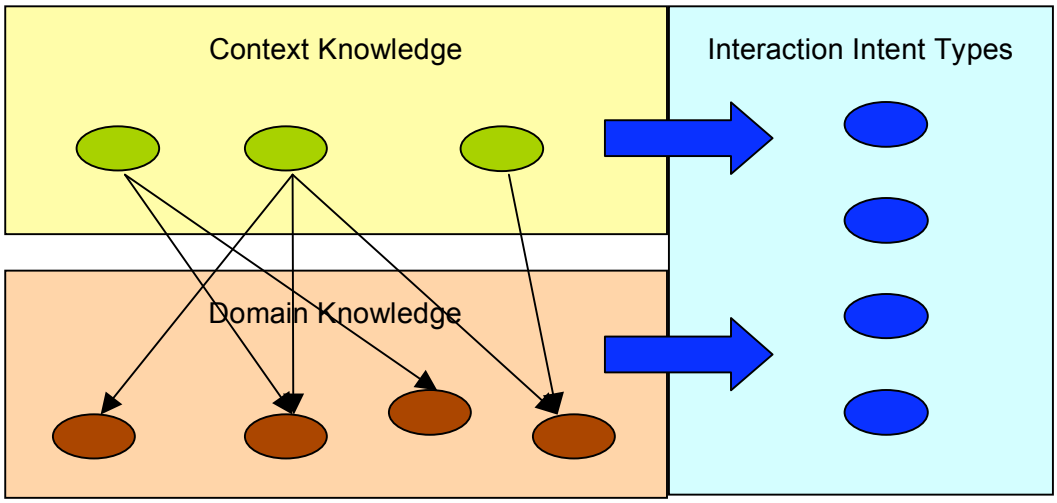


Figure 4-14. Context and Domain Knowledge Mappings and Use of Interaction Intent Types

Domain knowledge defines specific concepts, relations, grammar, vocabulary, rules of inference, and a unifying model for a particular universe of discourse. Though the domain knowledge is usually defined to support one or more purposes by participants in the discourse, it is sometimes necessary that in order to achieve a particular purpose, an agent must have other knowledge not necessarily within its purview and must then collaborate with other agents to discover and acquire this knowledge. In addition other context aspects such as different environmental situations, changing business rules for an organization, changing organizational structures, different missions, and many other aspects may be relevant to defining what knowledge is appropriate for a particular semantic interaction; this level of variability can be captured by a higher metalevel definitions of context that refers to relevant domain knowledge and interprets it within a specific contextual perspective and purpose. For example a domain knowledge definition might define the characteristics of specific types of insurance services and contracts, while different context models may refer to relevant knowledge in this domain from the perspective of an insurance customer, an insurance industry regulator, and an insurance service provider. In each of these cases the context may also refer to other knowledge domains, e.g., the insurance regulator context may also refer to state statutes regulating the insurance industry in that state.

Interaction Semantic Consistency Definition

With respect to Semantic Interoperability, it is the compatibility and mutually consistent logical deductions made by each interacting entity on the meaning of the exchanged knowledge that determines the level of congruence. If each interacting entity is using compatible context and domain

knowledge and also using compatible intention models, then the semantics of the interaction can be assured, but if the combined context and domain knowledge and intention semantic models are not mutually consistent, then the semantic interactions and subsequent deductions by the interacting entities on this shared knowledge will not be consistent. The intention models used here define the type of semantic interaction according to speech acts, and as such do not modify or constrain the domain and context models, but rather are means to index to the appropriate models.

With respect to domain knowledge models it is somewhat easy to determine the possibilities of Semantic Interoperability based on the set theoretic relationships between the concepts of the relevant domain knowledge models referenced by each interacting entity. In contrast where each interacting entity has a defined context identifying the knowledge appropriate for its purpose and a particular situation, semantic compatibility still follows the definition that each interacting entity should not make mutually inconsistent deductions regarding its shared domain knowledge, but may make different deductions within a context model relative to its purpose based on the relevant domain knowledge. By this we mean that it is possible for interacting entities to make different deductions within its context model, but still require consistency within the domain knowledge. In the case where an entity's context depends on the context deductions of other entities, then we have a situation where the joint context knowledge is used to create common knowledge that is consistent.

4.1.2.6.1.1 Detailed Multi-agent Semantic Interaction Model Values (3.1.4.6.1.1)

Since the set of context and domain and intention knowledge models known and understood by an interacting network entity defines in total the knowledge it uses for its purpose and role, we can define the following possible general semantic compatibility or congruence patterns. The semantic interoperability patterns will be described by four set theoretic relationships, e.g., disjoint, overlapping, subset, and equivalent. With each of four possibilities for context, domain, and intention models, we have a total of 4 by 4 by 4 = 64 possible individual patterns.

Of this large set, the following interdependent semantic interaction congruence patterns should be identified as to their possibility in the set of interactions being evaluated across agents. They are essentially defined about whether the context, domain, and intentions are explicitly defined, have some set theoretic consistent intersections. Each pattern is labeled by type as defined by the semantic dependency between the context, domain and intention knowledge of the interacting entities. Of course when there are more than two interacting entities, there may be different congruence patterns for each pair for a specific semantic interaction, possibly resulting in incompatible semantic interoperation and inconsistent common knowledge, potentially resulting in errors by applications. Joint consistent knowledge is one of the reasons for networking and incompatibilities of Semantic Interoperability may cause problems depending on the need for consistent knowledge across the interacting entities. Refer to section **Error! Reference source not found.** for the set theoretic mutually consistent definitions within the specific subdimension of each pattern.

The patterns of interaction are coded as sets of context, domain, and intention and defined in the Table 4-5 along with their possible values.

Table 4-5. Semantic Interoperability Patterns for Interacting Network Entities

Multi-agent Interoperability Semantic Interaction Pattern Subdimension	Context Model and Information Exchange	Domain Model and Information Exchange	Intention Model and Information Exchange	Subdimension Values	Semantic Consistency
(CxC), (DxD), (IxI) Interdependent Models and Semantic Information Intersections	√	√	√	1. equ 2. sub 3. ove 4. dis	Semantic consistency depends entirely on deductions made both by context, domain and intention knowledge models used by each interacting entity. All context and domain models have to be compatible to ensure consistency. Intention model is used to partition exchanges used. Context and domain models each could have relationships to each other of overlap, subset, and equivalents.
(CxC), (IxI) Interdependent Context and Intention Model, Independent Domain Model	√	X	√	1. equ 2. sub 3. ove 4. dis	Depends on consistent context and intention models with either overlap, subset or equivalence relationships, but does not depend on consistency of shared domain knowledge model. Possible set or interactions that only focus on collaboration of activities, without sharing situational or domain knowledge, e.g., workflow protocols and metadata.
(DxD) Independent Context, Intention Model, Interdependent Domain Model	X	√	X	1. equ 2. sub 3. ove 4. dis	Only depends on consistency relationships between domain models and any of relationship of subset, overlap and equivalence may hold. Cntext models are typically disjoint. Might exist in closed networks where context definitions are implicit and set of interacting entities are bound at implementation and/or design time. Another example occurs when interacting entities have entirely different contexts, e.g., insurance regulator and insurance customer in case of context, where there are deductions on some common knowledge but their interactions are not reasonable due to different purposes and uses of domain knowledge, e.g., additional deductions may be inferred that are not contained in each other's context model.
(DxD), (IxI) Interdependent Domain and Intention Models, Implied Context	X	√	√	1. equ 2. sub 3. ove 4. dis	Pattern where domain knowledge is shared and information about intention of agent information message content is identified, e.g., pointers to specific domain ontologies for example.
Independent Context Model, Independent Domain Model, Independent Intention Model	X	X	X	1. dis	Not semantically interoperable, e.g., context and domain models are each disjoint and thus there can be no semantic interaction between networked entities.
Note: X = don't care √ ≡ dependent					

4.1.2.6.1.2 Semantic Interaction Pattern Relationship—Dimension Values (3.1.4.6.1.2)

The following semantic dependency relationships are further defined in the following, and are used as described previously to relate the context models and the domain models of interacting entities with each other. The relationship between a specific context and set of domain model are self-contained within the context models themselves, while the compatibility of the context and domain models used by each agent have the following relations. If there are classifications of types of interactions according to intent, then we need to also identify the relations between the intention classifications using the definitions as follows;

IE = { } set of interacting networked entities

C = {c0, c1, c2, …, ci} set of context models

D = {d0, d1, d2, …, dk} set of domain models

I = {i0, i1, i2, …, il} set of intention models

R = { dis, ove, sub, equ} set of pattern relationship types where

Disjoint relation

dis ≡ {Ai | $\bigcap_{i \in I} A_i = \varnothing.$ } where Ai is either of type C, D, or I

Overlap or Almost Disjoint

ove ≡ {Ai| $\bigcap_{i \in I} A_i < \infty.$ } where Ai is either of type C, D, or I

There are some elements common between the models

Subset

sub ≡ {A1 ⊆ A2 |∀x ∈A1, x ∈ A2}, where Ai ∈ C ∪ Ai ∈ D

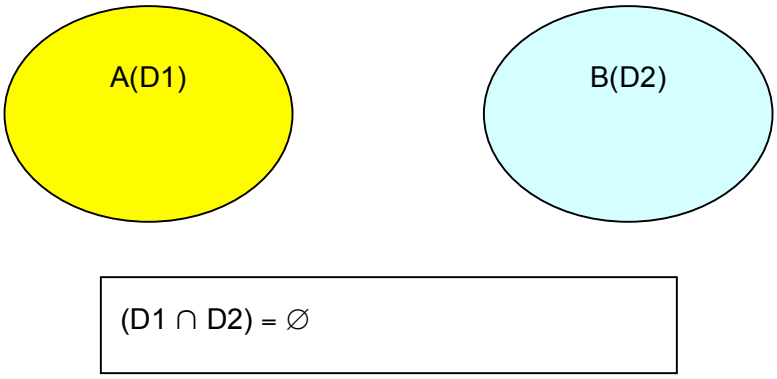
Equivalence

equ ≡ {A1 = A2 | A1 ⊆ A2 ∩ A2 ⊆ A1, |∀x ∈A1, x ∈ A2, ∀x ∈A2, x ∈ A1, where Ai ∈ C ∪ Ai ∈ D

Disjoint: The knowledge understood by the interacting entities are entirely separate and unique (Figure 4-15). The concepts understood by entity A are defined in domain D1, and are not understood by communicating entity B, and likewise the concepts of communicating entity B, D2, are not understood by communicating entity A. They are totally disjoint from each other. Each communicating entity, in order to understand what was given to it by the other entity would have to understand the other entities domain knowledge and model. This would require that A(D1, D2) and B(D1, D2), each understanding its own domain and the other entity's domain. Transformation is not possible here due to the differences in domain knowledge.

Figure 4-15. Disjoint Semantic Relation Pattern

Overlapping Domains: The domains of knowledge understood by the communicating entities are partially separate and partially common, in some parlance called partially disjoint (Figure 4-16). What



this implies is that there are some concepts that are mutually understood and other concepts that are not mutually understood.

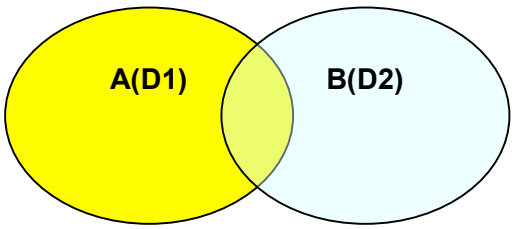


Figure4-16. Overlap Semantic Relation Pattern

Clearly in this situation partial shared and consistent understanding within a domain is possible. For example if Entity A has knowledge of a vehicle's maintenance concepts in its domain D1, while Entity B has knowledge of the same vehicle's operation instructions, and if both Entities have shared knowledge of the some of the constituent parts of the vehicle they could share and understand knowledge about the constituent parts but not understand each others knowledge about maintenance and operation respectively. If, instead, both entities understood maintenance concepts, but had knowledge about maintenance of different vehicles, then some of their

[(D1 ∩ D2) ⊂ (D2)] ∩ [(D1 ∩ D2) ⊂ (D1)]

concepts may have similar understanding, e.g., maintenance state, but other concepts would be different, e.g., “engine” versus “door.”

Subset Domains: In this case, the domains of knowledge understood by one of the communicating entities is a subset of the domain of another entity (Figure 4-17). In the model below all of the knowledge of entity B, D2, is understood by entity A, but only some of the knowledge understood by entity A is understood by entity B. We say that D2 is a subset of D1. In this case, we have asymmetric semantic understanding. Another perspective is to consider that the superset has additional knowledge supporting more detail than the subset. In this case, we might expect that D1 is a refinement of D2, which is more general.

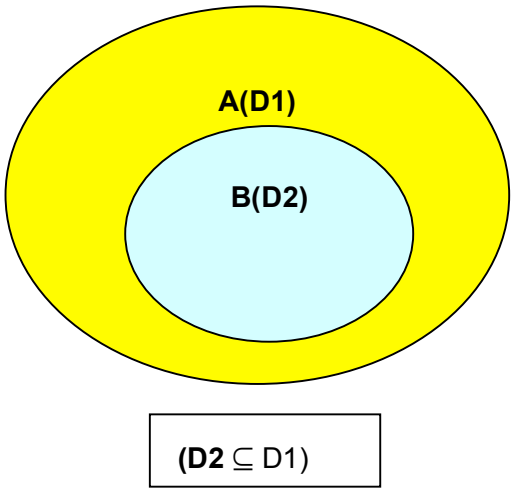


Figure 4-17. Subset Semantic Relation Pattern

Equivalent Domains: Domain knowledge shared between communicating entities is exactly equivalent, every concept in domain D1 is covered by a similar concept in domain D2, and vice versa (Figure 4-18).

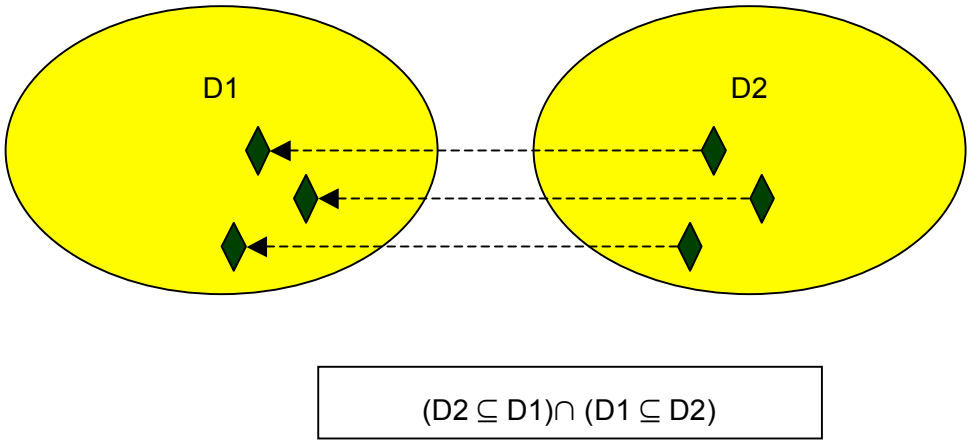


Figure 4-18. Equivalent Semantic Relation Pattern

In this case, there is full semantic consistency of shared understanding either due to the fact that the concepts in the domains each entity understands are exactly equivalent, or have been made equivalent through some form of transformation. In the former D1 = D2, while in the latter D1 ≅ D2. The only possibility left for misunderstanding is that context of the use of the knowledge in each domain by each communicating entity. Understanding the context of the other entity regarding its perspective on the domain of knowledge and its purpose for sharing, can influence the certain aspects of the semantic interpretation by each entity, e.g., the granularity of the knowledge, the timeliness, the accuracy, and the relevant facts of the domain for the purpose and role of the entity. For example each communicating entity may share the same domain of knowledge, but only require certain knowledge out of this domain for its purpose.

Semantic Interoperability Relations Application

These concepts can be used to identify their Semantic Interoperability compatibility relations between the set of context and domain knowledge models or specifications for each semantic interaction type and the interaction type definitions, whether human or technology. The Semantic Interoperability values are ordered with increasing consistency of semantic interpretation between the interacting entities, e.g., (dis, ove, sub, equ) where eq represents the highest level of semantic compatibility and dis represents the lowest level of compatibility (Figure 4-19).

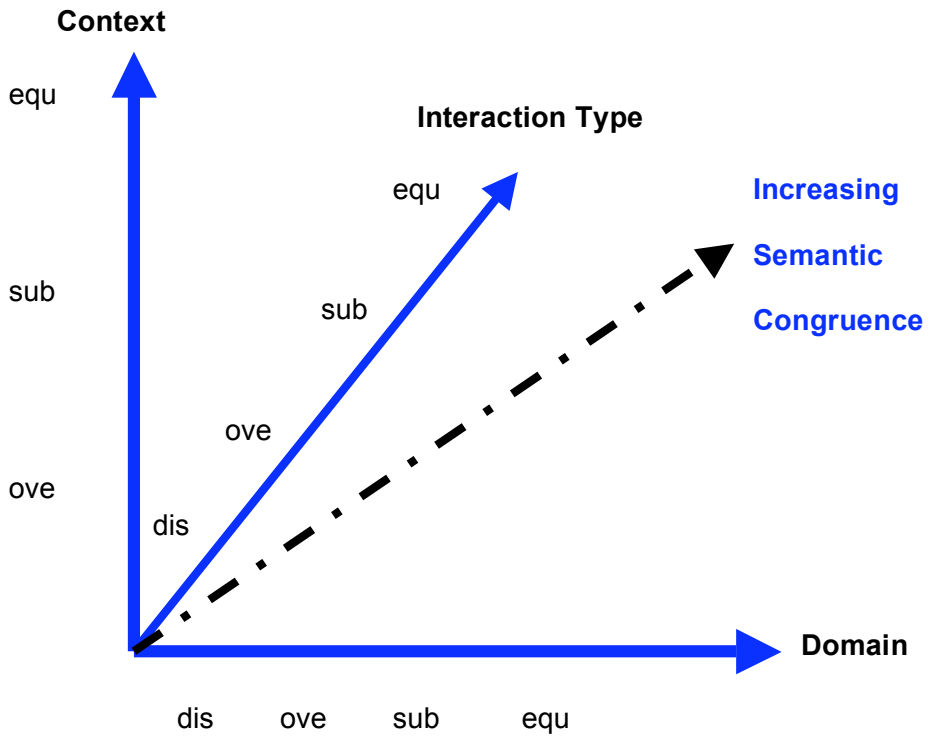


Figure 4-19. Semantic Interoperability of Context and Domain Interactions

4.1.2.7 Semantic Expressiveness Dimension (3.1.4.7)

Semantic interoperability involves interactions where it is desired to have "mutual and shared common understanding" reached by the collaborating agents where decisions or actions are taken based on the mutual interpretation of communications and shared information. Significant research has been

conducted over the past decades, to gain a better understanding of how humans communicate, use language, create knowledge about the world, classify concepts, and use pragmatics to affect an intended result using communicative speech acts. This work has led to a realization that semantic definitions can be integrated into networked technology designs and implementations. The underlying concepts for NCO have recognized these criteria for consistency of interpretation of the meaning of shared data (Semantic Interoperability) and communications in networked interactions, while the W3C Semantic Web initiative has evolved standards and languages to integrate semantic definitions (metadata) in web services and web page content.

Fundamental to all of this is the requirement to reach a common understanding of the meaning of the information represented in some language and domain of discourse; now referred to as knowledge representation and practiced in knowledge engineering[16]. In this section we are concerned with characterizing the expressiveness of the semantic representation of the interaction and models used for interpreting exchanged information, rather than comparing the domain of concepts that each agent can understand, as previously discussed.

Language Standards for Domain Semantic Knowledge

In the past WWW designs and implementations, the previously defined elements for a semantic interaction, {context, domain, intent} have not been defined in computer interpretable languages; while in the emerging Semantic Web, the domain knowledge element is being semanticized through the use of ontology languages such as RDF and OWL. Neither context nor interaction intent languages have yet been standardized by the WWW.

Language Standards for Intent Communications Semantic Knowledge

Yet the IEEE Computer Society has adopted the FIPA[17] set of standards, which includes the semantics for Agent Communication Language (ACL), the equivalent for defining the semantics for the intentional interaction type.

Language Standards for Context Semantic Knowledge

There are no current context standards, but there are efforts to define context theories and knowledge models that will enable domain knowledge mediation and logical deductions across multiple domain models. Examples include the OpenURL[18] effort for Context Sensitive Services, the W3C Web Content Accessibility Guidelines 1.0, [19] where users may be operating in different contexts, e.g.,

"*For those unfamiliar with accessibility issues pertaining to Web page design, consider that many users may be operating in contexts very different from your own:*

- *They may not be able to see, hear, move, or may not be able to process some types of information easily or at all.*
- *They may have difficulty reading or comprehending text.*
- *They may not have or be able to use a keyboard or mouse.*

[16] "Knowledge representation – Logical, Philosophical, and Computational Foundations", by John F. Sowa, published by Brooks/Cole, cataloged as ISBN 0 534-94965-7
[17] http://www.fipa.org/
[18] http://www.niso.org/committees/committee_ax.html
[19] http://www.w3.org/TR/WAI-WEBCONTENT/#gl-complex-elements

- *They may have a text-only screen, a small screen, or a slow Internet connection.*
- *They may not speak or understand fluently the language in which the document is written.*
- *They may be in a situation where their eyes, ears, or hands are busy or interfered with (e.g., driving to work, working in a loud environment, etc.).*
- *They may have an early version of a browser, a different browser entirely, a voice browser, or a different operating system*. "

This certainly defines context from a user perspective, and web design should take this context into account. To enable consistent use of semantic web data this context metadata and knowledge must be shared to enable context sensitive web accessibility.

Other examples of context knowledge are used in highway planning in California as evidenced by their Director's Policy[20] "*Context sensitive solutions meet transportation goals in harmony with community goals and natural environments. They require careful, imaginative, and early planning, and continuous community involvement.*" So context is also used by organizations to provide flexibility to adapt to different situations.

Limitations of Simple XML Metadata Efforts

The problem of defining consistent labels for exchanging data elements was solved by the XML[21] standard, and it was hoped that this would enable more open networks able to share data elements with each other, e.g., invoices, purchase orders, patient records, etc.

Commercial organizations, enterprises, and consortiums were formed to predefine XML data schemas with semantic definitions agreed to by the consortium members. The approach revolved around sharing metadata elements using XML metadata syntax. It was quickly discovered that XML was not sufficient, since the semantic definitions for XML data elements were still in document form, and not explicitly represented in the system or metadata, nor shared between systems, resulting in opportunities for error in semantic interpretations by different organizations. Though having a shared common syntax for data elements based on XML did help to ensure common labels for data elements and attributes, it did not constrain the networking elements as to the shared meaning within a model for the data elements. Even schemas with defined taxonomies and structural conceptual models were not sufficient, though aiding the design engineers, it did not provide a means for sharing explicit semantic representations between systems.

4.1.2.7.1 Semantic Expressiveness and Compatibility Dimensions and Values (3.1.4.7.1)

Each analysis step characterizes an aspect of semantic expressiveness according to a definition of a semantic expressiveness subdimension and its value set (Figure 4-20). Table 4-6 can be used for capturing the information generated in this process.

[20] http://www.dot.ca.gov/hq/oppd/context-solution.pdf
[21] http://www.w3.org/XML/

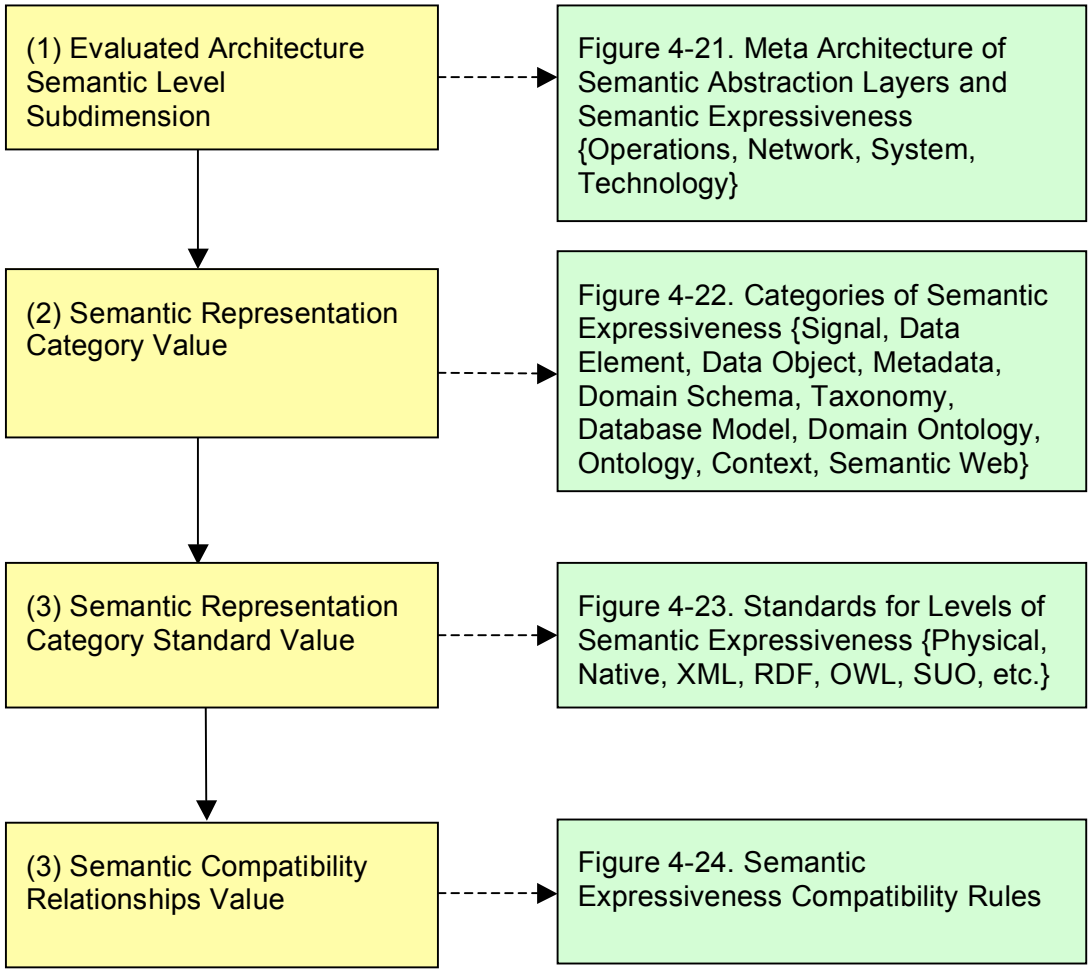


Figure 4-20. Semantic Expressiveness Analysis Process

Table 4-6. Semantic Expressiveness Subdimensions and Values

Evaluated Architecture Semantic Level Subdimension	Categories or Levels of Semantic Expressiveness	Semantic Standard or Language (Note there will be different possible standards for each architectural level and for different levels of semantic expressiveness)	Semantic Expressiveness Compatibility Rules	
Operations	1. Semantic Web 2. Context 3. Upper Ontology 4. Domain Ontology 5. Database Model 6. Taxonomy 7. Domain Schema 8. Syntactical Metadata 9.Data Objects 10. Data elements 11. Raw Signals	(List involved standards)	1. sub 2. equ 3. dis	1. Explicit ⊇ Explicit 2. Explicit ⊇ Implicit 3. Implicit ⊇ Implicit
Network	1. Semantic Web 2. Context 3. Upper Ontology 4. Domain Ontology 5. Database Model 6. Taxonomy 7. Domain Schema 8. Syntactical Metadata 9.Data Objects 10. Data elements 11. Raw Signals	(List involved standards)	1. sub 2. equ 3. dis	1. Explicit ⊇ Explicit 2. Explicit ⊇ Implicit 3. Implicit ⊇ Implicit
System	1. Semantic Web 2. Context 3. Upper Ontology 4. Domain Ontology 5. Database Model 6. Taxonomy 7. Domain Schema 8. Syntactical Metadata 9.Data Objects 10. Data elements 11. Raw Signals	(List involved standards)	1. sub 2. equ 3. dis	1. Explicit ⊇ Explicit 2. Explicit ⊇ Implicit 3. Implicit ⊇ Implicit
Technology	1. Semantic Web 2. Context 3. Upper Ontology 4. Domain Ontology 5. Database Model 6. Taxonomy 7. Domain Schema 8. Syntactical Metadata 9.Data Objects 10. Data elements 11. Raw Signals	(List involved standards)	1. sub 2. equ 3. dis	1. Explicit ⊇ Explicit 2. Explicit ⊇ Implicit 3. Implicit ⊇ Implicit

4.1.2.7.1.1 Knowledge Representation for Meta Network Architecture Layers Dimension (3.1.4.7.1.1)

In technology mediated semantics, especially networks, there are layers of semantic abstractions each layer having its own semantic definitions appropriate to its purpose.

In this meta architecture model for semantic domains of knowledge for networking solutions, there are historical forms of representation that are used to specify, model, design, and describe solutions at each

layer (Figure 4-21). There are different model representations that attempt to provide an overall integration of all of these layers and the functional dependencies between the elements across these layers, e.g., UML, and DoDAF, while other model representations focus on a particular problem area, e.g., performance queuing models, reliability models, database models, metadata models, semantic web OWL ontology models, etc. Currently there are no integrated models that provide the capability to link and associate all of the semantic metadata dependencies. This has led to problems for ensuring semantic consistency with models.

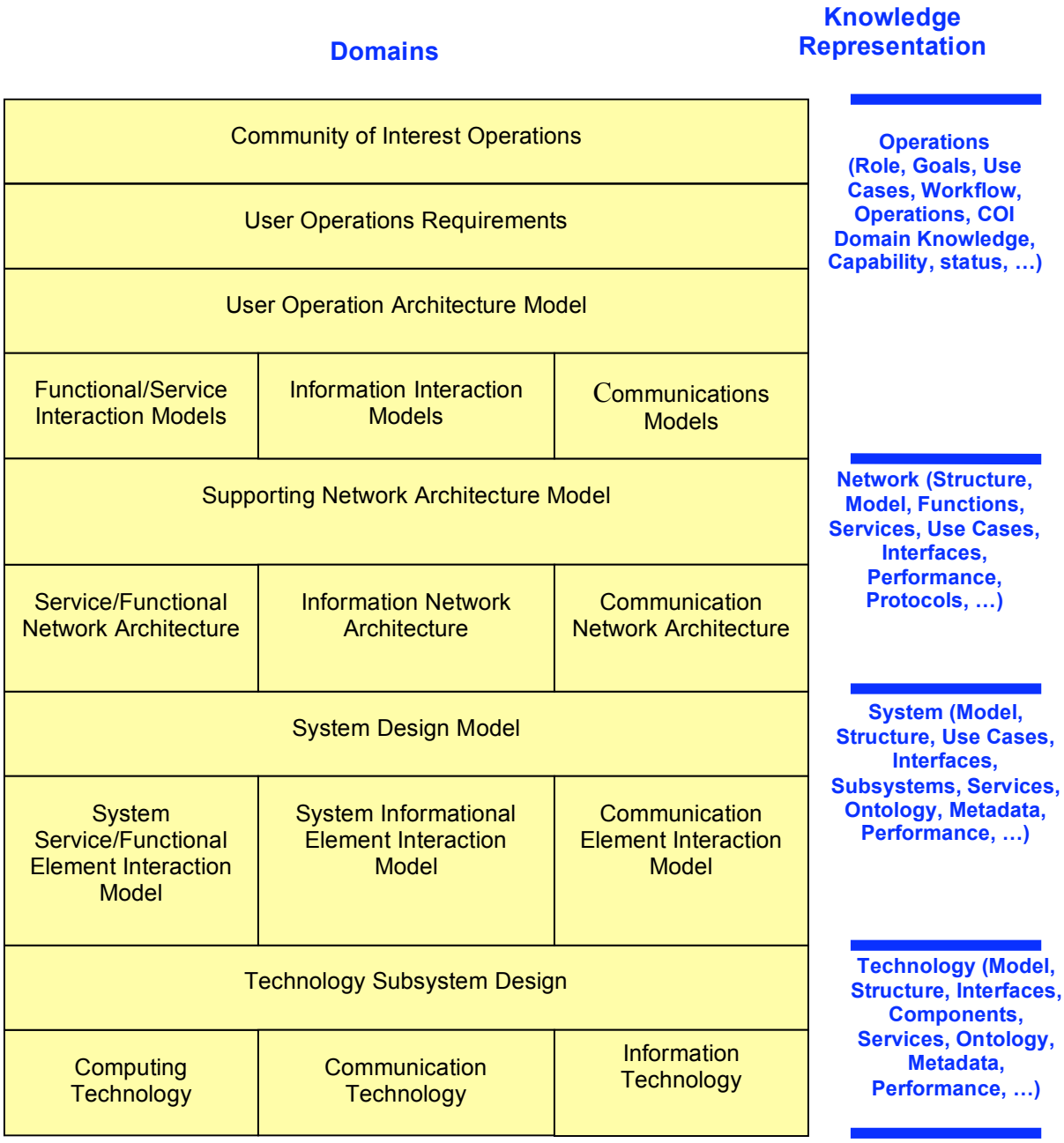


Figure 4-21. Meta Architecture of Semantic Abstraction Layers and Semantic Expressiveness

Typically, the semantics associated with a particular form of representation for a particular domain within a layer is well understood in an engineering community, e.g., protocol specifications and use of

finite automata (State machines) to represent the context dependent interaction sequence between peers. The form of representation for the design problem may be quite specialized, e.g., M/M/s network queuing models for performance characterization under exponential arrival offered traffic situations; thus reducing the size of the community able to understand the representation. For example the above protocol interaction model could be defined at a higher level of services description that the set of peer to peer interactions support, e.g., "sequenced packet data", etc.

Using a common language and semantic model for each layer of the multilevel meta architecture enables other agents to use this knowledge when creating or understanding dependencies between elements.

4.1.2.7.1.2 Characterizing the Meta Architecture—Dimension Values (3.1.4.7.1.2)

For purposes of characterization it is sufficient to identify the meta architecture layer of interest, the aspects that are characterized, and the general form of description.

Major Domain Category: Operations, Network, System, Technology

The following defines some of the types of semantic representations that can be used to define and describe the semantics of the meta architecture networking model.

4.1.2.7.1.3 Level of Semantic Representation Values (3.1.4.7.1.3)

A conceptual model for identifying different levels of semantic expressiveness is defined with higher levels representing more semantically expressive computer interpretable models and definitions, while lower levels have less expressivity to represent the knowledge available for computer interpretation and may only exist in documents and specifications. Each level of semantic expressiveness may have different languages and standards with different syntactical and semantically expressivity, though it would be better to utilize one standard (Figure 4-22).

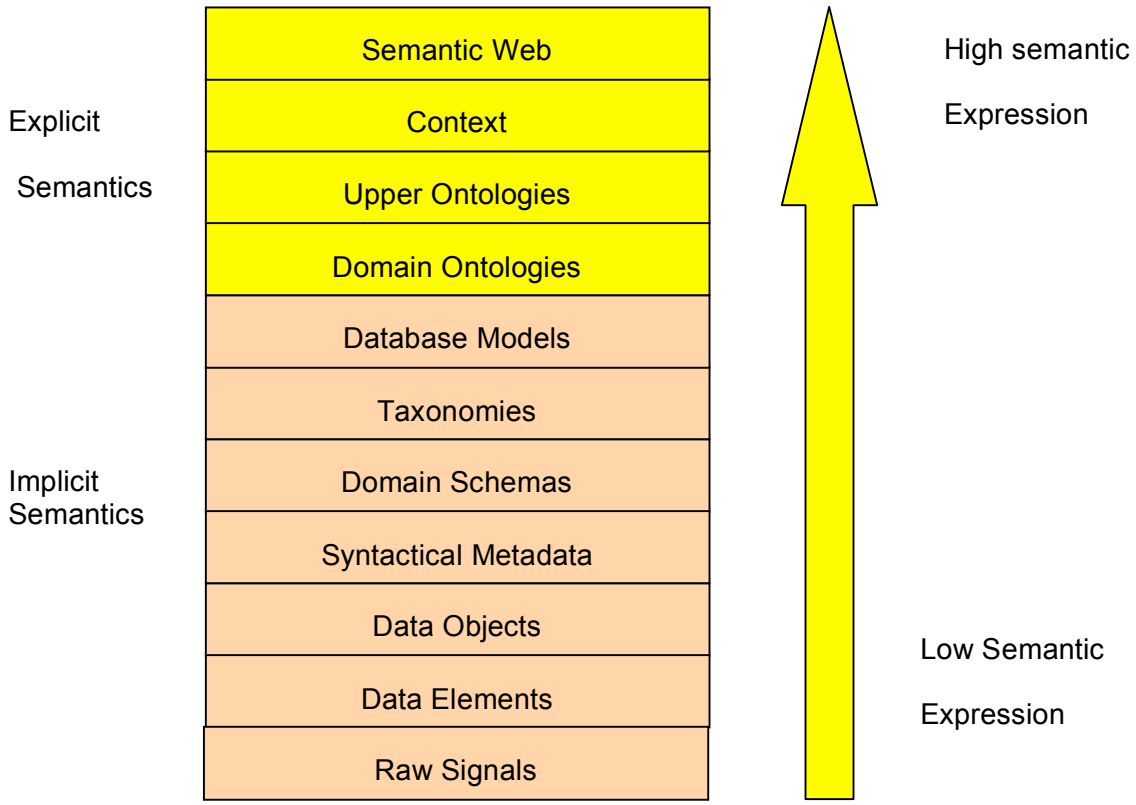


Figure 4-22. Categories of Semantic Expressiveness

Each level's semantic expressiveness capability is described in Table 4-7

.Table 4-7. Definitions of Levels of Semantic Expressiveness

Semantic Level	Description
Semantic Web	**Semantic Web** is a project that intends to create universal medium for information exchange by giving meaning (semantics), in a manner understandable by machines, to content of documents on Web. Currently under direction of Web's creator, Tim Berners-Lee of the W3C, Semantic Web extends WWW through use of standards, markup languages, and related processing tools. Semantic Web comprises standards and tools of XML, XML Schema, RDF, RDF Schema, and OWL. OWL Web Ontology Language Overview describes function and relationship of each of these components of Semantic Web: • XML provides surface syntax for structured documents, but imposes no semantic constraints on meaning of these documents. • XML Schema is language for restricting structure of XML documents. • RDF is simple data model for referring to objects ("resources") and how they are related. RDF-based model can be represented in XML syntax. • RDF Schema is vocabulary for describing properties and classes of RDF resources, with semantics for generalization-hierarchies of such properties and classes. • OWL adds more vocabulary for describing properties and classes: among others, relations between classes (e.g., disjointness), cardinality (e.g., "exactly one"), equality, richer typing of properties, and characteristics of properties (e.g., symmetry), and enumerated classes.

Semantic Level	Description
Context	Context in knowledge representation sense provides refined, perspective of relevant domain knowledge for specific context situation and purpose. Other definitions of context [22] considered here include *"In communications and linguistics, **context** is the meaning of a message (such as a sentence), its relationship to other parts of the message (such as a book), the environment in which the communication occurred, and any perceptions which may be associated with the communication. In other words, context is a "frame" through which we view a message.* *In computer science, **context** is the circumstances under which a device is being used, e.g. the current occupation of the user. (see also context awareness, context switch).* *In Artificial Intelligence, **context** is very much related to it's properties in communications, linguistics and philosophy. Research is being performed about how these aspects can be modeled in computer systems (e.g. logic-based) for use in automated reasoning."*
Upper Ontology	The IEEE's web site for Standard Upper Ontology[23], (SUO) provides the following definition of an upper ontology. "*An upper ontology is limited to concepts that are meta, generic, abstract and philosophical, and therefore are general enough to address (at a high level) a broad range of domain areas. Concepts specific to given domains will not be included; however, this standard will provide a structure and a set of general concepts on which domain ontologies (e.g. medical, financial, engineering, etc.) could be constructed.*"
Domain Ontology	A computer representation of concepts, their properties, their relationships to each other, any logical constraints on set membership, domain schema vocabulary, and integrated model comprising above with rules of inference for a specific domain of knowledge. Typically, level of granularity, scope of concepts, and set of facts that can be stated or inferred are guided by purpose of domain ontology. Multiple ontologies may be created for same domain or a subset of a domain for a variety of reasons, resulting in possible semantic mediation problems when sharing facts from somewhat equivalent domain ontologies. RDF/OWL languages is W3C standard for expressing ontology and facts within knowledge base consistent with domain ontology.
Database Model	*Wikipedia[24] reference: "The central concept of a database is that of a collection of records, or pieces of knowledge. Typically, for a given database, there is a structural description of the type of facts held in that database: this description is known as a **schema**. The schema describes the objects that are represented in the database, and the relationships among them. There are a number of different ways of organizing a schema, that is, of modeling the database structure: these are known as database models (or data models). The model in most common use today is the relational model, which in layman's terms represents all information in the form of multiple related tables each consisting of rows and columns (the true definition uses mathematical terminology). This model represents relationships by the use of values common to more than one table. Other models such as the hierarchical model and the network model use a more explicit representation of relationships."*
Taxonomy	A hierarchical structure of classification of things or concepts. A schema that is structured in ahierarchical fashion now becomes taxonomy.
Domain Metadata Schemas	Typically defines vocabulary and syntactical structure for each data element representing set of concepts within domain of interest and narrative description for semantic definitions of each vocabulary element within domain. An excellent example is XML Schema[25], "*An **XML schema** is a description of a type of XML document, typically expressed in terms of constraints on the structure and content of documents of that type, above and beyond the basic syntax constraints imposed by XML itself. An XML schema provides a view of the document type at a relatively high level of abstraction.*"
Syntactical metadata	Typically metadata describing some data structure. XML metadata is an excellent example of this form of representation, where data elements are bracketed [**Temp** <*temp value*> /**Temp**]. When presented with message or document or data record with data values bracketed by metadata labels, it facilitates reference to semantic meaning of *data value* within some narrative in specification. In COIs where significant effort was expended to reach agreement on XML metadata syntactical data structure and semantic definition in narrative, hope was generated to facilitate consistent semantic interpretation of exchanged data values. It is easily seen that semantic interpretation of meaning of data values is contingent on reaching agreement, and in interpreting semantic definitions in specifications. It is only syntactical structural definition with implied semantics derived from specification narrative.
Data Objects	Data objects are newer form of representation, where multiple data elements are associated with

[22] http://en.wikipedia.org/wiki/Context
[23] http://suo.ieee.org/
[24] http://en.wikipedia.org/wiki/Database
[25] http://en.wikipedia.org/wiki/XML_schema

Semantic Level	Description
	qualities of object. In object-oriented data models, labels of object qualities are associated together in syntactical structural definition, e.g., table of relational database where each record represents a unique object with various qualities expressed as fields of record. Data objects have no semantics defined except in narrative of specification. It is only a syntactical structural definition with implied semantics derived from specification narrative.
Data Elements	Data elements are typically defined as elements of a larger structure, e.g, object, message, data record, etc. They typically have their syntax defined within document specifications, as well as a narrative defining intended meaning of each data element. For example, "Temp" data element may have specification defining that it is 4th 8 bit word in a message content structure, or a field definition in database table. Specification will also typically indicate data value type used to represent content of data element, integer, decimal, etc. Semantics of data element will typically define whether it represents ambient temperature, temperature of a physical object, etc. It is only syntactical structural definition with implied semantics derived from specification narrative.
Raw Signals	Raw signals have no metadata. They are typically in some physical form, e.g., as electrical signals, etc., and have been classified in OSI model at the physical layer. The semantics of the signal is typically defined in documents, standards, or specifications and may have meaning identified at multiple layers of meta architecture model.

The extreme bottom levels of the table have no explicit semantics visible to the observer of the raw signal; all the semantic meaning would have to be discovered by reading documents of various forms describing the meaning of the signal. Interestingly, this can get quite complex if the observer is a design engineer or technician with knowledge of modern modulation and encoding schemes.

Semantic Expressive Example

An example of increasing semantic meaning applied to something as simple as a 0-12 Vdc signal is described in the following:

- For example, a signal on a lead may have a specification constraining its value within a range of +12 Vdc to 0 Vdc,
- While the next level of semantics might indicate that this signal represents a data element comprising a continuous or set of discrete values in this range,
- The next level of semantics might specify that these values represent a property of a data object, e.g., the height of some liquid in a vessel,
- While additional metadata might specify that object is a fuel storage container,
- While a domain schema might specify the vocabulary elements "height," "fueltank," 'vehicle',
- While additional semantics might structure some of the schema elements in a hierarchical taxonomy, e.g., fueltank $\supseteq$ storage tank,
- While additional semantics might indicate that the "height" is a field value in a database table definition for fueltanks,
- While a domain ontology might define all of the elements of a vehicle and the related properties of each vehicle element and their effects on other vehicle elements, such as "emptytank" equivalent to "0 Vdc" infers engine "nonoperation," etc.
- Additional semantics such as context would define the use and purpose of this data, such as indicating flight status and travel range.
- Additional semantic elements, such as directories linking domain semantic ontologies in the Semantic Web to compatible web sites, would enable multiple web sites with this information to represent semantic information about their vehicles and fuel states consistent with domain

ontology. Making all of this semantics explicit and available to a network of interacting entities would enable this raw signal and all of its meaning to be available to others in a Semantic Web sense.

As can be seen from this very simple example, moving up the semantic expressiveness level ensures greater confidence in achieving mutual consistent understanding.

4.1.2.7.1.4 Representative Standards for Semantic Expressiveness Categories—Values (3.1.4.7.1.4)

Typical standards and concepts used for some of these levels of semantic expressiveness are illustrated in Figure 4-23. Each of these levels may be used to represent the level of semantic expressiveness for the Semantic Interoperability analysis described previously. Thus semantic interoperability not only depends on the compatibility of the context, domain, and interaction types, but also on the level of semantic expressiveness used by each interacting entity to represent knowledge in these three areas.

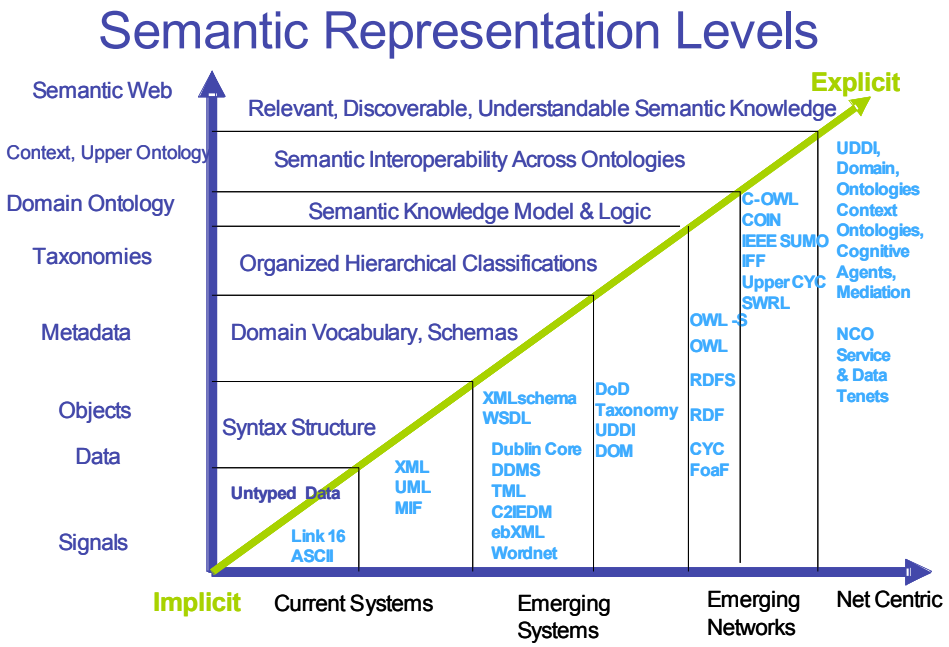


Figure 4-23. Standards for Levels of Semantic Expressiveness – Standards Dimension Values

There may be situations in which the domain knowledge is compatible between two interacting entities, but which use different levels of semantic representation, e.g., one uses a simple XML metadata-defined element, while the other uses a full domain OWL-described ontology. In this situation we have the potential for correct semantic interoperability, but the danger is that the semantics used by the XML associated entity is subtly different than the semantics of the OWL ontology using entity. Work would have to be done to ensure that the semantics are compatible by reading the intent of the XML defined data elements, and then determining if a mediation function could transform the XML syntax and implicit semantics to the OWL syntax and explicit semantic model.

4.1.2.7.1.5 Semantic Expressiveness Relations Values (3.1.4.7.1.5)

Given that the concepts in the analysis of Semantic Interoperability are similar or compatible, then what is left is to compare the representations themselves from a semantic perspective, not a syntactical perspective. If the two semantic representation levels are in the implicit region, the work can be quite substantial and problematic, with a highly probable nonsemantic interoperability result. If they both are in the explicit region, the work for mediation is feasible. If the two forms of semantic representation cross the explicit and implicit representation boundaries, it is generally feasible to achieve semantic interoperability by matching the lower level representation to the higher level. The semantic representation compatibility rules across architectural layers and semantic representation are shown in Figure 4-24.

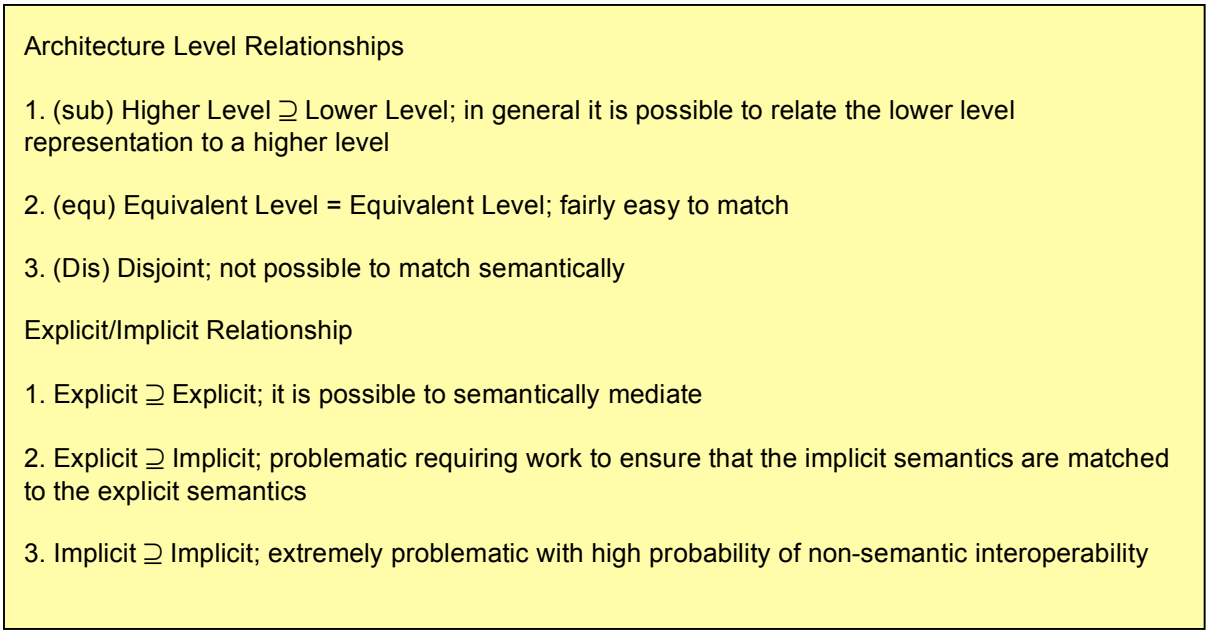


Figure 4-24. Semantic Expressiveness Compatibility Rules

4.2 Emerging Capability/Domain-Independent Scope Dimensions – 2.0

4.2.1 Organizational Business Model and Culture - 2.5

It is useful to consider and characterize the scope of a capability in terms of factors relating to the organizational context(s) in which the capability is exercised. In particular, there are relevant properties of organizations that vary based on their business models and cultures. These differences can manifest themselves in the form of the following:

- Policies constraining the nature of information sharing and the quality of service that is acceptable within the organizational scope.
- The degree to and manner in which interaction is facilitated within and outside the organizational scope.
- The degree to and manner in which the organization makes network access ubiquitous to its members.

In the following, the term "organization" is used to denote the organizational scope under consideration. It may be a conventional organization with relatively fixed boundaries, such as a military unit or a commercial company, or it may be a more logical organization with more fluid boundaries.

4.2.1.1 Policy-Driven Constraints - 2.5.1

Organizations establish policies that can constrain how efficiently and smoothly NCO can be conducted. Or, viewed conversely, said policies can enable NCO to be conducted more effectively (or at all) in a given context by ensuring that information is adequately protected and provided when needed. Such policies address issues of trust, information assurance, and quality of service.

4.2.1.1.1 Information Assurance Constraints - 2.5.1.1

Information Assurance is one of the key areas in which Policy-Driven Constraints apply. Real world tasks and capabilities require information flow within and across security domain boundaries, including interoperation with coalition and commercial partners. In a net-centric environment, trade-offs are required between information protection and information access to enable the net-centric ideal of posting information and services on the network for discovery and access. Organizations establish and enforce policies to regulate such information flow and to define their protection vs. access tradeoff decisions.

This dimension is closely coupled with the Net-Readiness IA dimension. The Policy-Driven Constraints addressed by this dimension help drive the technical mechanisms for implementing IA to enable net-readiness in the given organizational context or "virtual enterprise" within which operations are to be conducted over the network.

The emerging GIG IA architecture organizes IA capabilities in terms of four key IA functional areas: assured information sharing, highly available enterprise, cyber situational awareness and network defense, and assured enterprise management and control. Each of these encompasses a set of potential Policy-Based Constraints, mirroring the potential mechanisms in the Net-Readiness IA dimension. The value set discussion in the following is structured in accordance with these four areas, but the potential values are not yet fleshed out. Further subdimensions will emerge as part of this process.

Value Set

An initial assessment of the space. Some subdimensions will emerge.

- Assured Information Sharing
  - Identification and authentication
    - Where/when required
    - Mechanisms required by policy
    - Single sign-on scope and mechanisms
  - Labeling
    - Where/when required
    - Labeled object granularity
  - Authorization

        - Risk-adaptive policy
        - Discretionary access control policy
            - Person-based, role-based
        - Mandatory access control policy
            - MSL, MLS, Compartmented
        - Cross-domain access control policy
        - Authorization object granularity
- Highly Available Enterprise
    - IA transport partitioning policy (e.g., by COI)
    - IA policy-based routing policy
    - Protection of information in transit—application, network, link encryption
        - Where/when required
        - Mechanisms required by policy
- Cyber Situational Awareness and Network Defense
    - IA situational awareness needs
    - Vulnerability assessment and management policy
    - Attack response policy
    - Quarantining policy
- Assured Enterprise Management and Control
    - Policy for managing and protecting security information
        - Identity, privilege, policy, key/cert, IA management and control information
    - Audit policy

4.2.1.2 Networking Facilitation - 2.5.2

The degree to, and manner in which an organization encourages and facilitates networking can significantly affect the kinds of capabilities that an organization deploys and uses. In particular, it can affect the net-centricity of the capabilities that are used. Traditional organizations that rely more on person-to-person, telephone, and basic email interaction may not need nor want advanced, net-centric technologies and capabilities. In contrast, organizations that take advantage of, and encourage web-based interaction, online collaboration, and use of mobile devices will demand a more advanced technology infrastructure and use net-centric capabilities to a greater degree. Most organizations have recognized the trend toward the latter behavior and are moving in that direction, but there are still substantial differences between organizations in the degree to which they have made this organic to their culture.

The dimensions described below distinguish between the social and the technical factors involved in facilitating networking.

4.2.1.2.1 Social Facilitation of Networking - 2.5.2.1

Network Ubiquity in the context defined here is constrained by the concept of "Organizational Business Model and Culture" and as such focuses on the organization's networking of business elements rather than the problems of networking technology. We are interested in how prevalent throughout the organization the capabilities of networking are supported and available, e.g., the ubiquitous level of networking support and capabilities are encouraged across the organization, for business model elements and for the social networking of human agents in collaborative activities. The concept of COI, with its definition of members having common interests defines potential regions of network ubiquity within an organization, while similarly defining possible regions with little if any common interests. Yet, this definition ignores the larger organizational or enterprise concept in which different suborganizations within the enterprise though having common interests requiring intra-organizational networking, still require inter-organizational networking due to the different complementary roles each suborganization has in the enterprise. In the latter larger enterprise model there is need for enabling collaborative networking across organizational boundaries and roles.

The concept of "Network Ubiquity" is the basis for "Network" in the NCO term, e.g., the ability for gaining the advantages of networking through sharing and access to a multitude of individual capabilities, information, and resources. The focus of this dimension is to characterize the ability of these business model elements to participate in the network, to gain access to other network capabilities, and to form collaborative, mutually supporting structures that provide capabilities no one element could provide on its own. NCOIC is not interested in characterizing the mechanisms for achieving these structures, but rather the level of comprehensiveness for networking across the elements and agents of an organization.

Business Model Elements and Communities of Interest

The business model elements comprise human agents, business level functions, services, resources, processes, information, metadata, and created business artifacts. The key concepts defined here are that all business model elements required some level of networking to have achieved their effect; no element exists in isolation from the rest of the organization. The networking possibilities are exemplified in Figure 4-25 where we show:

1. All Business Model Elements are part of some suborganization.
2. COIs are formed with multiple business model elements who through networking are able to cooperate on achieving their goals and tasks.
3. Business model elements by definition of being a member of at least one-suborganization have a default community of interest associated with that suborganization.
4. Suborganizations may have multiple communities of interest within them.
5. Communities of interest may span suborganizations.
6. Business model elements may be members of more than one community of interest.

Business Model Element Networking Dependency Definition

The concept of networking to collaborate to have effective organizational impact by a business model element can range through a continuum from low to high:

- High networking dependency occurs when its effective organizational impact is heavily dependent on the collaboration of others in the community.

- Low networking strength occurs when its effective organizational impact is primarily due to its capability and does not require collaboration with other business model elements, e.g., it is more independent.

What is being characterized is the dependency of collaboration for a business model element to have effective organizational impact. A few examples illustrate these properties.

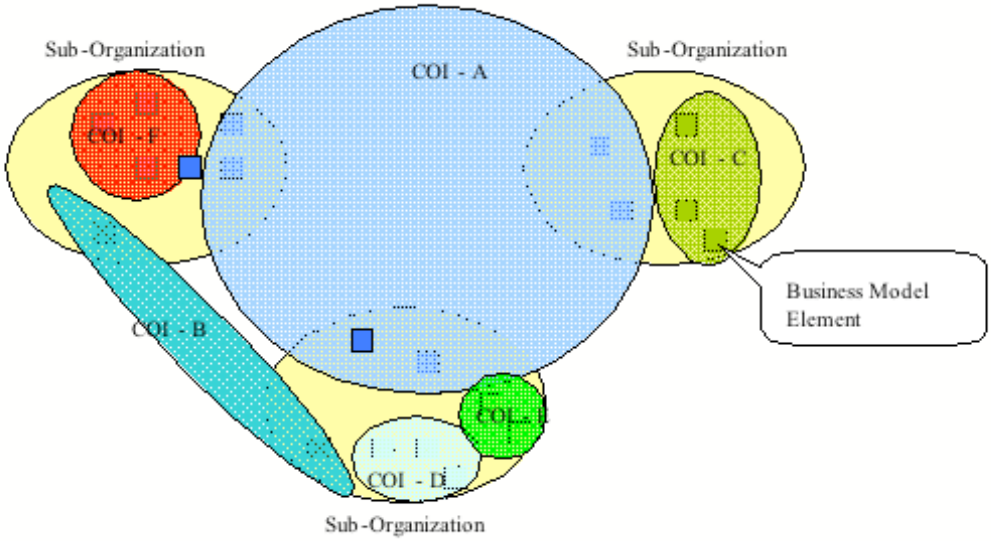


Figure 4-25. Networking Business Model Elements, Suborganizations and Communities of Interest (COIs)

First Responder Example

Typically, a first responder in an emergency response is highly trained in some discipline, and works in coordination with others at the emergency scene. A first responder's organizational impact is closely associated with its own local COI, rather than across the entire scene. Other first responders responding to the emergency have a similar ability to coordinate their respective response roles. Thus, a reasonable assessment of networking dependency would be ***moderate***, since there is required local coordination, but little coordination with the wider organization, once the location and nature of the emergency has been identified.

Emergency Response and Disaster Recovery Control Center

The members of this team control multiple first responder teams and COIx and are responsible for a coordinated response by a larger part of the whole organization. This includes determining assessment of the emergency, determining the appropriate response plan, identifying available resources, allocating resources, and operating the common, communication, and control procedures and adapting plans to the situation at hand. This organizational element would rate a network dependency value of ***high***, since it could not achieve its objectives without complete dependency on other teams to provide it with information, to act on their commands, and to communicate the real-world situation to the center as they find it.

Cultural Aspects of Networking

Other aspects of networking ubiquity are the constraints and expectations that are part of the organization's culture. Human agents in the organization are expected to fulfill the stated policies of the organization in achieving their goals, while fulfilling their traditional functional roles. Depending on the organizational culture, the individuals may, or may not, be expected to extend beyond their normal roles in collaborative efforts when the overall objective would fail without this adaptation. Rigid and static cultures constrain individuals from exercising adaptation through cooperation with others, while flexible cultures can adapt to new situations because of the dynamic nature of their organization and the freedom given to individual members and COIs to negotiate their responsibilities and activities. This defined a cultural aspect of network ubiquity that provides either a static or dynamic networking structure capability.

We can classify the following aspects of Cultural Network Ubiquity beyond networking dependency.

Culture Constraints

An organization is highly culturally constrained where there are a great many constraints on the individual members of the organization when networking to coordinate their activities, and where each activity and role is described in great detail, and where there are punishments for not conforming. In contrast at the other extreme, an organization that relies on the training and experience of its members and leaders to adapt to new situations, and effectively encourages this capability will in some sense have fewer cultural constraints on member actions.

We may want to consider a Culture Network Ubiquity subdimension with values ranging from highly culturally constrained to low culturally constrained.

Dynamic Aspects of Networking

This aspect of networking ubiquity refers to the capability of an organization to dynamically change its processes, plans, resource allocations, and policies to a range of real world situations. Since plans are premised on an expectation of reasonable predictability of the dynamics of a real world situation, it is invariably experienced that most real world situations do not conform to these expectations, and as a result, the effective organizational impact could be compromised by this mismatch. An organization with a model for dynamic adaptation and high coordination can effectively adjust its behaviors to these mismatches, while an organization that is static and does not have the capability to change its behavior will result in less effective organizational impact. This subdimension does not attempt to identify the reasons for this dynamic capability, as was done with the Culture Dependency, but rather just characterizes whether the networking structure is dynamic or static, or something in-between.

Ubiquity

Ubiquity refers to the concept of "being everywhere at once." We can say that network ubiquity connotes the concept of the networking capability or networking benefits being everywhere in the organizational business model and culture. This "being everywhere" dimension can be characterized as percentage metric of business elements that can participate in the networking, e.g., for identified desired networking capabilities, and/or the percentage of business elements that can gain access to another element. There may also be cases where a local subnetwork within a business organization is fully networked, while being constrained with respect to the larger community, as illustrated in Figure 4-25.

If one has an organizational capability to identify and classify objects that can only be accessed in one COI, and is not accessible to other COIs, and if there were five COIs involved, we might say that within the supported COI, the network ubiquity for object classification is 100%. Taking the perspective of the larger community, the network ubiquity characterization for this element is only 20%. Although this example captures the ubiquitous nature of the availability of this capability, there are other circumstances of availability, which could provide the same percentage of network ubiquity, but have entirely different interpretations. A second example with dynamic constraints instead of the previous static constraints could have accessible capability through all COIs, but with only sufficient capacity to support the equivalent of one COI population. Thus in this second example, at any moment in time, there is only 20% possibility throughout the organization to access this capability due to performance constraints, while in some sense 100% of the organization has potential accessibility. This indicates that this dimension should consider a static dimension and a dynamic dimension.

4.2.1.2.1.1 Community of Interest Dependency - 2.5.2.1.1

This subdimension of Social Facilitation of Networking is aimed at capturing the degree to which a given organization or set of organizations supporting a network centric capability or virtual enterprise utilize collaboration across a community of interest that addresses the operational space in which the organization operates, as well as with adjacent communities of interest that have some current or potential interaction with that operational space. This is a potentially rich area for future network centric capabilities and may leverage the growing body of work in social networks and the popular acceptance of social networking capabilities on the Internet. In addition, it reflects the recent initiatives within the US DOD to formalize communities of interest as an element of a net-centric data and service governance framework. An initial cut at a possible high-level value set is offered below. It may make sense to break this down to additional subdimensions for more object and quantifiable characterizations of this capability attribute.

Value Set

Table 4-8 Community of Interest Dependency Value Set

Value	Description
Independent Member of suborganization, only default suborganization COI	Member essentially independent, little if any coordination with others, except for job assignment
Collaborative Individual, Member of suborganization, only default suborganization COI	Member collaborates with others in default suborganization COI, has little role in larger organization
Collaborative Individual, Member of suborganization, default suborganization COI and possible other COIs in suborganization	Member also collaborates with other COIs in sub-organization, has little role with other suborganizations
Collaborative Individual, Member of suborganization, default suborganization COI and possible other COIs in multiple suborganizations	Member also collaborates with other suborganizations and multiple COIs.

4.2.1.2.1.2 Cultural Dependency - 2.5.2.1.2

This subdimension is intended to measure the degree to which the organizational culture of entities participating in a network centric capability exhibit and reward or facilitate network centric behavior within and among participating organizations. Clearly this is an emerging area being explored by a variety of parties, many of which come from outside the technical domain. The working group has

developed an initial value set below, but welcomes additional contributions in this area, including possible subdimensions and additional or alternate value sets for measuring this network centric attribute.

Value Set

Table 4-9. Cultural Dependency Value Set

Value	Description
Policy constrained member with specific defined roles for individuals	Members have no flexibility to adjust their activities or behaviors to changed situations, there are severely constrained to only perform specific actions according to plans and with restrictive policies limiting their activities with others, if any.
Members may collaborate with others according to less restrictive policies	Member collaborates with others within well defined policies for each situation prescribing what possible collaborations can occur. Situations not covered by policies are not covered.
Member can collaborate with others with only policy guidelines	Members may adapt their behaviors and negotiate different roles and tasks as adaptations to plans, when new situations occur and are only constrained by policy guidelines

4.2.1.2.1.3 Dynamic Structural Dependency - 2.5.2.1.3

Related to the previous dimension, this subdimension is aimed at characterizing the degree to which the organizations involved in a network centric capability or operation exhibit flexibility in organizational structure. For certain organizational types this may be dictated by law or regulation as well as culture, and thus is not something that a given organization has the freedom to change by itself. In other cases, it may be a product of the personalities and culture of the leadership of the organizations. Regardless of the cause or forces that drive organizational behavior in this area, the reality is that all organizations are not equally flexible in adapting their structure to support dynamic network centric capabilities and operations. This has real world consequences in the service interfaces and behavior exhibited by systems developed and operated by such organizations on the network. An initial value set for this dimension is proposed below.

Value Set

Table 4-10. Dynamic Structural Dependency Value Set

Value	Description
Static structure	Members have no flexibility to adjust their activities or behaviors to changed situations, the relationships between members are strictly defined and no changes are allowed.
Somewhat dynamic	The specific collaborative activities and roles can change within existing associations between members and COIs
Dynamic	Collaborative activities and roles can change and new associations can be made between members and COIs.

4.2.1.2.1.4 Accessibility - 2.5.2.1.4

This dimension is another measure of the degree to which organizations participating in a network centric capability facilitate social interaction among participants. It focuses on how well organizations make their participants visible and accessible to each other to support collaborative behaviors. The currently proposed value set below is focused on a single organization, or treats a multi-organizational

network centric capability as a single "virtual" organization. Future extensions to this dimension may want to consider measuring intra-organizational accessibility separately from inter-organizational accessibility. Most real-world organizations have significantly different policies and mechanisms that apply to internal communications vice collaboration with entities that are considered external to an organization. Such policies and mechanisms can clearly facilitate or inhibit network centric behavior of organizations participating in a network centric capability. The Working Group also realizes that a simple percentage of numbers of people measure may not adequately characterize this network centric attribute, but it is some indicator and is reasonably easy to measure. We welcome suggestions for refinement of the proposed value set and dimensional structure.

Value Set

This dimension has two values, one addressing how much of an organization a member of that organization has access to, and the other addressing how much of the organization has access to that member (Table 4-11).

Table 4-11. Accessibility Value Set

Value	Description
Member has access to subset of organization	% of members in organization COIs, or suborganizations this member has collaborative networking access to.
Subset of organization has access to this member	% of members in organization COIs, or suborganizations having collaborative networking access to this member

4.2.1.2.2 Technical Facilitation of Networking - 2.5.2.2

The social and organizational aspects of networking are crucial, but in today's world, much of the networking we do is facilitated and enabled by network technology. In this realm, NCOIC focuses on two areas:

- The various networking facilitation technologies themselves.
- The degree to which these technologies are accessible and available for use in a given context.

4.2.1.2.2.1 Facilitating Technology - 2.5.2.2.1

Technology to facilitate networking is an exploding, rapidly changing field. The technologies available today are already vastly different than they were a mere 5-10 years ago, and will probably be vastly different 5-10 years from now. Thus, this area of the SCOPE model must continually evolve to track these changes in technology.

The web is becoming a predominant player in this area. Some network facilitation technologies remain independent of the web, such as voice, email, instant messaging, and video, but all of these things are now available through the web as well, and it is likely that the web's role as an integrating infrastructure will continue. Because the web has vulnerabilities (as do some of the other technologies here, some in conjunction with the web), network safety technology to ensure users' trust in the network is also critical.

Value Set

Each of the following may represent a distinct subdimension or value space for evaluating technical capability in these areas. These must be fleshed out accordingly.

- Voice communication
- Video communication
- Electronic personal messaging (including email)
- Websites and portals
- Network services infrastructure
- Semantic network technology
- Online collaboration and sharing tools
- Remote network access infrastructure
- Mobile networking devices
- Network safety technology

4.2.1.2.2.2 Freedom of Network Access - 2.5.2.2.2

Even though technology can be a tremendous enabler of networked interaction, the amount and kind of access to such technology can vary widely in different contexts for reasons of cost, security, trust, maturity and reliability of available infrastructure, and so on.

One issue with freedom of network access is that organizations generally treat interaction with external organizations more guardedly than they do internal interaction, and establish barriers to inhibit such interaction. They establish policies for what kinds of information can be shared with whom, and establish technical mechanisms such as intranets to help keep information within their organizational boundaries. Some organizations are more open to external interaction than others, and this can be reflected in the policies they adopt and the infrastructure they establish to support their policies.

Value Type

Conceptual continuums

Value Set

Each of the three areas below may represent distinct subdimensions.

- Freedom of access to networking technology
  - No network access.
  - Network access restricted to physically secured areas.
  - Network access provided widely within workplace.
  - Remote network access enabled via VPN, dial-in, etc.
  - Remote network access encouraged via use of mobile devices such as laptops.
  - Ubiquitous network access encouraged or required via use of mobile devices such as laptops and other established and emerging mobile network access technologies.
- Freedom of network reach and sharing
  - No network reach.
  - Network reach restricted to localized network.

  - Network reach restricted to enclave, intranet, or similar boundary.
  - Inter-network access enabled, but narrowly limited by ports, protocols, and/or domains/Ips, etc.
  - General internet access enabled, with standard restrictions for security and trust reasons (standard firewall).
  - Open and broad internet access enabled, with minimal restrictions for safety.
- Freedom of network sharing
  - No network sharing.
  - Sharing of network resources across organizations is severely limited.
  - Sharing of network resources across organizations is enabled, based on standing agreements and well-regulated exceptions.
  - Sharing of network resources across organizations is encouraged (with appropriate safeguards) to promote inter-organizational networking and cooperation.

4.2.2 Life Cycle Control Dimension - 2.6

Evaluation of net-centricity of a system is shaped over its product life cycle by the controls placed on the life cycle phases. These phases span a cycle that begins with concept development, through operations, to the end of the system's life, commonly referred to as disposal. For purposes of discussion in this paper, the latter part of the development phase and the operation phase are the most significant. Controls provide both benefits and drawbacks, technical, cost, and schedule-related. These benefits and drawbacks may be measured by the SCOPE dimensions and value set associated therewith.

One should be cautioned that there are interdependencies between programs. These couplings interact, if not handled with care, in a way to reduce the accuracy of the SCOPE evaluation. One should recognize that network-centric systems operate in an evolutionary environment where new systems are being added, existing systems are being modified/extended, and older systems are being encapsulated or retired in a continuum.

The System Life Cycle occurs in an environment in which the system providers, operators and acquirers don't belong to a single organization, a single military service, a single government department, or even to a single country. This means that possible system interdependencies and interactions over the network are increasingly subject to coordination and collaboration among multiple stakeholders, including acquirers, builders, operators, and user communities.

To maintain consistency throughout this document, the subdimensions in the paragraphs below have more than one level. For example, the Stakeholder Alignment subdimension has values that may differ throughout the life cycle of the program. As a result, Stakeholder Alignment has values that differ over the life cycle. So, the Stakeholder Alignment subdimension is subdivided further into the life cycle/operations types as follows: (1) concept development, (2) engineering development, (3) initial operation, (4) coordinated operations, and (5) modified operations.

4.2.2.1 Stakeholder Alignment - 2.6.1

Ownership of a system has many aspects. Physical ownership, virtual ownership, and practical ownership define the stakeholders that use a system. In addition, the interaction between systems, or a system-of-systems, conjures a different vision depending on the role of ownership: assets, maintenance, and operations. There are other control dimensions that describe coordination of assets owned across military Service boundaries, department/agency boundaries, government/commercial boundaries, and transnational boundaries.

For military systems, a warfighter's stake in a system element is narrowly proscribed through training and doctrine. However, this training and doctrine must be capable of recognition of merger and overlap of systems on an ill-defined time line. From a logistics standpoint, delivery and maintenance is as vital as training and doctrine. Without on time delivery, modern combat doctrines are not possible. Without repair and maintenance or replacement in a timely manner, a warfighter's stake in a system may not contribute to NCO. If the logistics tail is not in step with the combat action, it is truly a case of the tail wagging the dog.

Coordination between diverse elements of the same force using the same or similar equipment can contribute to improved effectiveness in combat. This coordination is a tenet of net-centric Logistics. Similarly, coordination may achieve improved effectiveness even in quasi-operational domains when security contractors and military force elements need to cooperate.

Theatre commanders who are the virtual owners of systems of systems elements have a controlling ownership role that cannot be ignored. The ability to move forces in the environment, nominally a logistics role, has become a key parameter in warfare planning and execution. The information systems elements that can be reconfigured in a timely manner provide the tactical commander needed flexibility. Interoperability of elements can replace commonality as the parameter giving this flexibility in a system of systems. There will never be a case where commonality will be totally achieved, because it takes time to produce and/or deliver equipment and to a lesser extent, load software updates.

Ultimately, the goal would be to move from an acquisition system that is supported by one acquisition agent and one customer community toward aligning to a diverse acquisition system in which systems are being built and used by different enterprises and even countries. Coordination difficulty and amount of time required for acquisition may increase, but the improved effectiveness that results will be worthwhile. Viewed from the negative or feasibility perspective, the chance of achieving such coordination in what is being built by the different entities diminishes. That is why loose coupling approaches become the preferred way of doing things when enterprise and national boundaries are crossed.

To achieve interoperability in an international environment may not be easily achievable in specific instances. However, explicit multinational effort and adequate time may make it possible, assuming that the interests of the different enterprises/nations are in reasonable alignment, or "confluence."

Systems already deployed and operated are increasingly being enhanced and upgraded for affordability. The moment these systems are delivered, they need to evolve to accommodate interoperability with other systems coming online over time. In some sense, the current vogue in "acquisition versus support" makes this somewhat problematic, but network centricity drives this possibility. Also, many systems are now developed and deployed incrementally, using the spiral process. The spiral process offers opportunities for implementing changes to leverage network

capabilities that come on line after the initial system deployment. However, a system already deployed will have barriers/resistance to making changes to accommodate inherent system capabilities deployed after the system's requirements are "cast in concrete" due to budgetary constraints.

Value Type

Some of the value types that may be derived from the above (or similar) discussions are as follows:

- Program interdependence.
- Coordinated asset delivery.
- Acquisition diversity.
- Incremental development using a spiral process.

Value Set

The values relate to definition of the stakeholders and a description of how they interface to achieve net-centricity, or how they align. Clearly, the latter is the more complex problem. Stakeholder identification must address stakeholders over the life cycle. Alignment can range from understanding the identity of the other stakeholders in the concept development phase to physical interoperability with another stakeholder in an operational environment.

Table 4-12 provides an evaluation of Stakeholder Alignment. The common elements of stakeholder alignment are defined as follows for the development and deployment context:

- Net-Readiness—Ability to deliver capability in a network context.
- Capability/Domain-Independent Scope—The range of scope or context supported.
- Capability/Domain-Dependent Scope (performance)—Quantity, quality, speed, etc., of capability provided.
- Technical/Economic Feasibility—The feasibility or risk of providing capability.

The stakeholders described in previous paragraphs have various degrees of interest as a function of the life cycle and include the following:

- Concept developers (customers and suppliers).
- Engineering developers (contracting officer/buyer, contracting officers' technical representatives, oversight organizations).
- Test and evaluation organizations.
- User oganizations (warfighters, operators).
- System owners (various, dependent on item).
- Logistics organizations (product managers, etc., transporters, maintainers, buyers).

Table 4-12. Stakeholder Alignment Value Set

Subdimension	Value	Comments
Concept Development	Understanding of similar research efforts and activities in these fields	
Engineering Development	Net Readiness	Fit for GIG interface in the projected environment

	Capability Independent Scope	Verify overall scope or context in projected environment
	Capability Dependent Scope	Performance verification
	Technical/ Economic Feasibility	Shortcomings noted, cost realism, support development
Initial Operation	Net Readiness	Operational evaluation
	Capability Dependent Scope	Performance validation
Coordinated Operations	Net Readiness	Interoperability validation
	Capability Independent Scope	Scope re-evaluation
	Capability Dependent Scope	Performance re-evaluation
	Technical/ Economic Feasibility	Support costs and timelines validated
Modified Operations (Upgrade or downgrade)	Net Readiness	Interoperability re-validation
	Capability Independent Scope	Roles and modes of operations redefined
	Capability Dependent Scope	Performance re-evaluation
	Technical/ Economic Feasibility	Support costs re-validated

Clearly, weighted scores should be employed where possible, due to the particular system and phase of the life cycle.

4.2.2.2 Life-Cycle Timeline Congruence - 2.6.2

A first step in making acquisition a net-centric process is the "joint" program wherein the roles and responsibilities of the participants are well proscribed. To extend this concept to the larger user environment, roles for all service users based on their interaction with the specific procurement program must be created. In addition, interaction with other programs in the acquisition pipeline at stages where interaction is both possible and productive is a further measured value of net-centricity in a procurement activity.

From a technical perspective, collaboration between acquisition activities, in terms of technology forecasting, as well as attrition of the installed bases, is vital for a net-centric environment. Deploying a new technology in a net-centric manner implies compatibility with the displaced technology to the fullest extent, i.e., full backward compatibility. Clearly, this is possible with today's and tomorrow's technological advances, such as JTRS implementation of legacy waveforms.

As indicated previously, acquisition follows a prescribed path, typically a decade in duration. Information communication technology in the recent past has matured in somewhat less than half that time, resulting in the fielding of less capable equipment. Current emphasis on commercial standards helps to allow eventual use of commercial-like, if not commercial equipment. This shortens the time line, but produces other problems of ownership and procurement.

Within the current process defined milestones of program phases, the government program manager has considerable latitude to request collaboration, but not force processes leading to interoperability. A contractor's program manager can request collaborative actions, but can not make them happen. In general, the process tends to be to convince the other system program managers of the advantages of collaboration to reduce time lines in terms of ultimate deployment of system elements.

Systems already deployed and being operated are increasingly being enhanced and upgraded for affordability. The moment these systems are delivered, they still need to evolve to accommodate interoperability with other systems coming on line over time. In some sense, the current vogue in acquisition versus support makes this somewhat problematic, but network centricity drives this possibility. Many systems are now developed and deployed incrementally, using the spiral process,

offering opportunities for implementing changes to leverage other capabilities on the network that come on line after the initial deployment of a system. A system already deployed has barriers/resistance to making changes to accommodate new system capabilities deployed after the system's requirements are "cast in concrete" due to budgetary constraints.

Value Type

- Customer Process Stability—In government service, particularly within the military, there is a rotation process that contributes to instability during almost all phases of the life cycle. In the design of a military airplane, the procuring activity generally organizes a "cockpit working group" that advises the contractor in human aspects of cockpit design. This group meets every few months on a scheduled basis. It is not unusual to have a different pilot at each meeting who will repudiate the decisions made by his or her predecessor. To minimize this effect, strong processes must be placed into effect and observed to minimize the cost and schedule effects.
- Legislative Branch Support—Continuous efforts by the service sponsor and contractor are necessary. Instability caused by lobbying may be beyond the scope of this discussion.
- Technology Forecasting/Evaluation—For example, Moore's Law being misused. Moore's Law extrapolates growth of processing and storage capabilities based on history. Some contractors build this past progress into their plans for the future without having concrete knowledge on which to base their proposed design . If technology forecasting is correctly handled within contracting, it could include spiral development concepts to allow optional increases in capability that would be acceptable to the end users.

Value Set

Table 4-13 describes phased parameters relating to Timeline Congruence over the System's Life Cycle.

Table 4-13. Life Cycle Timeline Congruence Value Set

Subdimension	Value	Comments
Development Phase	Not constrained by parallel or competing developments or upgrades to existing products	Difficult to achieve a high score here
Operational Phase	Available in a timely manner	Carefully planned operational evaluation is necessary
Logistics Support	Time Definite Delivery	Coordination begins during development
	Failure Anticipation and Reporting	High level of BIT and automatic reporting
	Demand Based vs fixed capability	Supplier network
	In transit visibility	Already underway: RFID

Example

Development of the Global Positioning System (GPS) is an example of a success story where a spiral process for user equipment development was achieved in a natural but unplanned evolution. This program, conceived as a joint program managed by the Air Force, was planned as a standard Air Force program, which was to move through exploratory development, advanced development, and engineering development into production.

When advanced development was successfully concluded, the program evolved into a series of engineering developments in parallel with low rate production contracts. Through extensive testing, the units produced under these contracts fed back into engineering development data that, although delaying the completion of engineering development, actually advanced the long term program schedule through enhanced user feedback.

The ultimate military GPS user equipment deployment bifurcated into a GPS/Inertial Navigation System capability and a GPS only radio. Legislative branch pressure opened the program to civil use. The civil program has spawned applications never envisioned by the Joint Program Office (JPO) at its establishment, moving the around the development spiral on still another arm.

Stability of the JPO had several beneficial effects. The technical concept for the space segment and the user equipment segment changed little over the advanced development/engineering Development phases.

4.2.2.3 Cost of NCO Implementation - 2.6.3

Consideration of NCO implementation cost over the entire life cycle can serve to lessen the development impact of additional features, sometimes referred to as gilding. Budgetary constraints between development and production budgets can serve as a hindrance in that regard.

NCOIC serves a unique role works influence the services to examine the entire life cycle cost aspect of NCO features in development. Clearly, this is a task ahead of us, but one with a real payoff.

4.2.2.3.1 Cost of Timeline Incongruence - 2.6.3.1

The development phase of a program includes a high annual cost without a warfighter product as a resultant. This phase lasts typically 3 to 6 years in duration for a product or upgrade program, and a decade for a new platform. Inattention to the duration of this program makes it a target for termination either by its sponsor or congress. Interference by a competing program or parallel dependent program could be a crucial factor not only to the cost, but termination potential.

The operation phase of a program may be seriously affected by delays in the “own” system or other supporting systems.

When support activities do not properly forecast, maintain, and replace fielded equipment, the operators, being driven by combat plans, take alternative courses, sometimes at the expense of an unsupported fielded system, providing for premature disposal.

Value Type

- NCO Definition Stability—The most authoritative definition of NCO as viewed by DOD is found in “Net Centric Operations – Warfare (NCOW)” a document that is currently undergoing revision. Contained within this document are detailed descriptions of the requirements for NCO. Early versions of this document, which were released publicly, described the NCO Enterprise Services (NCES) that would form the information access for the nodes in the GIG. The information contained in these briefings indicates that the original nine NCES (services) have morphed into something else. To the extent that these services were contained in a contract with the government requires a revision to the effected program, providing instability.

- Standards Evolution—Examples of the effect of standards evolution

  Internet Protocol version 6 (Ipv6)—Ipv6 is a new internet protocol that provides increased addressing capability. The U.S. DOD has mandated that all development programs will move to this standard in the future, even though it is not complete and mature. It is not clear how this mandate has affected programs in development such as JTRS, which, at the time of this writing, is continuing to use the previous standard, Ipv4. Clearly, it would be more costly to implement this protocol in the future on the JTRS radios than it would have been to implement at the beginning of the program, assuming that Ipv6 was fully defined.

  8.33 KHz Channelization—VHF air traffic control frequency channel spacings from their onset were 25 KHz. In an effort to provide more channels, the European nations requested the ITU to reduce the spacing by a factor of 3 to 8.33 KHz. Aircraft that were not so equipped could be denied flight clearance by the agency governing European flights. With this threat, aircraft operators moved swiftly to equip their aircraft, increasing their communications costs, but providing spectrum for the future.

- Information Security Policies—The GIG is a concept that will allow users and providers a means of information exchange not currently available. Existing systems that restrict information exchange are denoted as "stovepipes" in the vernacular, in part due to the classification of information contained therein. One goal of the GIG is to eliminate the stovepipes, thus freeing the bandwidth to be shared by the enterprise (in this case, the U.S. government). The method of accomplishing this is to implement Multiple Independent Levels of Security (MILS). MILS is a technique that allows a mix of classified information to flow across a given communication channel. When implemented, MILS will allow not only U.S., but international users, to exchange information as provided by intergovernmental treaties and laws of each nation.

Value Set

Table 4-14 provides the dimensions and appropriate values for this dimension.

Table 4-14. Cost of Timeline Incongruence Value Set

Subdimension	Value	Comments
Development Phase	Not constrained by parallel or competing developments or upgrades to existing products	Difficult to achieve a high score here
Operational Phase	Available in a timely manner	Carefully planned operational evaluation is necessary
Logistics Support	Time Definite Delivery	Coordination begins during development
	Failure Anticipation and Reporting	High level of BIT and automatic reporting
	Demand Based vs. fixed capability	Supplier network
	In transit visibility	Already underway: RFID

Clearly, weighted scores should be employed where possible due to the particular system context.

Example

NCES Services have been cited as the keys to GIG interface. From the originally defined 9 services, the U.S. government has realized the need for flexibility in their definition. To meet this need, a reorganization of the NCES services is taking place. This is happening while the original 9 are still in work and have been cited in development contracts. As a result, programs that were started with the

original 9 and have developed workarounds due to lack of definition may be delayed due to needed scope changes. Changes of scope are costly, and provide a variable delay to each affected program.

4.2.2.3.2 Cost Allocation to Stakeholders - 2.6.3.2

A joint program presents unique aspects in its allocation of cost. Typically, the U.S. Marine Corps desires to use equipment that is compatible with U.S. Army equipment, and often chooses the exact equipment. This presents a problem for the developing command that is trying to assess a fair allocation of cost. What was a fair allocation of cost for the service may change over the life cycle due to numerous factors. This type of allocation was seen with the Eurofighter program, which caused delays and cost over runs.

Value Type

- Operational Planning—Operational planning has several phases: (1) generic training, (2) war planning, (3) battle planning, and (4) mission planning. Each has an effect on utilization of the assets needed to perform a mission. The cost allocation to the service or platform procurement entity reflects the history of how a similar set of missions have been performed in previous conflicts. To assess the cost of new equipment that executes a different operational plan than was performed a decade ago to a service based on history may be erroneous. This factor must be considered in cost allocation.
- Use of Commercial Equipment—Increasingly the use of commercial equipment (and software) is finding favor with the government in the information communication field. Commercial processing and communication equipment is virtually thrown away in less than a decade, often less than 5 years. If this equipment can be used in the military environment, studies have shown that it is cost effective by an order of magnitude to switch to commercial equipment.
- Coordination Within Government—There are many facets of coordination on government programs. Following are examples for interdepartmental and intradepartmental coordination.

  Defense Department and State Department—In the development phase of the life cycle, coordination between stakeholders can have the greatest effect. In the world of coalition warfare, it does not help to join the battle to find that one flank can not interchange information. It is important to consider the perfecting treaties with friendly parties at the beginning of a program, so that information may be legally interchanged between coalition parties on the battlefield. To achieve positive effects for the coalition partner without violating national sovereignty, technology upgrades are providing a way forward. MILS is among those programs working to make this achievable with the use of commercial technology. These techniques will allow a mixture of information to be transferred over the same media between diverse parties without compromise of the security of any party, given a well defined State/Defense Department strategy.

  Branches of the Military—Within the DoD, a series of efforts has been underway for three decades to coordinate activities.

  The result is a development cycle time for major systems, and some products, that has approximately doubled during that period. Concurrently, system effectiveness has improved, in part due to better coordination between the services. The current administration has a major effort not only to retain and improve the levels of effectiveness, but also to slash the development cycle time in excess of 50%. In terms of operational performance from a conventional warfighting perspective, operational effectiveness has been substantially

increased, due in large part because of the attention to basic communications interoperability between services and the supporting technology thereto. The current NCO effort to improve information interoperability should achieve positive results due to the standards based efforts underway in development and acquisition.

Value Set

The value for this element may be expressed as a number between 0 and 5, with 5 being optimum.

4.2.2.3.3 Incentive Model - 2.6.3.3

Uncertainties in acquisition may be categorized as Cost, Schedule, and Technical. Cost uncertainties largely reflect technical and schedule parameters, although in a System of Systems environment, cost uncertainties may reflect uncertainties in other programs, related or not, due to budget considerations. From a technical standpoint, planning and execution are both germane. In program planning, technology forecasting is a powerful tool that may be used not only for determining the technical risk to undertake but also to understand what technical course other similar developments are taking, and what modifications are taking place in the installed base. In execution, coordination between programs should not be underestimated for its potential benefits. It is vital not to make the same mistake twice. A general issue in network centric environments concerns the lack of interoperability knowledge of additional systems in an environment as a particular system is in development. That risk, dependency management, raises indemnification issues for a system incorporating some service or capability based on the existence of some other system's service that has not yet been deployed. This may be modeled as an incentive and risk absorption issue.

Value Type

Incentives may have several forms. A fixed price contract where a contractor can reap the rewards of reduced cost provides the most incentive. For a development contract, the U.S. government often uses a Cost type contract with Fixed, Incentive, or Award Fee, or some combination thereof.

There may be more subtle incentives to a contractor in terms of market share: follow on business, adjacent market penetration, or political advantage.

Value Set

The value for this element may be expressed as a number between 0 and 5, with 5 being optimum.

Example

The B-1B was built on an incentivised schedule to have 100 aircraft flying by a date determined by the president, with the goal of influencing the Soviet to continue the Cold War. This fixed-price contract achieved its goals in a short time through a streamlined process.

4.2.3 Emerging Semantic Interoperability Dimensions – 2.7

4.2.3.1 Global Versus Specialization of Domain Knowledge for Communities of Interest Dimension (3.2.5.5)

The expansion of common knowledge is globally expanding due to the technologies and applications shortening time and distance for communications, news, entertainment, and travel. This has had the

effect of increasing global knowledge and thereby the capability of many languages to represent similar concepts, subject to social and other constraints that influence meaning. Conversely, there is increasing complexity requiring specialization of knowledge where separate communities of interest create their own vocabularies and concepts, which are not understood without significant education.

These two counter forces (Figure 4-26) one increasing ubiquitous knowledge through globalization and the other decreasing ubiquity through specialization, is affecting the ability for humans to reach a common understanding of shared knowledge. The failure of semantic consistency is due less to lack of understanding of a language, but rather due to the inconsistency of world models and associated beliefs that interpret communicated expressions according to this model, and also by differences in deep background knowledge underlying the world model.

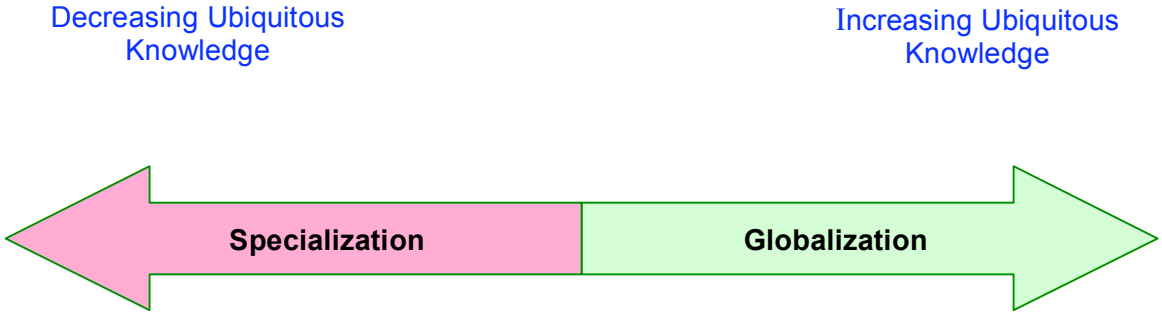


Figure 4-26. Globalization versus Specialization

Larger organizations with multiple communities of interest will be required to spend resources to enable its specialized knowledge to be expressed in a form that can be used and understood by other communities. Examples already exist in the health care community where lay people can acquire medical dictionaries and web descriptions of pathologies of diseases and possible treatment, which most reasonably educated people can understand, while other literature on the same subject is much more technical and specialized for physician use. The same knowledge is represented at different levels of detail for different but overlapping purposes. The background knowledge that a physician brings with his model is much more objective and based on science and education and training than the patient brings with her background knowledge. They need to communicate a shared understanding. In this case, the physician's knowledge overlaps and subsumes the patient's knowledge in this area.

The problem of specialization is especially true where one community requires information from another community for its purposes, e.g., expense reports for accounting. The specialized community defines the knowledge needed in terms that anyone can understand, but is directly interpretable by accountants to their expense accounting concepts. The person providing expense reports does not require specialized accounting knowledge to communicate relevant accounting information. The lesson here is that specialized communities who define the information required from other communities in simpler terms without having to share specialized metadata about their world model, can more successfully map the received information to their specialized world model.. One would be interested in whether the domain knowledge is oriented toward a more global common knowledge or to a specialized community. Harrison's reference on "Internal Medicine" would be an example of a specialized knowledge within a community of interest, medical students and physicians. Webster's Dictionary would be an example of a more global domain of knowledge that is valuable to a very large community of humans with no inherent specialization classification, such as physicians.

Classification of the global nature of the instances of domain knowledge, context knowledge, intent knowledge comprising the semantic interaction could be specified anywhere on a continuum between Specialized and Global. As we approach specialized knowledge, we fine typically a unique lexicon, unique semantic models of concepts, and fewer humans who would understand this specialized knowledge at a detailed level. Over time, what was once specialized could become more global, such as ubiquitous integration of this specialized knowledge into education curriculums. Most students in the U.S. are now educated in the concepts of geometry, while only a few hundred years ago we would have found only the elite at universities being exposed to the concepts of Euclid's geometry. With respect to literacy, universal education in developed countries has increased the literacy level of the general population and expanded their general knowledge, while in centuries past, only the elite were literate.

4.2.3.1.1 Global versus Specialization of Domain Knowledge for Communities of Interest—Dimension Values (3.2.5.5.1)

When using this dimension model (Figure 4-27) first determine the proportional size of the population being evaluated (Evaluated Population) with respect to a Reference Population and then determine whether the knowledge is specialized or common with respect to the Evaluative Population. This analysis should map an evaluation to one of the four quadrant values:

- Small-Specialized
- Small-Common
- Global-Specialized
- Global-Common

If greater detail is needed one could place an indication (X) on the diagram to notionally identify where in the quadrant the evaluation result occurs.

Definitions

- Reference Population ≡ the base population, which the Evaluative Population size is referred to as either Small or Global
- Evaluative Population ≡ the population, which is evaluated with respect to specialized or common knowledge
- Global – Small ≡ proportional size of Evaluative Population with respect to Reference Population (Evaluative/Reference)
    - Small = (Evaluative/Reference) ≤ 0.5
    - Global = (Evaluative /Reference) > 0.5
- Specialized ≡ where the detailed understanding of the knowledge within a Evaluative Population varies due to its complexity and the need for this level of detail by the members of this population. Not all surgeons will have the detailed knowledge of performing heart surgery. The knowledge is specialized to varying levels of detail among surgeons, Some will know the detailed procedures, while others will only know the situations where heart surgery is indicated.

- Common ≡ where the knowledge is commonly understood at the same level of detail among the members of the Evaluative Population. All surgeons are aware of the need to prevent accidental infection during the surgical procedure.

Medical Example 1 Small-Common

Reference Population: Surgeons of the United States

Evaluative Population: Cardiac Surgeons:

Knowledge: Open Heart Surgery

Population Proportional Value: Small

Specialization Level Value: Common

From this set of values for Population Context and Specialization Level we see that the correct classification is "Small–Common," indicating that only a small percentage of the reference population is being evaluated and the knowledge of open heart surgery is common among all members of the cardiac surgeon population.

Medical Example 2 Small-Specialized

Reference Population: Surgeons of the United States

Evaluative Population: Cardiac Surgeons:

Knowledge: Quadruple bypass open heart surgery

Population Proportional Value: Small

Specialization Level Value: Specialized

From this set of values for Population Context and Specialization Level we see that the correct classification is "Small–Specialized," indicating that only a small percentage of the reference population is being evaluated and that detailed knowledge of this surgical procedure is varied among the Evaluative Population.

Medical Example 3 Global –Common

Reference Population: Surgeons of the United States

Evaluative Population: Board Certified Surgeons

Knowledge: Infection prevention techniques during surgery

Population Proportional Value: Global

Specialization Level Value: Common

From this set of values for Population and Specialization Level we see that the correct classification is "Global–Common," indicating that a large percentage of the reference population is being evaluated and that detailed knowledge of infection prevention is uniform or ubiquitous among the Evaluative Population.

Medical Example 4 Global-Specialized

Reference Population: Cardiac Surgeons

Evaluative Population: Board Certified Cardiac Surgeons

Knowledge: Quadruple bypass open heart surgery

Population Proportional Value: Global

Specialization Level Value: Specialized

From this set of values for Population and Specialization Level we see that the correct classification is "Global–Specialized," indicating that a large percentage of the reference population is being evaluated and that detailed knowledge of this surgical procedure is varied among the Evaluative Population.

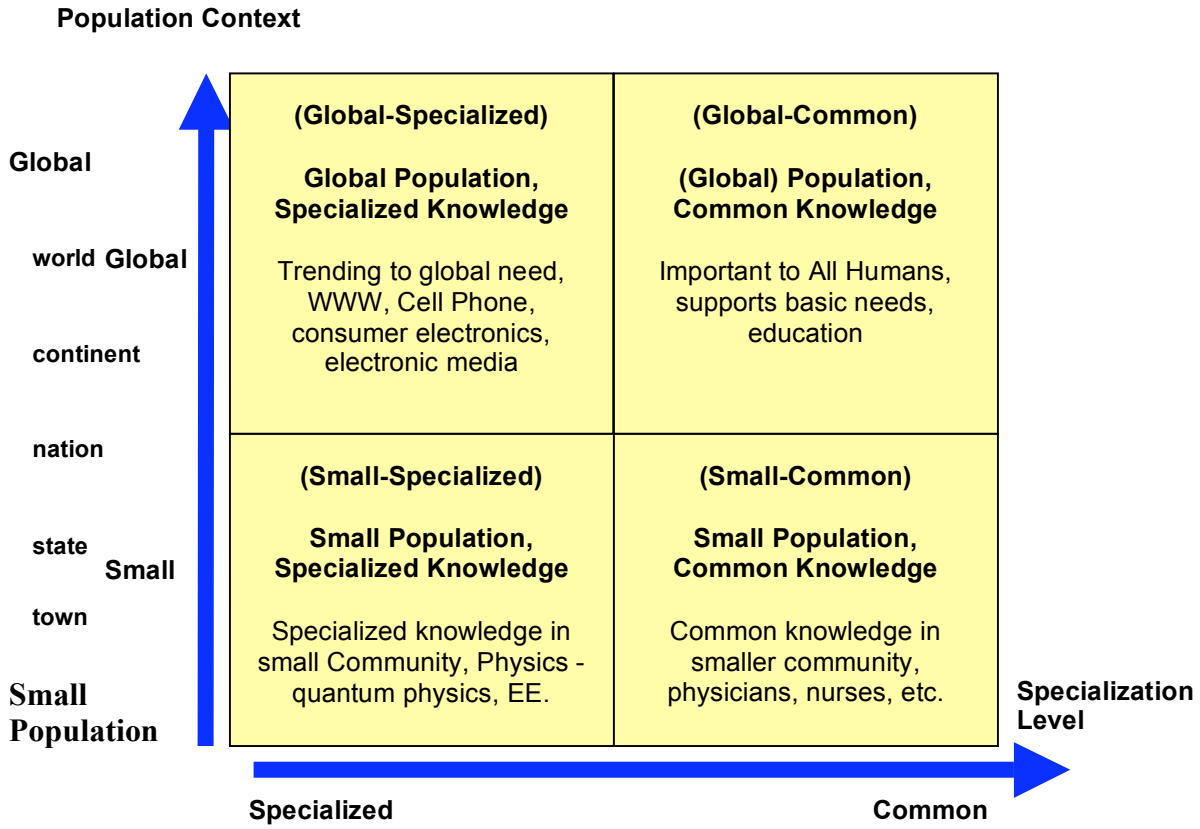


Figure 4-27. Global versus Specialized Knowledge Classifications

4.3 Emerging Technical and Economic Feasibility Dimensions

4.3.1 Emerging Run-Time Computing Resources Needed Dimensions

4.3.1.1 Processor Utilization (3.4.3.3)

Processor utilization involves the amount of processing resources an application is allowed versus what it needs to function. Applications come with a variety of requirements involving both execution speed and utilization of peripheral processes. Peripheral processes are processes that occur on other devices connected to the central processor by a bus. Some peripheral processes may involve mandatory

synchronization events with the master process. Execution speed requirements vary from best effort to fixed deadline to hard real-time.

Processor utilization itself is an extremely complex topic. In the past, processor speed was often described in terms of MIPS. Recently, it has been described in terms of processing cycles per second, e.g. 300 MHz, and most recently the use of equivalent MHz is finding favor. There are numerous phenomena that can affect processing speed:

- The type of cache used.
- The number of peripheral processes required for mandatory synchronization events.
- The number of mandatory synchronization events.
- The speed of the busses connecting the processor to peripheral devices.
- The number of processors and how tasks are dispatched.
- The operating system and how it schedules processing. The operating system may impose restraints on the amount of resources a process can use.

The amount of resources allowed may vary with processor loading and is usually controlled by the operating system based on priority. Processors utilized may exist within a single element or be distributed over a network.

Value Type

Subjective client and server processor resources required.

Value Set

- Respond whenever.
- Respond within time limit.
- Respond quickly.
- Hard real-time requirements.

Example

The effect of this example on net-centricity is based on the expected/required performance of the web service or application. Web service applications may use a client-server methodology where the main part of the service runs on a dedicated server. As such, the priority can be raised as necessary. Client applications usually run at the lowest priority on the user's host machine and share available processor resources with all other similar applications. As such they are subject to being interrupted for varying periods of time by higher-priority applications. Because of the nondeterministic nature of network services, processes with hard deadlines are best run on dedicated processors.

4.3.1.2 Nodal Quality-of-Service–(3.4.3.4)

Nodal QoS may be a consideration for net-centricity especially if a particular level or type of QoS is required. In this respect it is similar to processor loading except it deals with the loading of the communications media providing network access to services on a given set of network nodes. This is in contrast to overall network transport QoS, which is not associated with the specific network capacity of any given node. Services with high bandwidth requirements for service invocation/response and/or very high service request/response frequencies can generate local network capacity bottlenecks on the

nodes, which host those services. This can be especially true in the case of forward deployed or "tactical edge" service providing nodes involving things like ISR platform data collection/access in operationally "hot" areas. Nodal QoS may include best effort processing, bandwidth reservation for low-latency, priority processing, etc. Elements that impact QoS include the available bandwidth and the instantaneous load on the nodal network. The instantaneous load may vary based on application interactions resulting both from the nodal network as well as external applications.

Value Type

QoS requirement

Value Set

- Best effort.
- Reserved bandwidth for low latency communications plus best effort for unreserved.
- Reserved bandwidth for low latency communications plus priority processing for unreserved.
- Priority processing for both low latency reserved and unreserved.

Example

Voice and streaming video are usually considered low-latency communications. Often a portion of the available bandwidth is reserved for low-latency communications, and the rest is used for all other communications. The unreserved bandwidth can be best effort or it may include processing messages based on priority. In the case of low-latency communications voice usually has a higher priority than streaming video. When the low latency reserved bandwidth is insufficient, it may be necessary to either institute priority processing or increase the reserved bandwidth.

Glossary

Term	Definition
Autonomic	Acting or occurring involuntarily, without conscious control.
Autonomic Computing	Computer systems capable of self-management
Capability	The ability to perform actions or tasks
Context	The surroundings, circumstances, environment, background or settings, which determine, specify, or clarify the meaning of an event
Effects-Based Operations	
Enterprise	A company, business, organization, or other purposeful endeavor
Interoperability	The ability of two or more systems or components to exchange information and to use the information that has been exchanged.
Knowledge	All cognitive expectances that an individual or organization actor uses to interpret situations and to generate activities.
Net-Centric	Participating as a part of a continuously-evolving, complex community of people, devices, information and services interconnected by a communications network to achieve optimal benefit of resources and better synchronization of events and their consequences.
Operations	The method or practice by which actions are done;
Programs	A set of structured activities
Semantic	Of or relating to meaning or the study of meaning
Semantic Interoperability	The mutually consistent interpretation of shared knowledge between networked entities consistent with a semantic model in a defined context.
System	A group of independent but interrelated elements comprising a unified whole.
Swim Lane	A type of process flow diagram that depicts what or who is working on a particular subset of a process.
Time Binding	The amount of time required for a (SOA) service requestor and (SOA) provider to bind to each other.

Acronyms

A partial set of Acronyms used in this document is provided here. Consider referencing the NCOIC Lexicon for additional information on any of the net centric terms used in this document: https://www.ncoic.org/wiki/Lexicon

ACL	Agent Communication Language
API	Application Programming Interface
ASC	Accredited Standards Committee
ASCII	American Standard Code for Information Interchange
C2IEDM	Command and Control Information Exchange Data Model
CA	Cognitive Agents
CCJO	Capstone Concept for Joint Operations
CENTRIXS	Combined Enterprise Regional Information Exchange System
COCOMS	Combatant Commands
COI	Community of Interest
COTS	Commercial off the Shelf
CPU	Central Processing Unit
DBMS	Database Management System
DNS	Domain Name Service
DOD	Department of Defense
DoDAF	DoD Architecture Framework
DOTMLPF	Doctrine, Organization, Training, Materiel, Learning and Education, Personnel, and Facilities
DRM	Data Reference Model
DVD	Digital Video Disc
EBO	Effects-Based Operations
EDI	Electronic Data Interchange
FCB	Functional Capability Board
FEA	Federal Enterprise Architecture
FIPA	Foundation for Intelligent Physical Agents
GIG	Global Information Grid
GOTS	Government off the Shelf
GUI	Graphical User Interface
HTML	Hypertext Markup Language
IA	Information Assurance
IEEE	Institute of Electrical and Electronics Engineers
IO	Information Operations

IP	Internet Protocol
IER	Information Exchange Requirement
IPT	Integrated Product Team
JCIDS	Joint Capabilities Integration and Development System
JFC	Joint Functional Concept
JIC	Joint Integrated Concept
JOC	Joint Operating Concept
JTA	Joint Technical Architecture
KPP	Key Performance Parameter
LAN	Local Area Network
LISI	Levels of Information System Interoperability
MIB	Management Information Base (network configuration data base)
MIPS	Million Instructions Per Second
MLS	Multi Level Security
MoDAF	Ministry of Defence Architecture Framework
MOE	Measure of Effectiveness
MOP	Measure of Performance
MSL	Multiplc Security Levels
NASA	National Aeronautics and Space Administration
NATO	North Atlantic Treaty Organization
NCES	Network-Centric Enterprise Services
NCO	Network-Centric Operations
NCOIC	Network-Centric Operations Industry Consortium
NCOW RM	Network-Centric Operations and Warfare Reference Model
NGO	Nongovernmental Organization
NIAG	NATO Industrial Advisory Group
NML	NATO Maturity Level
NNEC	NATO Network Enabled Capability
NR KPP	Net-Ready Key Performance Parameter
OWL	Web Ontology Language
PAID	Procedures, Applications, Infrastructure, and Data
PKI	Public Key Infrastructure
PMESII	Political, Military, Economic, Social, Infrastructure, and Information
QoS	Quality of Service
RA	Reactive Agent
RAM	Random Access Memory
R&D	Research and Development

SCOPE	Systems, Capabilities, Operations, Programs, and Enterprises
SICF	Semantic Interoperability Conceptual
SIF	Semantic Interoperability Framework
SII WG	Services and Information Interoperability Working Group
SNMP	Simple Network Management Protocol
SoSA	System of Systems Analysis
STANAG	NATO Standard
TOGAF	The Open Group Architecture Framework
TRL	Technology Readiness Level
UDDI	Universal Description Discovery and Integration
URL	Universal Resource Locator
U.S.	United States
W3C	Worldwide Web Consortium
WAN	Wide Area Network
WG	Working Group
WSDL	Pg 4
WWW	Worldwide Web
XML	eXtensible Markup Language

References

[1] David S. Alberts, John J. Garstka, Fredrick P. Stein, *Network Centric Warfare: Developing and Leveraging Information Superiority*, CCRP Publ., 2nd Edition (Revised). Aug 1999, Second Print Feb 2000.

[2] David S. Alberts, John J. Garstka, Richard E. Hayes, David A. Signori, *Understanding Information Age Warfare*, CCRP Publ., 2001.

[3] (JCIDS) CJCSI 3170.01E. "Joint Capabilities Integration Development System," May 11, 2005 (https://acc.dau.mil/CommunityBrowser.aspx?id=18467)

[4] NNEC NATO Networked Enabled Capability - see the following link for current information http://194.7.80.153/website/home.asp?menuid=10

[5] DOD Architecture Framework Version 1.5 - http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_I.pdf, http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_II.pdf, http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_III.pdf.

[6] MODAF - http://www.modaf.com.

[7] TOGAF - http://www.opengroup.org/architecture/togaf8-doc/arch/.

[8] GIG - http://www.nsa.gov/ia/industry/gig.cfm.

[9] NCES - http://www.disa.mil/nces/.

[10] (LISI) C4ISR Interoperability Working Group, Department of Defense. Levels of Information Systems Interoperability (LISI). Washington, D.C., 1998 – http://www.sei.cmu.edu/isis/guide/introduction/lisi.htm.

[11] (Intergalactic Radiator) Adapted from a presentation by Capt Yurchak of the US Navy OPNAV, dated 1999

[12] (JOC) Joint Operating Concepts http://www.dtic.mil/jointvision/joc.htm

[13] (JFC) Joint Functional Concepts http://www.dtic.mil/futurejointwarfare/concepts/approved_ccjov2.pdf.

[14] (JIC) Joint Integrating Concepts http://www.dtic.mil/futurejointwarfare/jic.htm

[15] (JFC) Joint Future Concepts – see [16] below

[16] (CCJO) DoD CAPSTONE CONCEPT for JOINT OPERATIONS, Version 2.0, August 2005 - http://www.dtic.mil/futurejointwarfare/concepts/approved_ccjov2.pdf.

[17] (DOTMLPF) Defined as Doctrine, Organization, Training, Materiel, Leadership and education, Personnel, and Facilities in Joint Pub 1-02 http://www.dtic.mil/doctrine/jel/new_pubs/jp1_02.pdf

[18] (PMESII) Political, Military, Economic, Social, Infrastructure, and Information http://www.dodccrp.org/events/2006_CCRTS/html/papers/007.pdf

[19] (EBO) Effects Based Operations – see [18] above

Appendix A - Future SCOPE Dimensions

Introduction

During development of the current version of the SCOPE model, a number of dimensions and subdimensions were identified by the team as part of the overall dimension tree structure. However, while there was general agreement on the need for these dimensions in the model on a conceptual basis, they were insufficiently developed to include them in this version of the SCOPE document. In some cases, this was due to lack of volunteer authors. In other cases, it was resource conflicts with established NCOIC WGs regarding ownership of specific SCOPE dimensions. For example, the Network Utilization, Latency, and Availability dimensions overlap the technical expertise of the Mobile Network WG, but they could not spare the resources to provide their expertise to the SCOPE document in these areas without impacting their own delivery commitments. In other areas, the concept might be clear enough, but translating it into reasonably measurable value types and value sets has proven to be a challenge.

We anticipate that most, if not all, of these dimensions will be included in the next release of the SCOPE model. We also expect that the need for additional dimensions and subdimensions will be uncovered by member companies and by NCOIC IPT participants as they apply SCOPE in the development of operational descriptions (ODs) and protocol functional collections (PFCs). Readers who are interested in this general area or who have specific suggestions or contributions to make to the SCOPE model are encouraged to join the NCOIC SCOPE WG and help flesh out these and other dimensions for measuring/assessing network centricity..

Net-Readiness Dimensions

1. Evolvability
    a. Service Evolvability
    b. Application Evolvability
    c. Information Evolvability
2. Architecture Dependency

Capability/Domain-Independent Dimensions

1. Overall Scope (existing dimension – possible new subdimensions include:)
    a. Enterprise Topology
2. Policy-Driven Constraints (exists as emerging dimension – possible new subdimensions include:)
    a. Trust Constraints
    b. Quality-of-Service Constraints
3. Inter-Enterprise Adaptability

Technical and Economic Feasibility Dimensions

1. Transport Capacity Needed (existing dimension – possible new subdimensions include:)
    a. Network Utilization

i. Network Throughput
ii. Class of Network traffic mix
iii. Percentage of Network Capacity

b. Network Availability
i. Network Robustness
ii. Network Mobility

2. Service Capacity Needed
a. Service Invocation Rate
b. Simultaneous Service Sessions
c. Information Transaction Capacity