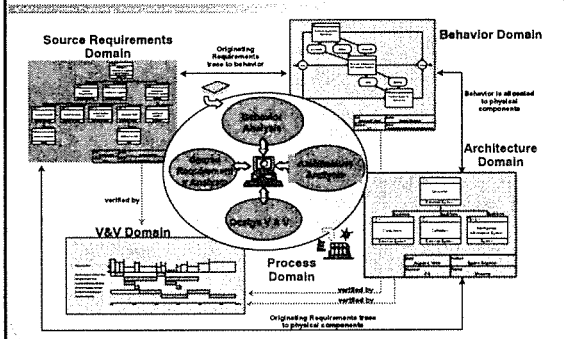
CORE 소개 : C4ISR

2003. 05.

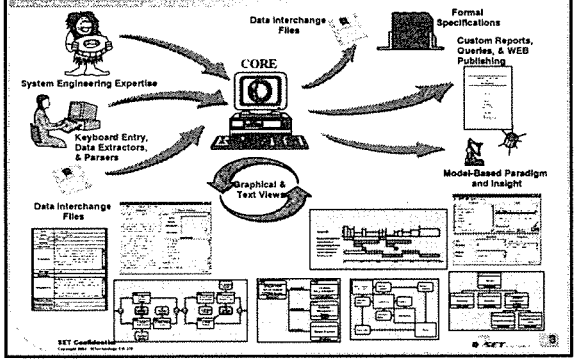
에스이테크놀로지㈜

440-860 경기도 종인시 백암면 고안리 633-2 사서함 25 고등기술연구원 9층
Tel. 031-330-7671-4 Fax. 031-330-7118

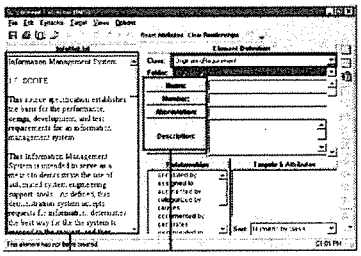
Systems Engineering Process and CORE



CORE Concept of Operation

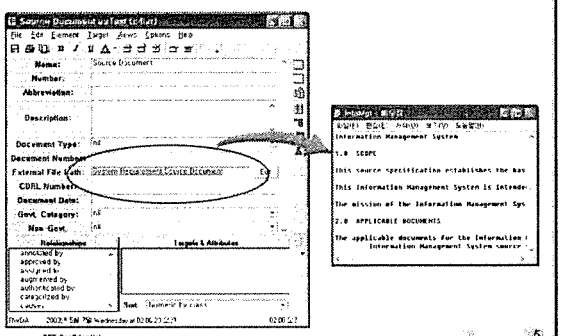


Capturing Originating Requirements via the Element Extractor

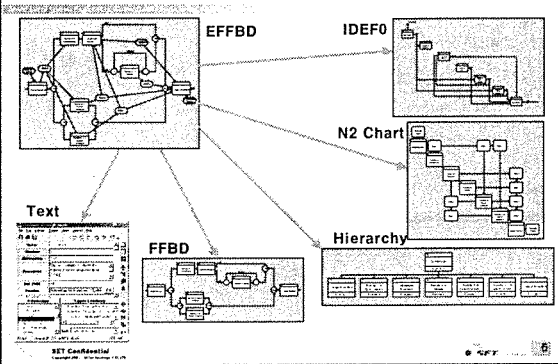


You Do Not Have to Retype Any Source Text.

External Reference

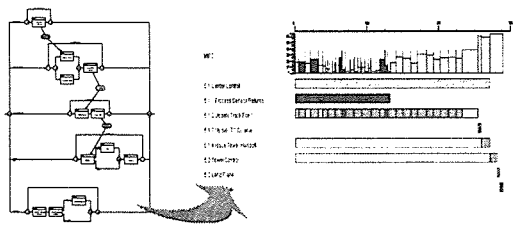


Many Faces of Behavior



Simulation

Quickly Identify Bottlenecks, and Resource Contention with COREsim



COREsim automatically generates the simulation model from the system functional model!

Automatic Report Generation

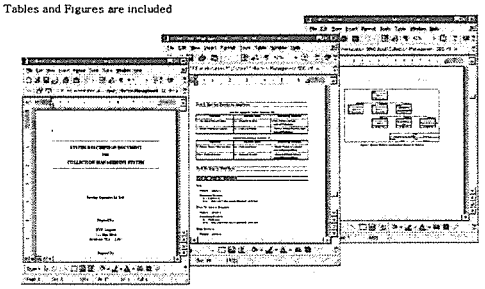
- Various types of reports (RTF, HTML, ASCII,CSV) are generated from database

System Description Document

Table of Contents

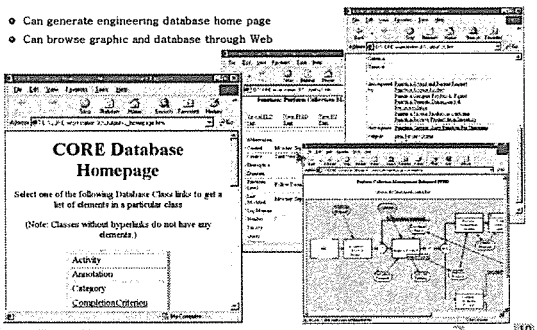
Customized Report Generation

- Can make organization specific report template
- Tables and Figures are included



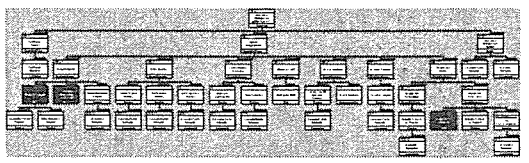
Enables Web-Based Communication

- Can generate engineering database home page
- Can browse graphic and database through Web

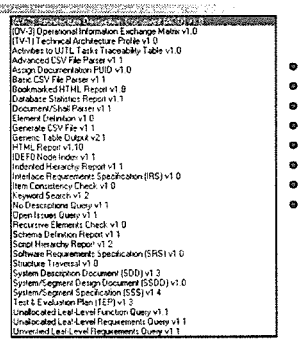


Provides Full Traceability

- Traceability of Requirement-Function-Component
- Traceability of Risk management, Configuration management, Engineering decisions are provided



Various Report Templates

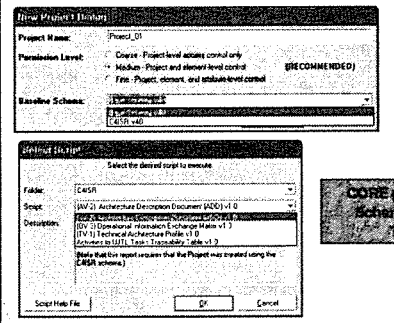


- System Description Document
- System/Segment Design Document
- System/Segment Specification
- Software Requirements Specification
- Test and Evaluation Plan
- Interface Requirements Specification
- Generic Table Output
- C4ISR View(AV, OV, TV)

Application Area

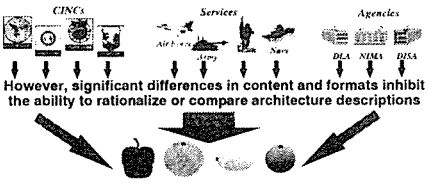
- Systems Engineering
- Business Process Reengineering
- Project Management
  - Can be linked with MS Project
- Validation and Verification
- Knowledge Management
- C4ISR

C4ISR Schema and Reports in CORE



Background: Why An Architecture Framework?

Military organizations are developing architectures that represent their contributions and relationships with respect to overall military operations



However, significant differences in content and formats inhibit the ability to rationalize or compare architecture descriptions

Disparate and unrelatable architecture products lead to non-integrated, non-interoperable, and non-cost effective capabilities in the field

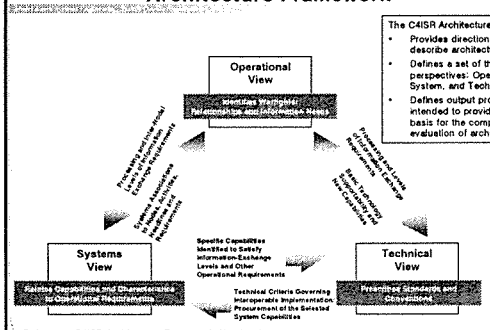
Reprinted from "C4ISR INCOSE Tutorial", A.H. Levis and L.W.Wagenhals, March 2001

The C4ISR Architecture Framework was Defined to Resolve the Problem

- Definition of C4ISR: Command, Control, Communication, Computers, Intelligence, Surveillance, and Reconnaissance.
- In 1998, DoD mandated the use of the C4ISR Architecture Framework for use on all ongoing and future architectures.
- C4ISR provides a common basis for comparison of architectures.
- C4ISR products:
  - Provide the means for determining interoperability.
  - Contribute to the development of the Operational Requirements Document (ORD).
  - Provide information exchange requirements (IERs) for system development.

The Three Perspectives of the C4ISR Architecture Framework

The C4ISR Architecture Framework:
- Provides direction on how to describe architectures:
- Defines a set of three perspectives: Operational, System, and Technical; and
- Defines output products intended to provide a common basis for the comparison and evaluation of architectures.



Reference: C4ISR Architecture Framework, Version 2.0

The Specified C4ISR Architecture Products

AV-1	Overview and Summary Information
AV-2	Integrated Dictionary
OV-1	High-level Operational Concept Graphic
OV-2	Operational Node Connectivity Description
OV-3	Operational Information Exchange Matrix
OV-4	Command Relationships Chart
OV-5	Activity Model
OV-6C	Operational Event/Trace Description
SV-1	System Interface Description
SV-2	Systems Communication Description
SV-3	Systems Matrix
SV-4	System Functionality Description
SV-5	Operational Activity to System Function Traceability Matrix
SV-6	System Information Exchange Matrix
SV-7	System Performance Parameters Matrix
SV-8	System Evolution Description
SV-9	System Technology Forecast
SV-10	System Activity Sequence & Timing
TV-1	Technical Architecture Profile
TV-2	Standards Technology Forecast

Note:

- **Blue indicates Essential products**
- **All other products are considered Supporting**

C4ISR Architecture Products and CORE

C4ISR Architecture Products		Product	CORE Equivalent
4.2.1	**Essential Framework Products**		
4.2.1.1	Overview and Summary Information	AV-1	Section 1 of ADD
4.2.1.2	Integrated Dictionary	AV-2	ADD Report
4.2.1.3	High-Level Operational Concept Graphic	OV-1	External Graphic
4.2.1.4	Operational Node Connectivity Description	OV-2	Physical Block Diagram for Operational Element
4.2.1.5	Operational Information Exchange Matrix	OV-3	OV-3 Report script
4.2.1.6	System Interface Description	SV-1	Physical Block Diagram for System/Component
4.2.1.7	Technical Architecture Profile	TV-1 (TV-2)	TV-1 Report script
4.2.2	**Supporting Framework Products**		
4.2.2.1	Command Relationships Chart	OV-4	Organizational Hierarchy with annotations
4.2.2.2	Activity Model	OV-5	IDEF0 & FFBD, EFFBD
4.2.2.3	Operational Activity Sequence and Timing Descriptions	OV-6a, 6b, 6c	CORE.sim output products
4.2.2.4	Logical Data Model	OV-7	External Graphic
4.2.2.5	Systems Communications Description	SV-2	Physical Block Diagram
4.2.2.6	Systems' Matrix	SV-3	Report script (future)
4.2.2.7	Systems Functionality Description	SV-4	EFFBD, Function Hierarchy, FFBD, IDEF0, N²
4.2.2.8	Operational Activity to System Function Traceability Matrix	SV-5	Generic Table
4.2.2.9	System Information Exchange Matrix	SV-6	Report script (future)
4.2.2.10	System Performance Parameters Matrix	SV-7	Report script (future)
4.2.2.11	System Evolution Description	SV-8	External Graphic
4.2.2.12	System Technology Forecast	SV-9	Report script (future)
4.2.2.13	System Activity Sequence and Timing Descriptions	SV-10a, 10b, 10c	CORE.sim output products
4.2.2.14	Physical Data Model	SV-11	Report script (future)
4.2.2.15	Standards Technology Forecast	TV-2 (TV-1)	TV-2 Report script

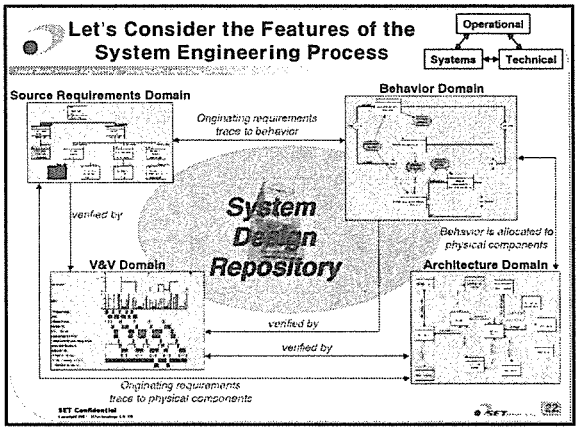
Problems

- C4ISR provides no guidance on designing or implementing a specific architecture
  - Comparing products developed by various teams is difficult.
  - Performing "engineering by viewgraph" is inadequate.
- Teams develop independent products and views:
  - Not really independent.
  - Generated without regard to flowing requirements to downstream activities.
  - Impose additional data and semantic interfaces.
- C4ISR views are outside the mainstream of established processes.
  - Views must be managed to maintain synchronization with the design process.
  - System development affects the architecture as refinement occurs.
- The need to maintain multiple, consistent, related views necessitates a database management system.

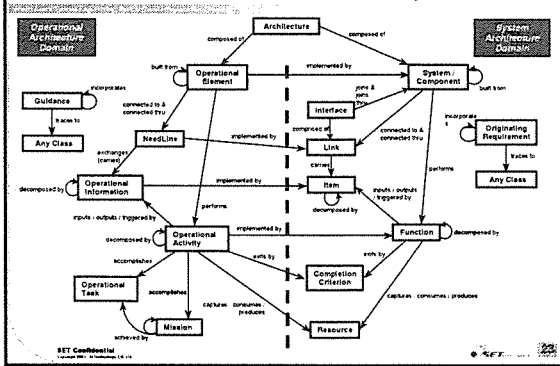
System Engineering as a Solution

- Apply systems engineering principles, processes, and comprehensive tools to C4ISR.
  - Capture the systems, operations, and design elements in a common data repository;
  - Unify products with a strong systems engineering process;
  - Incorporate simulation to evaluate, validate, and compare architecture capabilities.
- Expand the systems engineering environment to combine C4ISR and system engineering.
  - Extend the underlying systems engineering design language to:
    - Support development and specification of the specialized C4ISR architecture-specific products; and
    - Harmonize C4ISR and Systems Engineering views and processes.
- Exploit proven system engineering practices to develop systems and architectures.

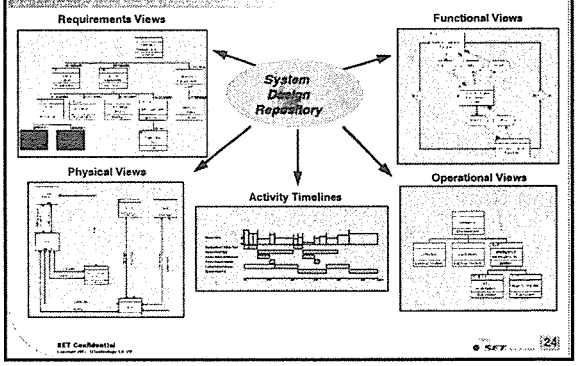
Let's Consider the Features of the System Engineering Process



An Integrated System Engineering and C4ISR Schema: Consistent and Compatible



Integrating the Repository and View Generators Provides Consistency

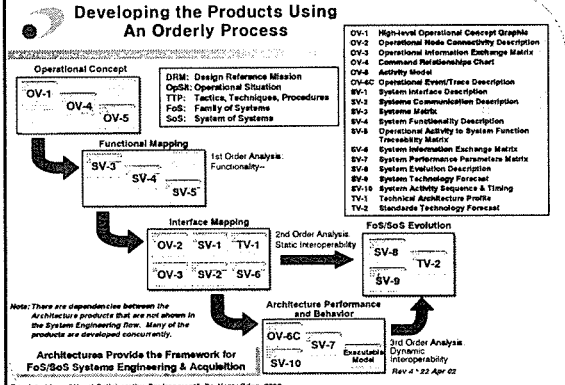


Producing C4ISR Products Automatically

C4ISR Architecture Products	Product	Automated Representation
Essential Framework Products		
Overview and Summary Information	AV-1	High level graphic of systems, environment, and users
Integrated Dictionary	AV-2	Document of all terms used in all products
High-Level Operational Concept Graphic	OV-1	High level graphic of operational concept with missions, organizations, geographic configuration, connectivity, etc.
Operational Node Connectivity Description	OV-2	High level graphic of nodes, activities, connectivities, etc.
Operational Information Exchange Matrix	OV-3	Matrix of info exchanged between nodes, media, quantity,
System Interface Description	SV-1	Graphic of systems and components and their interfaces
Technical Architecture Profile	TV-1	Standards that apply to a given architecture
Supporting Framework Products		
Command Relationships Chart	OV-4	Graphic of C and C and relationships between organizations
Activity Model	OV-5	Graphic of activities, relationships, I/O, cost, performs, nodes
Operational Activity Sequence and Timing Descriptions	OV-6a, 6b, &c	Graphic describing operational activity sequence and timing that identifies business rules, responses, and actions
Logical Data Model	OV-7	Graphic that documents data reqts. and bus. Process rules
Systems Communications Description	SV-2	Graphic of physical nodes and related comm. lincuts
Systems Matrix	SV-3	Matrix of relationships among systems in an architecture
Systems Functionality Description	SV-4	Graphic of functions performed by systems and info. flow
Operational Activity to System Function Traceability Matrix	SV-5	Matrix of mapping of system functions back to operational activities
System Information Exchange Matrix	SV-6	Matrix of details of data exchanges among elements
System Performance Parameters Matrix	SV-7	Performance of hardware and software elements
System Evolution Description	SV-8	Graphic of planned evolutionary improvement of a system
System Technology Forecast	SV-9	Prediction of applicable emerging technologies
System Activity Sequence and Timing Descriptions	SV-10a 10b, 10c	Graphic of system activity sequences and timing
Physical Data Model	SV-11	Physical implementation of message formats, structure...
Standards Technology Forecast	TV-2	Delineation of emerging standards expected to apply

- Generate specific graphical, matrix, and document views from the integrated environment.
- Ensure C4ISR views are consistent with each other as well as the current system design.

Developing the Products Using An Orderly Process



Note: There are dependencies between the Architecture products that are not shown in the System Engineering flow. Many of the products are developed concurrently.

Architectures Provide the Framework for FoS/SoS Systems Engineering & Acquisition

Rev A • 22 Apr 02

Reprinted from "Naval Collaborative Environment", Dr. Harry Crisp, 2002

Integrated Repository Approach for C4ISR has Benefits Across the Development Cycle

Independent	Integrated
Independent drawings	Consistent views
Static diagrams	Executable behavior
Data storage	Linked repository
Stored drawings	Dynamic view generation
Ad hoc process (inconsistent results)	Repeatable process (consistent results)
Manual change propagation across all affected products (by the systems engineer)	Automatic change propagation across all current and future products (by the engineering environment)

Effects of using the integrated repository, systems engineering/C4ISR approach vs. an independent, ad hoc C4ISR approach

The Integrated Environment Resolves the Data and Semantic Interface Problem Associated with Weaker Methodologies

Implementation of the integrated C4ISR and System Engineering processes provide:

- A repeatable and proven systems engineering methodology with over 30 years of successful application
- Integrated models for technical, operational, and system architectures
- A graphical notation to enhance the capture and representation necessary for communication and evaluation of candidate architectures
- Consistent C4ISR products produced directly from the data repository
- Executable models (simulation) for behavioral and performance analysis
- Support for the product life cycle

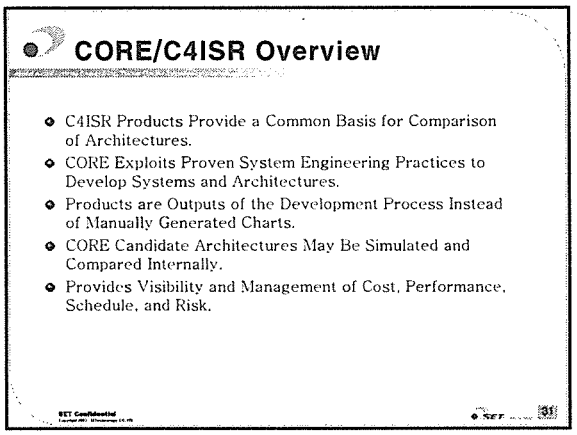
The system engineering environment is a powerful, yet flexible tool supporting the latest standards as well as providing a common repository in which to store, analyze, and produce engineering information in a collaborative environment. The application of systems engineering to C4ISR is a straightforward decision for the defining, specifying, and developing of successful systems. If one wants to take the effort from concept to operations, then C4ISR via system engineering is unquestionably the answer.

Applying CORE to C4ISR

- Methodology Has Been Used Successfully Since early 1970's
- Provides:
  - Integrated Repository (to support CADM)
  - Rigorous/Extensible Database Schema
  - Tailorable System Engineering Methodology
  - Views Automatically Generated from the Repository
  - Integrated and Consistent Views
  - Executable Models
  - Easy to Understand Graphical Notation

CORE/C4ISR Benefits

- **Supports the Complete System Engineering Process**
- **Integrated System Data Repository**
- **Consistency Guaranteed Between Views**
- **Views Tailored for Domain Experts**
- **Views Generated Directly from the Repository**
- **Simulations Facilitate Achieving Best Solution**
- **Tools and Process Easily Modified as Standards Evolve**


CORE/C4ISR Overview
C4ISR Products Provide a Common Basis for Comparison of Architectures.
CORE Exploits Proven System Engineering Practices to Develop Systems and Architectures.
Products are Outputs of the Development Process Instead of Manually Generated Charts.
CORE Candidate Architectures May Be Simulated and Compared Internally.
Provides Visibility and Management of Cost, Performance, Schedule, and Risk.