



C4ISR Architectures and Software Architectures

Rich Hilliard

rh@mitre.org

IEEE Architecture Working Group

<http://www.pithecanthropus.com/~awg/>



Contents

- ◆ “Architecture” in Context
- ◆ What is the *C4ISR Architecture Framework*?
- ◆ Issues with the *Framework*
- ◆ How to fix it?
- ◆ How does it relate to “Software Architecture”?



<Adjective> Architectures

- ◆ Application Architectures
- ◆ Data Architectures
- ◆ Enterprise Architectures
- ◆ Logical Architecture
- ◆ Makefile Architectures
- ◆ Operational Architectures
- ◆ Physical Architectures
- ◆ Security Architectures
- ◆ Systems Architectures
- ◆ Technical Architectures
- ◆ Occupant Architectures
- ◆ Heating and Lighting Architectures
- ◆ Building Code Architectures



Two Definitions We Like

- ◆ An **architecture** is the highest level (essential, unifying) concept of a system in its environment
 - IEEE Architecture Working Group, 1995
- ◆ The structure of the components of a program/system, their interrelationships, and principles and guidelines governing their design and evolution over time.
- ◆ SEI, 1995

What is the *Framework*?

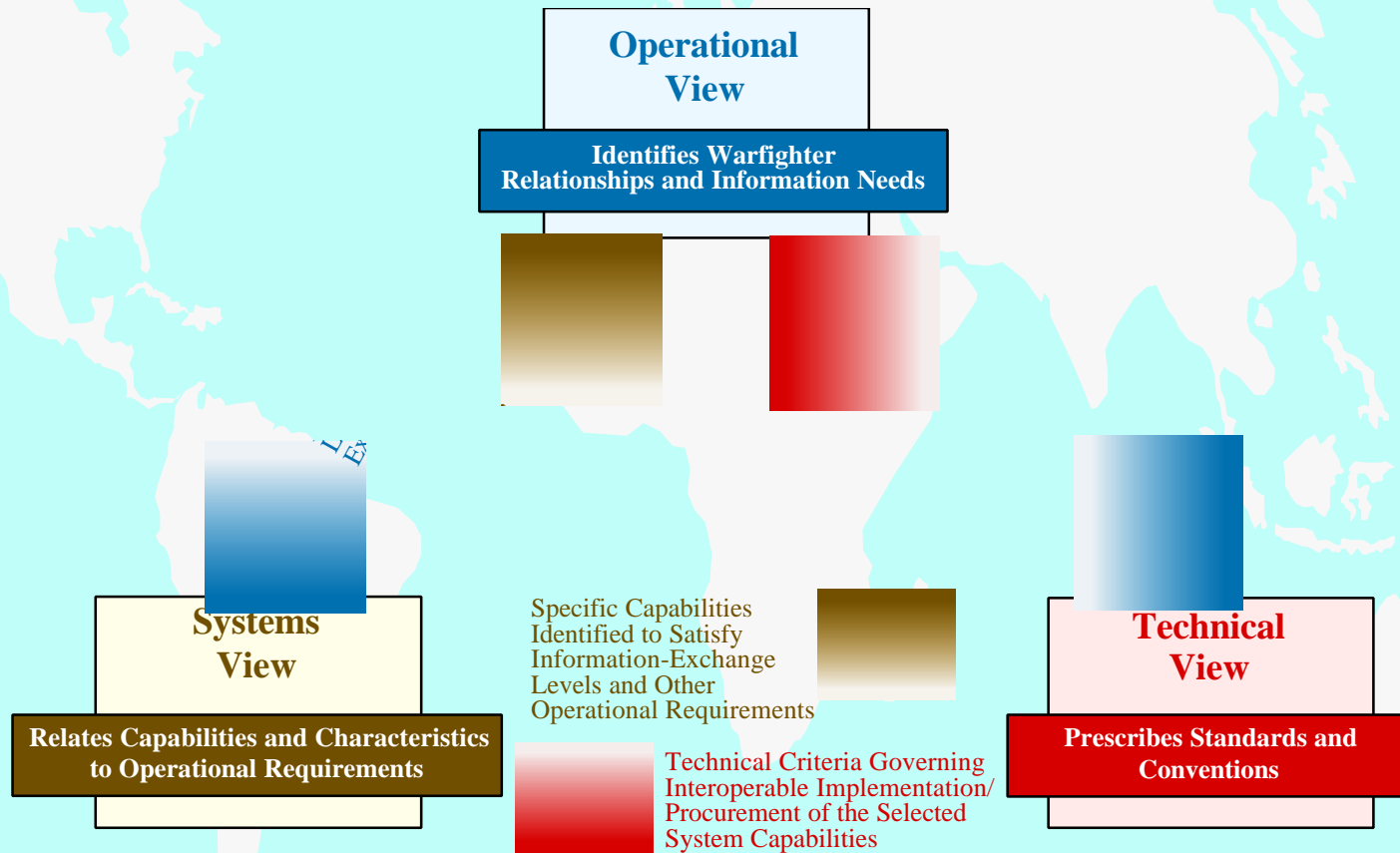
- ◆ The *Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework*
 - “... provides the rules, guidance, and product descriptions for developing and presenting architecture descriptions that ensure a common denominator for understanding, comparing and integrating architectures.”
 - OSD recently directed “that all on-going and, planned C4ISR or related architectures be developed in accordance with Version 2.0”

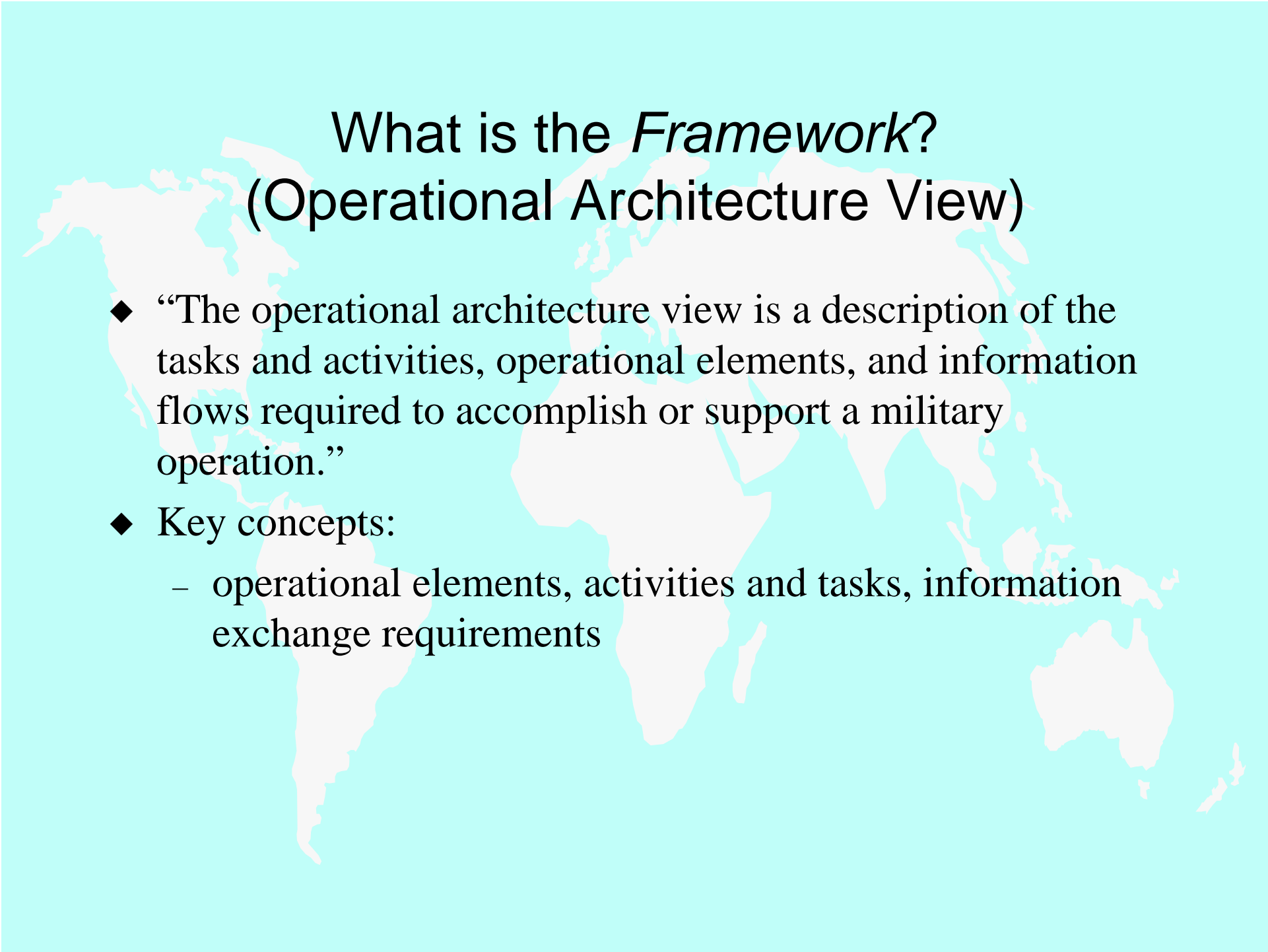


What is the *Framework*?

- ◆ “There are three major perspectives, i.e., views, that logically combine to describe an architecture ... the operational, systems and technical views.”
- ◆ All Framework quotations and examples in this briefing are taken from *Framework 2.0* document (available from <http://www.cisa.osd.mil>)

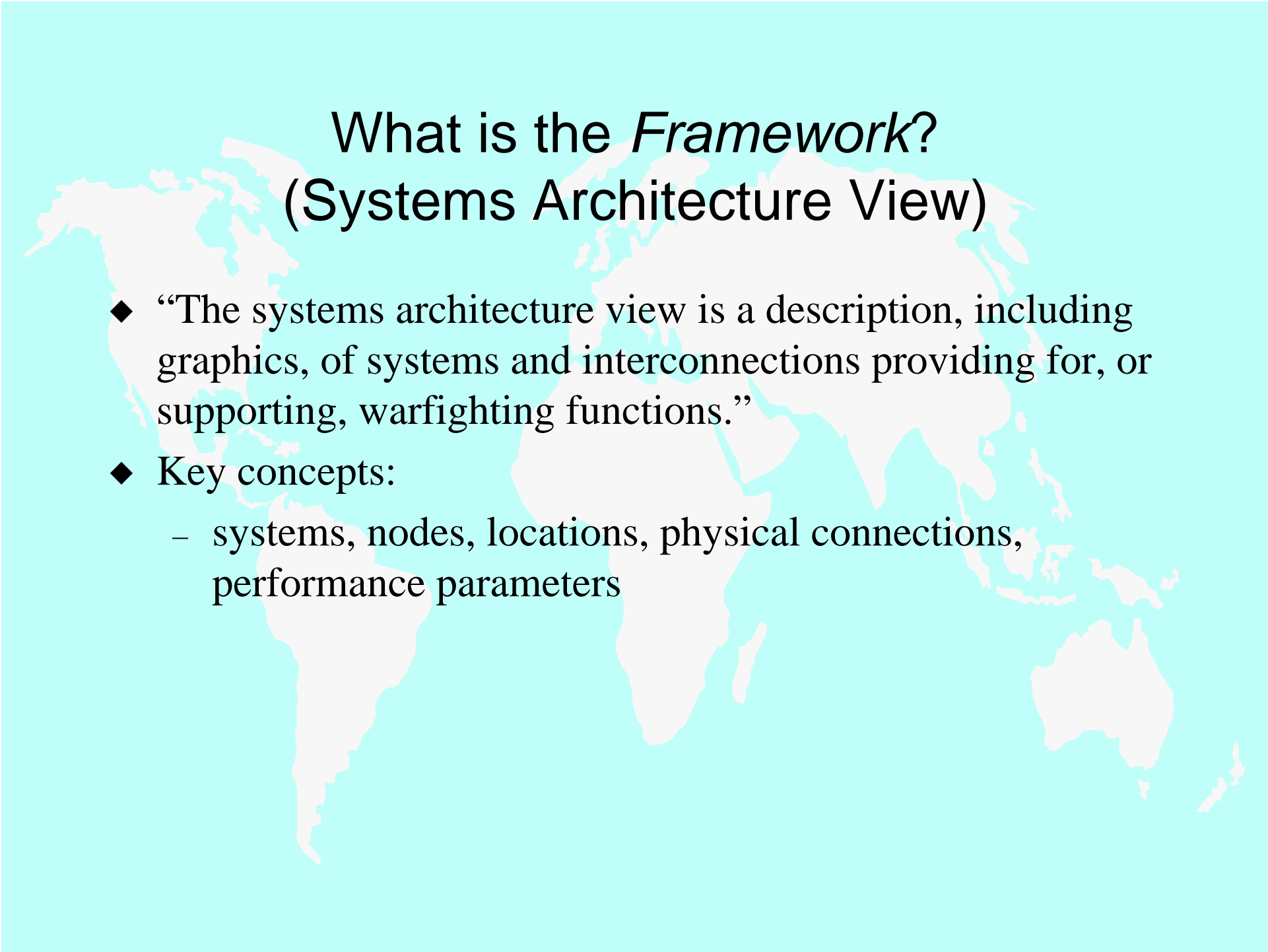
What is the *Framework*?





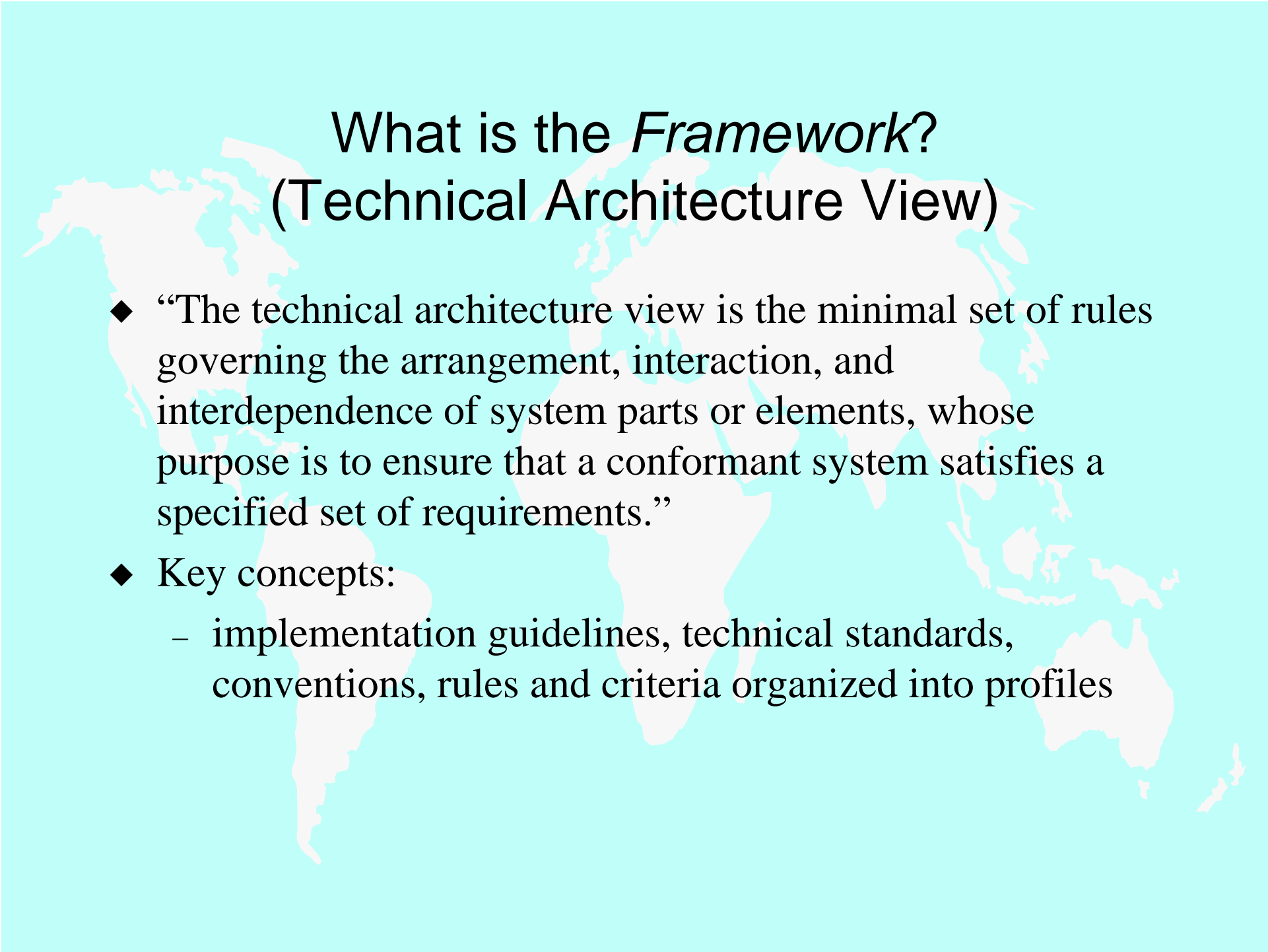
What is the *Framework*? (Operational Architecture View)

- ◆ “The operational architecture view is a description of the tasks and activities, operational elements, and information flows required to accomplish or support a military operation.”
- ◆ Key concepts:
 - operational elements, activities and tasks, information exchange requirements



What is the *Framework*? (Systems Architecture View)

- ◆ “The systems architecture view is a description, including graphics, of systems and interconnections providing for, or supporting, warfighting functions.”
- ◆ Key concepts:
 - systems, nodes, locations, physical connections, performance parameters



What is the *Framework*? (Technical Architecture View)

- ◆ “The technical architecture view is the minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements, whose purpose is to ensure that a conformant system satisfies a specified set of requirements.”
- ◆ Key concepts:
 - implementation guidelines, technical standards, conventions, rules and criteria organized into profiles

What is the *Framework*?

- ◆ The Framework identifies 26 architecture products
- ◆ 7 essential (i.e., required)
- ◆ 19 supporting (i.e., optional)

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
All Views (Context)	AV-1	<i>Overview and Summary Information</i>	Essential	Scope, purpose, intended users, environment depicted, analytical findings, if applicable (4.2.1.1)
All Views (Terms)	AV-2	<i>Integrated Dictionary</i>	Essential	Definitions of all terms used in all products (4.2.1.2)

Operational	OV-1	<i>High-level Operational Concept Graphic</i>	Essential	High-level graphical description of operational concept (high-level organizations, missions, geographic configuration, connectivity, etc.) (4.2.1.3)
Operational	OV-2	<i>Operational Node Connectivity Description</i>	Essential	Operational nodes, activities performed at each node, connectivities & information flow between nodes (4.2.1.4)
Operational	OV-3	<i>Operational Information Exchange Matrix</i>	Essential	Information exchanged between nodes and the relevant attributes of that exchange such as media, quality, quantity, and the level of interoperability required. (4.2.1.5)
Operational	OV-4	<i>Command Relationships Chart</i>	Supporting	Command, control, coordination relationships among organizations (4.2.2.1)
Operational	OV-5	<i>Activity Model</i>	Supporting	Activities, relationships among activities, I/Os, constraints (e.g., policy, guidance), and mechanisms that perform those activities. In addition to showing mechanisms, overlays can show other pertinent information. (4.2.2.2)
Operational	OV-6a	<i>Operational Rules Model</i>	Supporting	One of the three products used to describe operational activity sequence and timing that identifies the business rules that constrain the operation (4.2.2.3.1)
Operational	OV-6b	<i>Operational State Transition Description</i>	Supporting	One of the three products used to describe operational activity sequence and timing that identifies responses of a business process to events (4.2.2.3.2)
Operational	OV-6c	<i>Operational Event/Trace Description</i>	Supporting	One of the three products used to describe operational activity sequence and timing that traces the actions in a scenario or critical sequence of events (4.2.2.3.3)
Operational	OV-7	<i>Logical Data Model</i>	Supporting	Documentation of the data requirements and structural business process rules of the Operational View. (4.2.2.4)

Systems	SV-1	<i>System Interface Description</i>	Essential	Identification of systems and system components and their interfaces, within and between nodes (4.2.1.6)
Systems	SV-2	<i>Systems Communications Description</i>	Supporting	Physical nodes and their related communications laydowns (4.2.2.5)
Systems	SV-3	<i>Systems² Matrix</i>	Supporting	Relationships among systems in a given architecture; can be designed to show relationships of interest, e.g., system-type interfaces, planned vs. existing interfaces, etc. (4.2.2.6)
Systems	SV-4	<i>Systems Functionality Description</i>	Supporting	Functions performed by systems and the information flow among system functions (4.2.2.7)
Systems	SV-5	<i>Operational Activity to System Function Traceability Matrix</i>	Supporting	Mapping of system functions back to operational activities (4.2.2.8)
Systems	SV-6	<i>System Information Exchange Matrix</i>	Supporting	Detailing of information exchanges among system elements, applications and H/W allocated to system elements (4.2.2.9)
Systems	SV-7	<i>System Performance Parameters Matrix</i>	Supporting	Performance characteristics of each system(s) hardware and software elements, for the appropriate timeframe(s) (4.2.2.10)
Systems	SV-8	<i>System Evolution Description</i>	Supporting	Planned incremental steps toward migrating a suite of systems to a more efficient suite, or toward evolving a current system to a future implementation (4.2.2.11)
Systems	SV-9	<i>System Technology Forecast</i>	Supporting	Emerging technologies and software/hardware products that are expected to be available in a given set of timeframes, and that will affect future development of the architecture (4.2.2.12)
Systems	SV-10a	<i>Systems Rules Model</i>	Supporting	One of three products used to describe systems activity sequence and timing -- Constraints that are imposed on systems functionality due to some aspect of systems design or implementation (4.2.2.13.1)
Systems	SV-10b	<i>Systems State Transition Description</i>	Supporting	One of three products used to describe systems activity sequence and timing -- Responses of a system to events (4.2.2.13.2)
Systems	SV-10c	<i>Systems Event/Trace Description</i>	Supporting	One of three products used to describe systems activity sequence and timing -- System-specific refinements of critical sequences of events described in the operational view (4.2.2.13.3)
Systems	SV-11	<i>Physical Data Model</i>	Supporting	Physical implementation of the information of the Logical Data Model, e.g., message formats, file structures, physical schema (4.2.2.14)

Technical	TV-1	<i>Technical Architecture Profile</i>	Essential	Extraction of standards that apply to the given architecture (4.2.1.7)
Technical	TV-2	<i>Standards Technology Forecast</i>	Supporting	Description of emerging standards that are expected to apply to the given architecture, within an appropriate set of timeframes (4.2.2.15)

Issues with the Framework

- ◆ Do we need another MIL Standard?
 - in the face of recent DOD direction to use best commercial practices (such as IEEE, ISO, and de facto industry standards like UML)
 - Its products are summary in nature rather than “developmental”
 - against trend toward contractor-defined products
 - “... the Framework does not provide guidance in how to design or implement a specific architecture or how to develop and acquire systems-of-systems.”

Issues with the Framework

- ◆ What does it mean to conform to the Framework?
 - Produce all essential, applicable products
- ◆ No provisions in *Framework* for user comment, document revision and maintenance
- ◆ The authors haven't done their homework
 - The *Framework* ignores the substantial literature and emerging practice in architecture from the commercial world, industrial, research and standards communities
 - Mistakenly cites a definition of “architecture” as from IEEE (actually the definition appears to be from SEI)

Issues with the Framework (Conceptually)

- ◆ Everything isn't an architecture. As Mary Shaw put it:
 - “Let's not dilute the term architecture by applying it to everything in sight” (April 1995)
 - “Architecture” as used in the *Framework* is synonymous with “model”
- ◆ Does not distinguish problem and solution space orientations
- ◆ Systems Architecture View adopts a stove-piped mind set
- ◆ Operational-Systems-Technical split cuts across usual engineering partitions

How to fix it?

- ◆ Change the name to *C4ISR Modeling Framework*
- ◆ Make the notion of *view* more rigorous, following community practices (ISO, IEEE)
- ◆ Generalize “operational architecture view” to address multiple stakeholders of a DOD (or any system)
- ◆ Eliminate redundant architecture products, and allow best commercial practices in selection of notations and tools, managing at the viewpoint level

How does the Framework relate to Software Architecture?

- ◆ Software architecture “defines [a] system in terms of computational components and interactions among those components” (Shaw and Garlan)
- ◆ Key elements: components (clients, servers, databases, filters, layers, ...); connectors (procedure calls, shared variables, protocols, ...)



Issues with “Software Architecture”

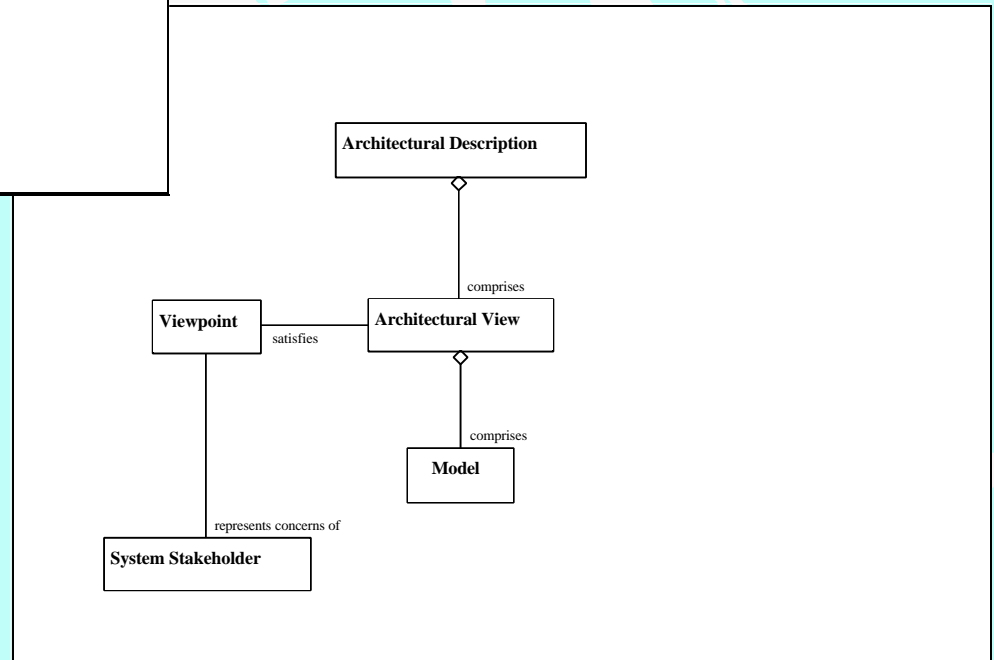
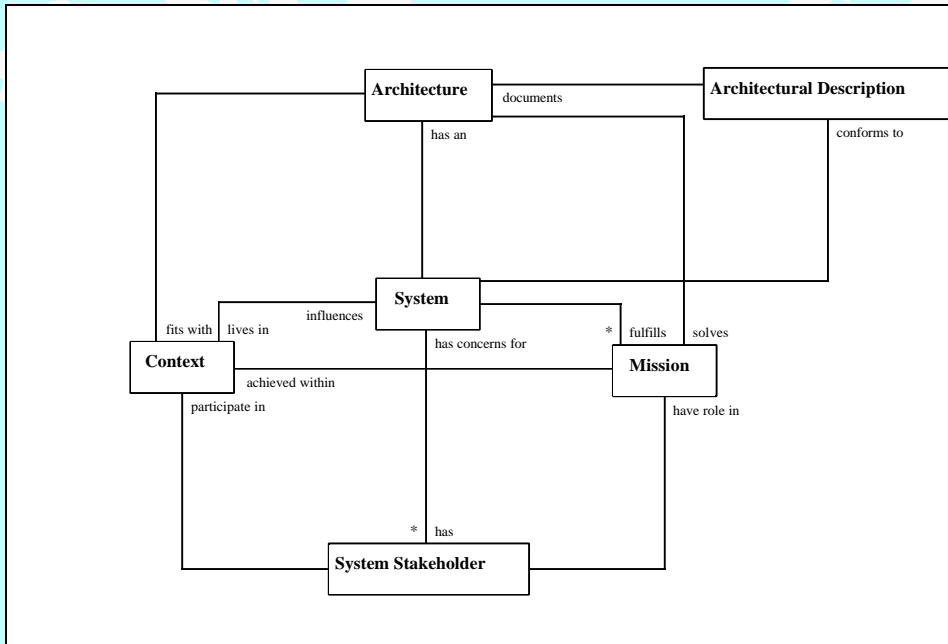
- ◆ Emphasis on software structure makes it difficult to deal with other system fundamentals, e.g.,
 - Distribution
 - Security
 - Behavior and dynamics



IEEE Architecture Working Group: Goals and Objectives

- ◆ Take a “wide scope” interpretation of architecture as applicable to software-intensive systems
- ◆ Establish a conceptual framework and vocabulary for talking about architectural issues of systems
- ◆ Identify and promulgate sound architectural practices
- ◆ Allow for the evolution of those practices as relevant technologies mature

Conceptual Framework



Example Architecture Viewpoints

Distribution View

Capability View

Production View

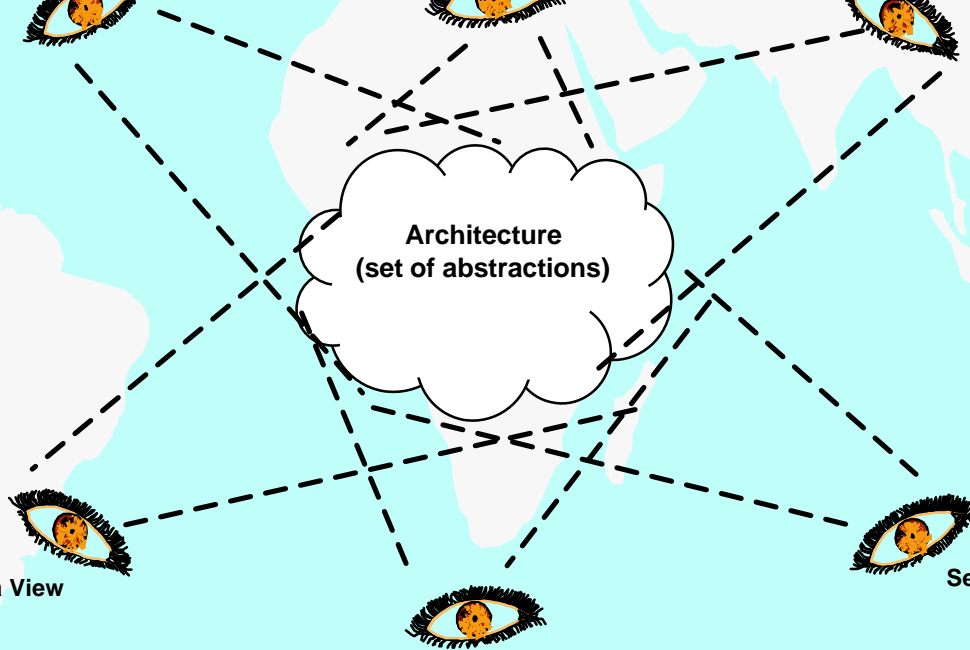


Architecture
(set of abstractions)

Data View

Communications View

Security View





Current Status and Plans

- ◆ Version 1.0 of P1471 *Recommended Practice*, has circulated to reviewers
- ◆ First ballot of P1471, October 1998
- ◆ Interested reviewers/participants contact:
 - Basil Sherlund (chair) **b.sherlund@computer.org**, or
 - **ieee-awg@spectre.mitre.org**
 - **<http://www.pithecanthropus.com/~awg>**