IBM Rational Suite® Support for the U.S. Department of Defense Enterprise Architecture/C4ISR Framework

Emmanuel Vergé
Technical Specialist
Development Tools Sales
IBM Software Group

Rational® software

A whitepaper from IBM Rational

Contents

Introduction

According to government research firm Input, the U.S. government is expected to have spent $1 billion on enterprise architecture (EA) efforts in 2003 and will spend another $1 billion in 2004. The federal government has made EA a part of the budget approval process. Agencies that cannot demonstrate how their enterprise architecture will accommodate their proposed IT investments will not be funded. Despite the mandate, enterprise architecture is becoming recognized as a sound business practice to align IT with an organization's mission, mitigate risk and deploy resources in the most cost-effective, efficient manner.

Transitioning to an effective EA requires a variety of tools and processes, and IBM Rational® tools and best practices can play a leading role in supporting this transition. This paper provides an overview of enterprise architectures, then offers an example of EA implementation using the *C4ISR* (Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance) Architecture Framework. In the context of U.S. Department of Defense (DoD) activities, EAs usually involve technical systems—such as remote control and telecommunications systems shared by satellites, aircraft, missiles, destroyers and so on. The underlying complexity of building such complex systems must therefore be entrusted to the Systems Engineering discipline. This white paper discusses how IBM Rational can help systems engineers utilize the IBM Rational Unified Process® methodology for Systems Engineering (RUP® SE) framework. It also examines the benefits of using RUP SE with the Unified Modeling Language (UML) and maps an EA under development to the DoD Architecture Framework artifacts.

Background on EA frameworks

The emergence of U.S. federal IT recommendations and guidance over the last few years is owed largely to Congress's passage of the Clinger-Cohen Act in 1996.[1] This act requires chief information officers (CIOs) of major departments and agencies to develop, maintain and facilitate the implementation of IT architectures as a means of integrating business processes and agency goals with IT. Several past experiences with federal

[1] For more information, visit http://wwwoirm.nih.gov/policy/itmra.html and http://www.cio.noaa.gov/itmanagement/cohenact.htm.

agencies[2] had shown that attempts to modernize IT environments without blueprints—models that simplify the complexities of how agencies operate today and how they want to evolve in the future—often resulted in uncontrolled investments and systems that were duplicative, ineffective or even nonoperational. EAs offer such blueprints.

The concept of EA dates back to the mid-1980s. At that time, John Zachman identified the need to use a logical construction blueprint—that is, an architecture—for defining and controlling the integration of systems and their components. In an article published in the *IBM Systems Journal* in 1987,[3] Zachman described a "framework" or structure for logically defining and capturing a computing architecture. Drawing parallels to the field of classical architecture and, later, to the aircraft manufacturing industry in which different work products (for example, architect plans, contractor plans, shop plans and bills of lading) represent different views of the planned building or aircraft, Zachman's framework identified the kind of work products needed to understand and thus build a given system or entity.

Since the late 1980s, architecture frameworks have emerged within the U.S. federal government, beginning with several federal entities issuing their own EA frameworks, including the DoD, the Department of the Treasury and the Federal CIO Council. Although the various frameworks use different terminology and somewhat different structures, they are fundamentally consistent in purpose and content, and they are being used today to varying degrees by many federal agencies. The emergence of federal frameworks occurred along with the creation of several general EA guidance documents addressing the "how" of EA enforcement—that is, how an organization should assess whether its proposed IT investments are compliant with its EA.

Typically, EA guidance targets development, maintenance and implementation, depicting in practical terms an end-to-end set of steps for managing an EA program. More specifically, these EA guides explain how to get started and organized, what kind of management controls are needed, what factors to consider in formulating an EA development approach and how to define the current and target architecture along with the plan for sequencing from the current system to the target. Noticeably, these guides strongly recommend use of commercial off-the-shelf (COTS) software under

[2] General Accounting Office, "Enterprise Architecture Use across the Federal Government Can Be Improved," February 2002, GAO-02-6.

[3] J.A. Zachman, "A Framework for Information Systems Architecture," *IBM Systems Journal*, vol. 26(3), 1987.

the classic principle of "not reinventing the wheel." Because EA deals with IT and business processes, these guides also represent the human factor, recommending job-roles definition as well as systematic architecture refresh and maintenance to ensure its currency and relevance.

From DoD enterprise architecture to systems engineering with UML

IBM views EA as essential tools—or, *work products*—for effectively and efficiently engineering business processes and for implementing and evolving the systems that support those processes. More specifically, EAs are descriptions derived and captured systematically—in useful models, diagrams and text format—of the mode of operation for a given enterprise. Modes of operation include (1) a single organization or (2) a functional or mission area that transcends more than one organization (for example, network centric warfare systems, financial management, air traffic control systems and federal agencies' logistics management). The architecture describes the enterprise's operations in both *logical terms*, such as interrelated business processes and business rules, information needs and flows, work locations or users, and *technical terms*, such as hardware, software, data, communications and security or performance attributes. It provides these perspectives both for the enterprise's current environment and for its future environment as well as a transition plan for moving from the current to the future environment.

In a report published in July 2002, Lockheed Martin Mission Systems states in its "Lessons Learned" section that:

> The domain of Systems Engineering (SE) and the essential elements of an Enterprise Architecture Framework (EAF) are tightly coupled:
> - SE artifacts (information) [are] basically leveraged to provide the basis for a "system architecture"
> - Good SE and producing the EAF are synonymous

In addition, SE has become synonymous with building systems that are cost-effective and most likely will meet users' real (as opposed to perceived) needs.[4]

---

[4] Abe Meilich, Ph.D., C.C.P (abraham.w.meilich@lmco.com), "Applying Tools and Methodologies to Develop C4ISR Architecture Framework Compliant Architecture Products," Lockheed Martin Mission Systems - Defense Information Systems, July 17, 2002, http://www.incose.org/chesapek/meetings/Tools_and_Methodologies_for_C4ISR.pdf.

When it comes to large-scale systems development, today's new challenges are clearly identified by system architects. Increased flexibility in software systems is actually allowing new system capabilities. Services can be provided by hardware, people or—at an ever-increasing rate—software. This has increased the complexity of the environment that business analysts, project managers and systems engineers are working in; and decisions about how a system capability is going to be realized must be made much earlier in the development life cycle. Sometimes, however, these design decisions must be delayed to facilitate design optimization.[5] According to the Standish Group, in 1995 the U.S. government spent nearly $81 billion on canceled projects. These projects and programs are failing primarily because of their systems development approach. This has led organizations to reevaluate their systems development capability and look to standards such as the Software Engineering Institute's Capability Maturity Model Integration (CMMI®) and ISO to provide benchmarks of their capability. But use of these standards alone is insufficient.

For multiple observable reasons (for example, budget cuts and the prevalence in today's projects of COTS as opposed to in-house products) and because of the emergence of recognized and proven modern development practices, large organizations have started to consider systems development more systematically, thus the heightened interest in EA. The introduction of iterative and agile methods for developing systems and the need for better risk management, pragmatic prediction or improved team organization are the essential drivers for such a change.

As thoroughly described by IBM Distinguished Engineer Murray Cantor,[6] the classic definition of a system is an assemblage of hardware, software and worker roles that collectively meets a business purpose. Many software developers are de facto system developers, because nearly all software tools—such as e-business applications, enterprise data integration initiatives and embedded software—contain some elements of a system's problem. But systems engineering is not simply software engineering with extra

---

[5] For more information, see Dave West's article in *The Rational Edge*: http://www-106.ibm.com/developerworks/rational/library/content/RationalEdge/jan03/TheRationalEdge_Jan2003.pdf.

[6] For more information, see Murray Cantor's previous articles in *The Rational Edge*: http://www-106.ibm.com/developerworks/rational/library/content/RationalEdge/nov01/TheRationalEdgeNovember01Issue.pdf, http://www-106.ibm.com/developerworks/rational/library/content/RationalEdge/dec01/TheRationalEdge200112Issue.pdf, http://www-106.ibm.com/developerworks/rational/library/content/RationalEdge/jul03/TheRationalEdge_July2003.pdf.

requirements. A systems engineering development process requires a set of activities to help define the system's architectural elements and requirements. Hence the need for a process framework such as the IBM Rational Unified Process and its extension RUP SE, which provides the guidance and templates necessary to support these systems-related activities. In addition to pure systems engineering projects, RUP SE addresses the needs of projects that:

- Entail concurrent hardware and software development
- Are large enough to require more than one development team
- Present architecturally significant deployment issues, such as a need for heterogeneous, distributed and geographically dispersed hosting platforms

As a fully defined process framework, RUP SE helps to unify the entire system design and development team, and enhances communication and collaboration among team members—which, in essence, does not differ from what EA frameworks are intended to do.[7]

Considering its numerous successes, the use of UML within RUP represents a key best practice that corroborates the need for better systems development.[8] UML is a visual modeling language developed to combat communication difficulties in the software development process. Soon after its initial release, UML gained the attention of the software development community, which realized its potential for much more than visual descriptions of software. In fact, its applicability and ease of use extends to many domains, including business modeling, data modeling and system modeling. It is well suited for developing precise and complete visual descriptions of the elements of an EA, largely because of the availability of UML-based graphical modeling tools that teams can use to create and maintain the architecture and its semantics in a central, browsable framework. Today, UML is a standard specification language maintained by the Object Management Group.

[7] RUP SE has been successfully deployed at Lockheed Martin for the future Aegis combat system—presentation at Rational User Conference 2002 Session MSP06.

[8] Source: http://www-306.ibm.com/software/awdtools/rup

C4ISR/DoDAF: A DoD enterprise architecture initiative

This white paper examines an example EA framework implementation: the initial version of the DoD Architectural Framework (DoDAF), known as C4ISR. This framework focuses on enabling the interoperation of disparate systems used to support military missions. Its purpose is to clearly define the objectives of the DoD's IT in accordance with its core mission. The **C4ISR/DoDAF** provides specific support for **Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance**, and it affords the DoD with several important capabilities:

- Allows the DoD to create systems that work effectively together
- Enables owners and investors to make intelligent buy, build and reuse decisions
- Helps system architects make good design decisions about where functionality should reside within the system's architecture
- Provides a common language to describe an integrated system that spans missions
- Allows large groups of stakeholders to work cooperatively and effectively on large, complex systems

In 1998, the DoD mandated the C4ISR Architecture Framework for all ongoing and future contracted architectures.

To achieve the *dominant battle space* awareness described in Joint Vision 2010,[9] the Assistant Secretary of Defense in 1995 launched an initiative to define a coordinated approach—that is, a framework for C4ISR architecture development, presentation and integration. The C4ISR Architecture Framework is intended to ensure that the architectures developed by the geographic and functional unified commands, military services and defense agencies interrelate between and among the organizations' operational, systems and technical architecture views. These architectures must be comparable and must integrate across joint and multinational organizational boundaries. A generic term is often used to refer to such an environment: *network centric warfare* (see Figure 1). It must be noted that C4ISR leverages

[9] Source: http://www.army.mil/public/mission_vision.htm

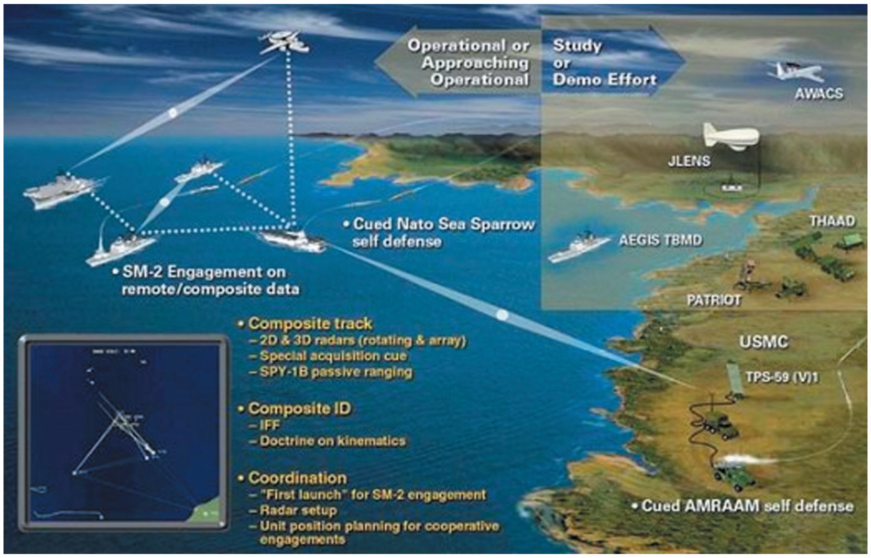


Figure 1: *DoD Network Centric Warfare illustration*

prior EA efforts from the Navy, the Air Force, the Marine Corps, the Defense Intelligence Agency and the Defense Information Systems Agency.

The C4ISR Architecture Framework[10] is the first instance of the Federal Architecture Framework to be extensively adopted and implemented throughout the DoD for all new systems. Represented as 27 work products that capture information, or *views*, about the architecture (see Table 1), the C4ISR framework provides an exhaustive blueprint for any DoD project.

[10] The C4ISR/DoDAF, in its present form (Version 2.0), can be found at http://www.c3i.osd.mil.

Table 1: *Essential and supporting C4ISR Architecture Framework work products*

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
All Views (Context)	AV-1	Overview and Summary Information	Essential	Scope, purpose, intended users, environment depicted and analytical findings, if applicable (4.2.1.1)
All Views (Terms)	AV-2	Integrated Dictionary	Essential	Definitions of all terms used in all products (4.2.1.2)
Operational	OV-1	High-level Operational Concept Graphic	Essential	High-level graphical description of operational concept (high-level organizations, missions, geographic configuration, connectivity and so on) (4.2.1.3)
Operational	OV-2	Operational Node Connectivity Description	Essential	Operational nodes, activities performed at each node, connectivities and information flow between nodes (4.2.1.4)
Operational	OV-3	Operational Information Exchange Matrix	Essential	Information exchanged between nodes and the relevant attributes of those exchanges such as media, quality, quantity and the level of interoperability required (4.2.1.5)
Operational	OV-4	Command Relationships Chart	Supporting	Command, control and coordination relationships between organizations (4.2.2.1)
Operational	OV-5	Activity Model	Supporting	Activities, relationships among activities, I/Os, constraints (for example, policy and guidance) and mechanisms that perform the activities; in addition to showing mechanisms, overlays can show other pertinent information (4.2.2.2)
Operational	OV-6a	Operational Rules Model	Supporting	In addition to showing mechanisms, overlays can show other pertinent information (4.2.2.2); 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	Operational State Transition Description	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	Operational Event/Trace Description	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	Logical Data Model	Supporting	Documentation of the data requirements and structural business process rules of the operational view. (4.2.2.4)

***Table 1**: Essential and supporting C4ISR Architecture Framework work products, continued*

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Systems	SV-1	System Interface Description	Essential	Identification of systems and system components and their interfaces, within and between nodes (4.2.1.6)
Systems	SV-2	Systems Communications Description	Supporting	Physical nodes and their related communications laydowns (4.2.2.5)
Systems	SV-3	Systems Matrix	Supporting	Relationships among systems in a given architecture; can be designed to show relationships of interest—for example, system-type interfaces or planned versus existing interfaces (4.2.2.6)
Systems	SV-4	Systems Functionality Description	Supporting	Functions performed by systems and the information flow among system functions (4.2.2.7)
Systems	SV-5	Operational Activity to System Function Traceability Matrix	Supporting	Mapping of system functions back to operational activities (4.2.2.8)
Systems	SV-6	System Information Exchange Matrix	Supporting	Detailing of information exchanges among system elements, applications and hardware allocated to system elements (4.2.2.9)
Systems	SV-7	System Performance Parameters Matrix	Supporting	Performance characteristics of each systems' hardware and software elements, for the appropriate timeframe(s) (4.2.2.10); planned incremental steps toward migrating a suite of systems to a more efficient suite
Systems	SV-8	System Evolution Description	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	System Technology Forecast	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	Systems Rules Model	Supporting	One of three products used to describe systems activity sequence and timing — constraints that are imposed on systems functionality because of system design or implementation (4.2.2.13.1)

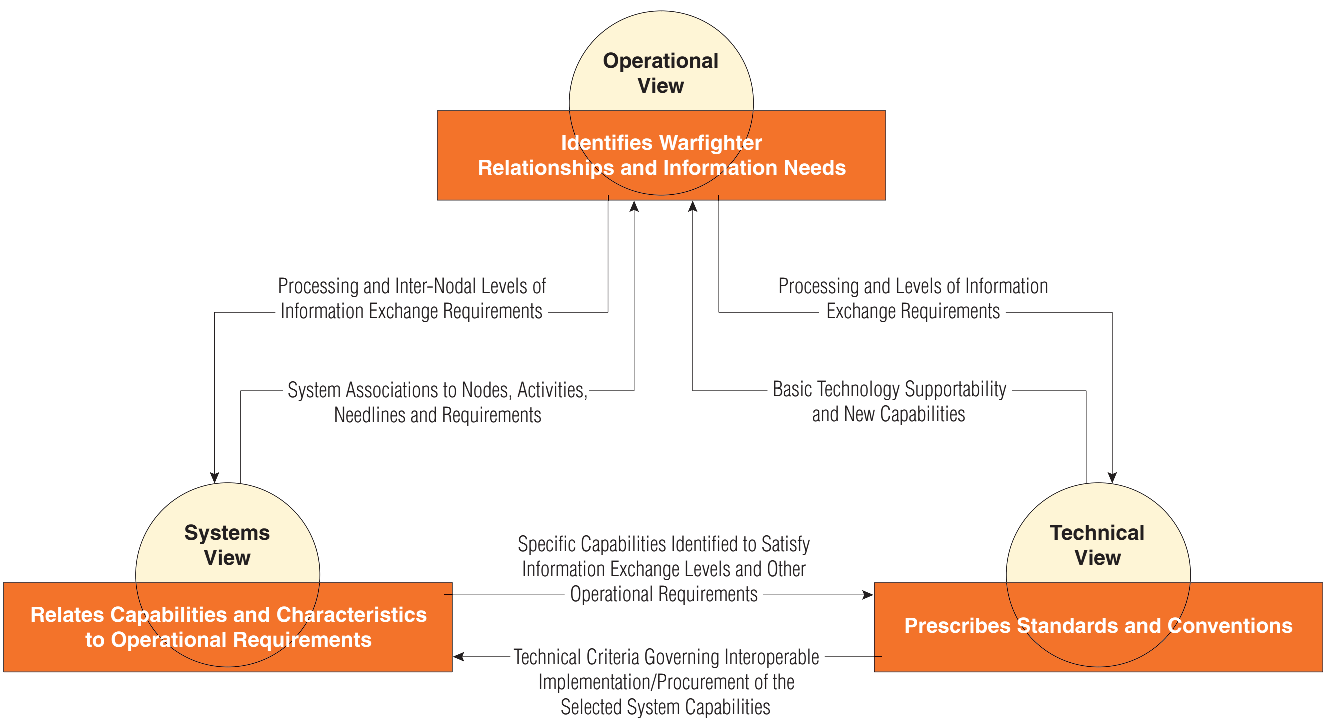
***Table 1**: Essential and supporting C4ISR Architecture Framework work products, continued*

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Systems	SV-10b	Systems State Transition Description	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	Systems Event/Trace Description	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	Physical Data Model	Supporting	Physical implementation of the information of the logical data model—for example, message formats, file structures and physical schema (4.2.2.14)
Technical	TV-1	Operational Activity to System Function Traceability Matrix	Essential	Mapping of system functions back to operational activities (4.2.2.8)
Technical	TV-2	Standards Technology Forecast	Supporting	Description of emerging standards that are expected to apply to the given architecture, within an appropriate set of timeframes (4.2.2.15)

Architecture views: Definitions, roles and linkages

In the context of C4ISR architectures, system architecture is represented as views. These views are expected to address the full range of systems—from sensors that collect information and pass it on, to processing and information systems, communications systems and shooters that require information to accomplish their objectives. **Systems architecture views** depict the functional and physical automated systems, nodes, platforms, communications paths and other critical elements that support information-exchange requirements and "warfighter tasks" described in the **operational architecture views**. Various attributes of the systems, nodes and required information exchanges are included according to the purpose of the specific architecture. In Figure 2, the linkages among the three main architecture views are represented with the dependency links between them.

Figure 2: *Fundamental linkages between C4ISR architecture views*



The operational architecture view
The operational architecture view[11] describes the tasks and activities of concern and the information exchanges required. These kinds of descriptions are useful for facilitating several actions and assessments across DoD, such as examining business processes for reengineering or technology insertion, training personnel, examining doctrinal and policy implications, coordinating joint and multinational relationships and defining the operational requirements to be supported by physical resources and systems (for example, communications throughput, specific node-to-node interoperability levels, information transaction time windows and required security protection).

The systems architecture view
The systems architecture view describes the systems of concern and the connections among those systems in context with the operational architecture view. The systems architecture view may be used for several purposes, including base-lining systems, making investment decisions concerning cost-effective ways to satisfy operational requirements and evaluating interoperability improvements. A systems architecture view addresses specific technologies and "systems." These technologies can be existing, emerging, planned or conceptual, depending on the purpose that the architecture effort is trying to facilitate (for example, reflection of the "as-is" state, transition to a "to-be" state or analysis of future investment strategies).

The technical architecture view
The technical architecture view describes a profile of a minimal set of time-phased standards and rules governing the implementation, arrangement, interaction and interdependence of system elements. The appropriate use of the technical architecture view is to promote efficiency and interoperability, and to ensure that developers can adequately plan for evolution.

Once the three architecture views have been fully described through the realization of work products, the C4ISR framework is considered complete.

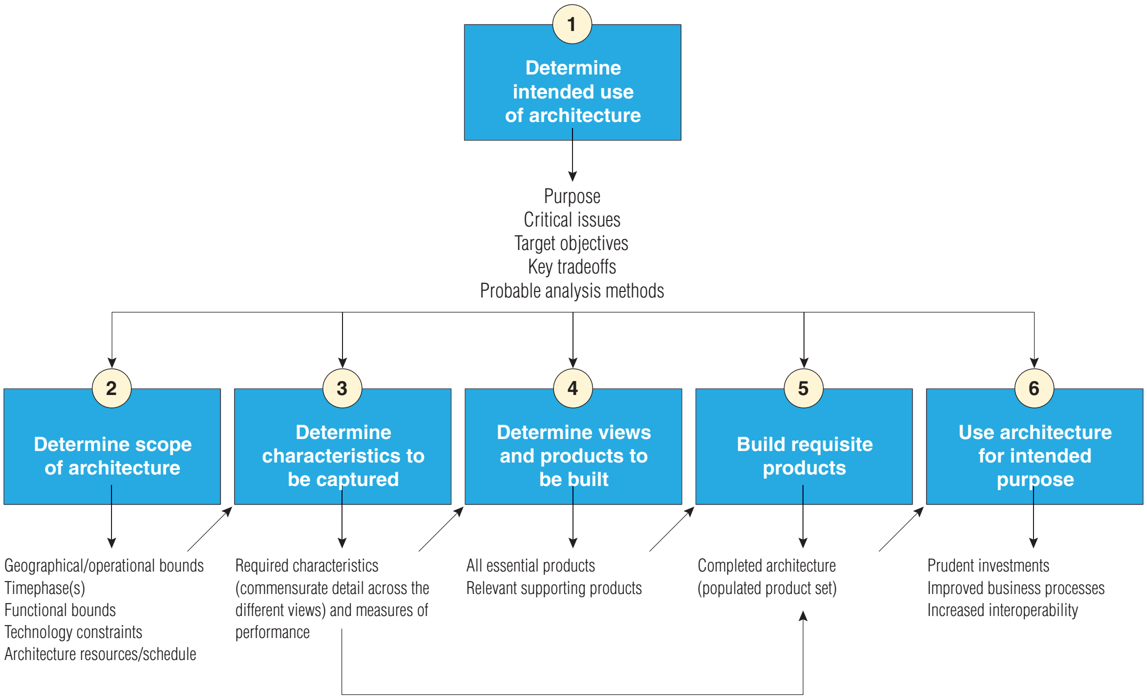
[11] Definitions are available in the C4ISR/DoDAF reference document.

Support for C4ISR using IBM Rational Suite: A full life-cycle solution to support enterprise software projects from requirements to release

The C4ISR/DoDAF provides not only the description of *what* should be accomplished, but also the relevant means to successfully implement the EA model. Although the C4ISR/DoDAF guidance is not prescriptive in terms of tools and processes to be used on a daily basis, some general recommendations and best practices are depicted. The C4ISR Architecture Framework contains four main types of guidance for the architecture development process: (1) guidelines, which include a set of guiding principles and guidance for building architectures that are compliant with the framework; (2) a basic process for using the framework to build and integrate architectures; (3) a discussion of architecture, data and tools that can serve as facilitators of the architecture-description process; and (4) a detailed description of the product types.

A general diagram in the DoDAF reference document, shown in Figure 3, illustrates the six steps for building the C4ISR framework.

Figure 3: *Six steps in building a C4ISR architecture*



C4ISR architecture products

As explained in the beginning of this white paper, the architecture products—the C4ISR work products—are those graphical, textual and tabular items that are developed in the course of building a given architecture description and that describe pertinent characteristics. When completed, this set of products constitutes the architecture description. These architecture products are different from the preexisting information sources that might have been used previously in building architectures, such as existing architectural models and technical reference models. Applicable extracts from these sources may be used in the architecture description itself as portions of products, and the completed architecture becomes an information source for other upcoming efforts.

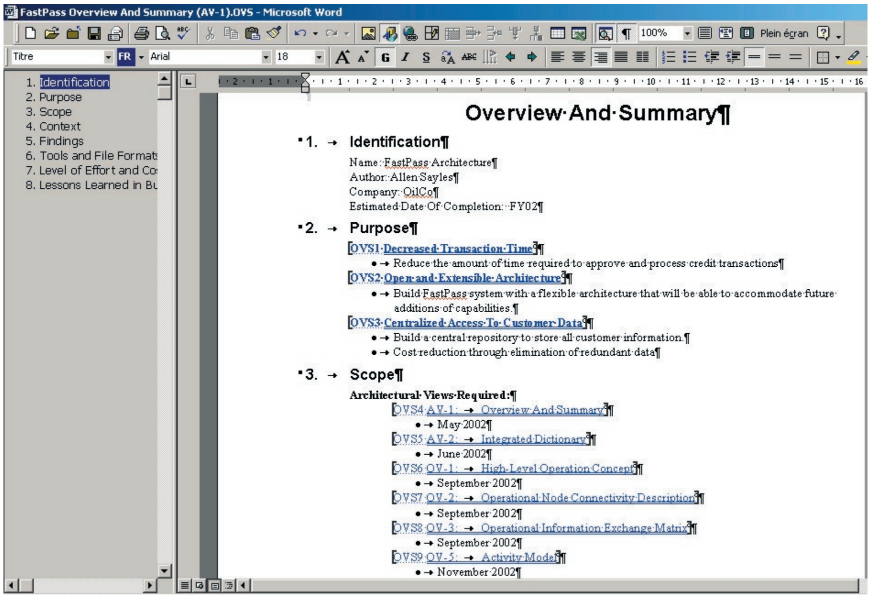
IBM Rational Suite support for C4ISR

Here is an exhaustive list of the C4ISR product references along with highlights and illustrations of IBM Rational Suite® support for these C4ISR framework products.

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
All Views (Context)	AV-1	Overview and Summary Information	Essential	Scope, purpose, intended users, environment depicted and analytical findings, if applicable (4.2.1.1)
• Serves as an initial planning guide for architecture development • Provides summary information concerning who, what, where, when and how				

IBM Rational Suite Support

Rational Suite supports formatting and enforcement of the Summary Information Directions through the Inception phase activity (see IBM Rational Unified Process) described in software development plan. Such guidance is provided within RUP and RUP SE.

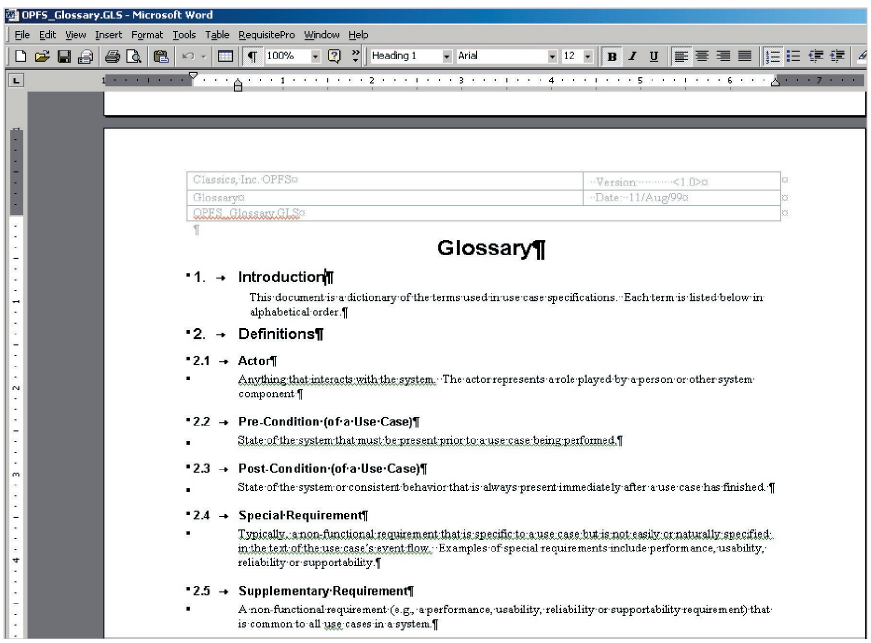


Template documents within IBM Rational Suite (in this screen shot, Microsoft® Word with IBM Rational RequisitePro)

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
All Views (Terms)	AV-2	Integrated Dictionary	Essential	Definitions of all terms used in all products (4.2.1.2)
• Provides a central location for all definitions and metadata • Allows architecture products to stand alone				

IBM Rational Suite Support

Glossaries and data dictionaries are key artifacts in the Rational Unified Process and can be created and maintained in IBM Rational Suite through IBM Rational RequisitePro® software, an integrated product for requirements and use-case management. The dictionary is supported and, in fact, can be automatically generated from the work the team has done while producing the model. The dictionary can be created and maintained in Rational Suite.

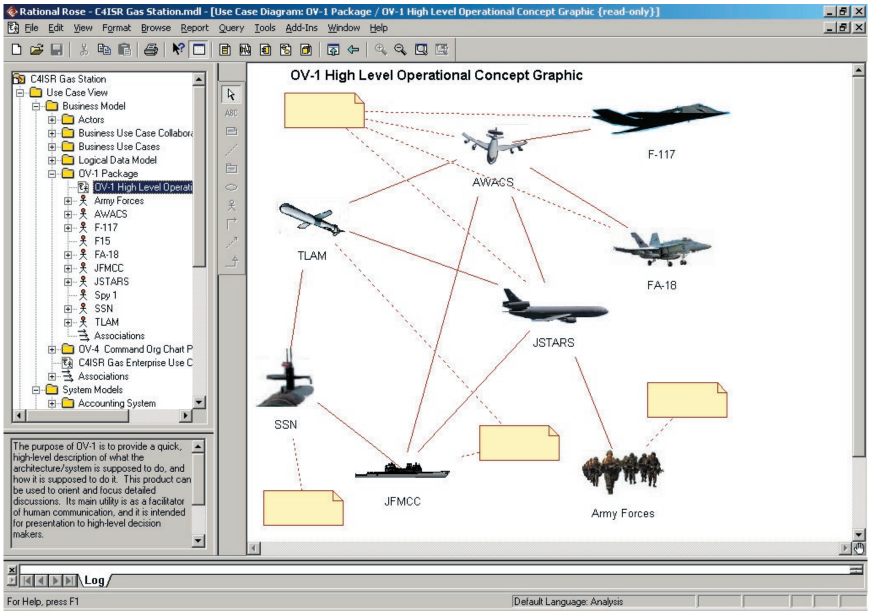


Dictionary document within IBM Rational Suite (in this screen shot, Microsoft Word with IBM Rational RequisitePro)

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Operational	OV-1	High-level Operational Concept Graphic	Essential	High-level graphical description of operational concept (high-level organizations, missions, geographic configuration, connectivity and so on) (4.2.1.3)
• Facilitates communications for high-level discussions and concentrates on the operational nodes, the communication between them and the characteristics of the information exchanged				

IBM Rational Suite Support

UML diagrams provide the capability to create nodes and elements, the linkages between them and the characteristics (attributes) of the information exchanged—with quantifiable metrics when needed. This is accomplished by defining stereotypes for the different nodes and elements that need to be represented. The diagram can be created and maintained in IBM Rational Suite.

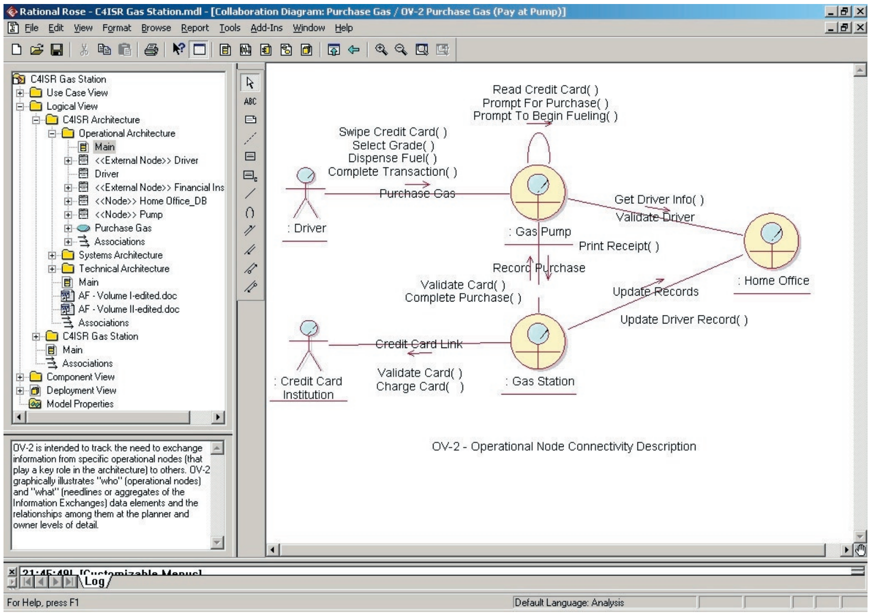


Use-case view with specific UML stereotypes (in this screen shot, with IBM Rational Rose)

Applicable Architecture View	**Product Reference**	**Architecture Product**	**Essential or Supporting**	**General Nature**
Operational	OV-2	Operational Node Connectivity Description	Essential	Operational nodes, activities performed at each node, connectivities and information flow between nodes (4.2.1.4)
• Facilitates communications for high-level discussions				

IBM Rational Suite Support

IBM Rational Suite provides support for the Operational Node Connectivity Description; and standard, reusable icons and graphics can be established for inclusion. Basically, UML provides capability to extend its notation through stereotypes for the different elements that need to be represented.



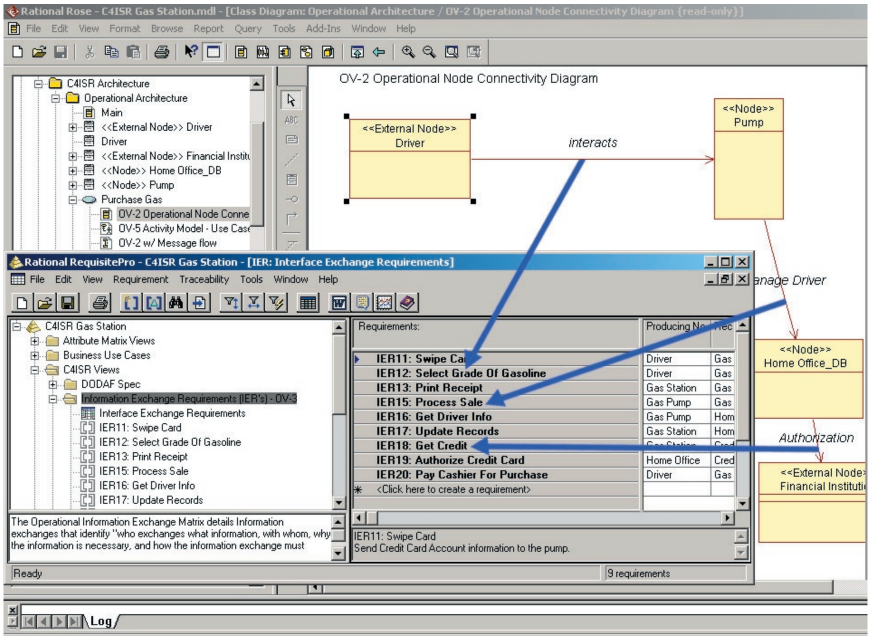
Operational architecture view with UML (in this screen shot, with IBM Rational Rose)

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Operational	OV-3	Operational Information Exchange Matrix	Essential	Information exchanged between nodes and the relevant attributes of those exchanges such as media, quality, quantity and the level of interoperability required (4.2.1.5)

- Captures information that is exchanged between nodes and the relevant attributes of those exchanges such as media, quality, quantity and the level of interoperability required

IBM Rational Suite Support

Information exchange matrix data can be captured in Rational Suite through IBM Rational RequisitePro matrix and traced to a relationship on a UML diagram, either in IBM Rational Rose®, IBM Rational Rose RealTime or IBM Rational XDE™ architecture and design modeling tools.



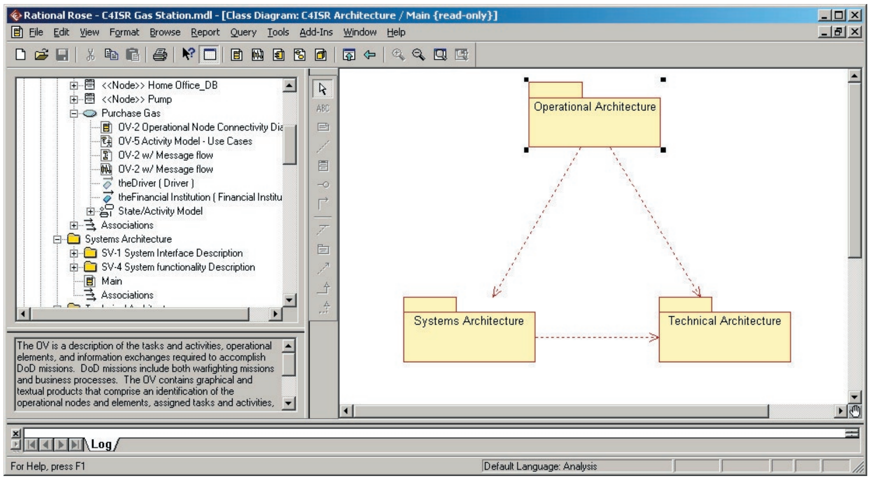
Traceability between UML diagram and information exchange details captured as requirements (in this screen shot, with IBM Rational Rose and IBM Rational RequisitePro)

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Operational	OV-4	Command Relationships Chart	Supporting	Command, control and coordination relationships between organizations (4.2.2.1)

- Shows command, control and coordination relationships between organizations

IBM Rational Suite Support

This chart is supported through IBM Rational Rose, IBM Rational Rose RealTime or IBM Rational XDE class or object diagrams with stereotyped relationships for command, control and coordination and stereotyped classes for organizations (graphical icons can be added for stereotyped elements).



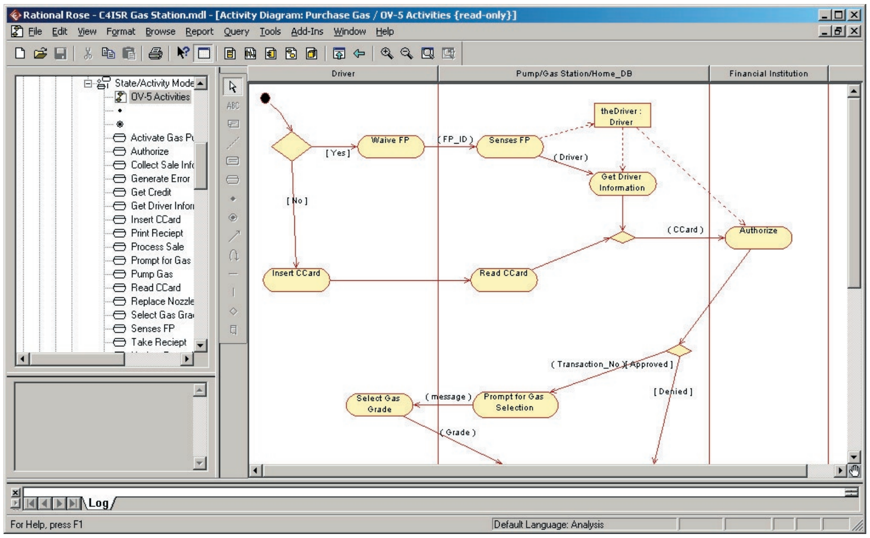
UML packages diagram (in this screen shot, with IBM Rational Rose)

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Operational	OV-5	Activity Model	Supporting	Activities, relationships among activities, I/Os, constraints (for example, policy and guidance) and mechanisms that perform the activities; in addition to showing mechanisms, overlays can show other pertinent information (4.2.2.2)

- Shows activities, relationships among activities, I/Os, constraints (for example, policy and guidance) and mechanisms that perform the activities; in addition to showing mechanisms, overlays can show other pertinent information

IBM Rational Suite Support

UML provides the capability to create activity models that show activities and their corresponding transitions including I/Os, constraints and mechanisms. The business use-case diagrams, activity diagrams and/or sequence diagrams are created in IBM Rational Suite.



UML activity diagram (in this screen shot, with IBM Rational Rose)

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Operational	OV-6a	Operational Rules Model	Supporting	In addition to showing mechanisms, overlays can show other pertinent information (4.2.2.2); 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)
• Is part of the architecture's operational view and extends the capture of business requirements and concept-of-operations information introduced by the logical data model				

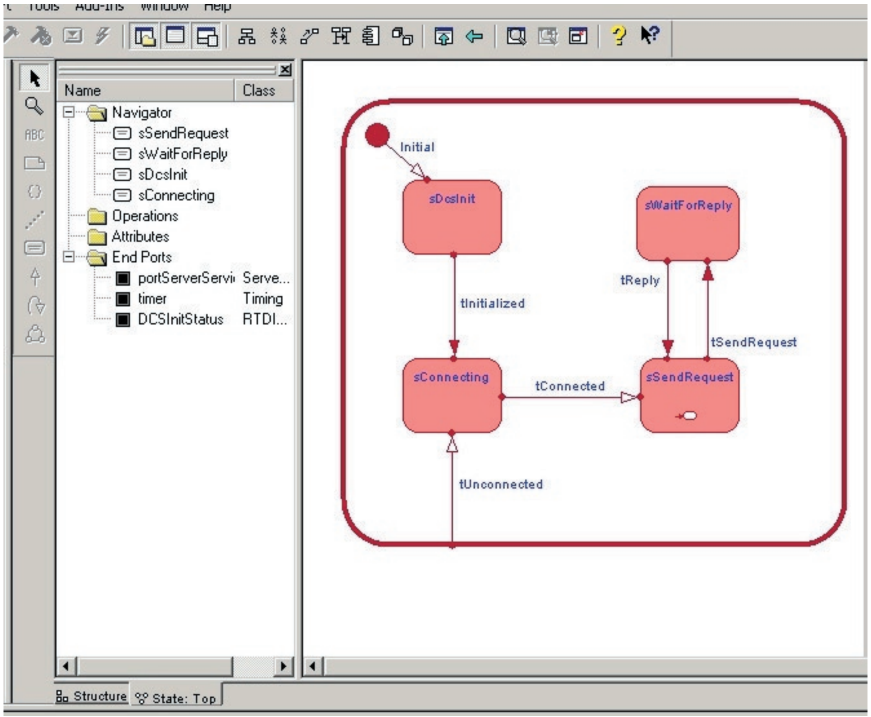
IBM Rational Suite Support

UML and its associated profiles provide the capability to present conceptual, logical and physical data models. In addition, a more formal description of the business rules can be captured in text using a formal language. UML also provides a formal language called the object constraint language, which can be annotated directly on a UML diagram. The business rules model can be modeled in various ways such as activity and state diagram decision boxes, multiplicity on associations on class diagrams or object interaction diagrams. Most importantly, this business rules model is created and maintained in Rational Suite and then consistency is maintained automatically through its model-based paradigm. Changes on one view that affect another view are automatically propagated (assuming there are no restrictions on such propagation) to the other view. Reports of inconsistencies are automatically generated from IBM Rational Suite.

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Operational	OV-6b	Operational State Transition Description	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)
• Relates events and states				

IBM Rational Suite Support

UML includes support for hierarchical state charts to model events and states. State charts are created and maintained in IBM Rational Suite.

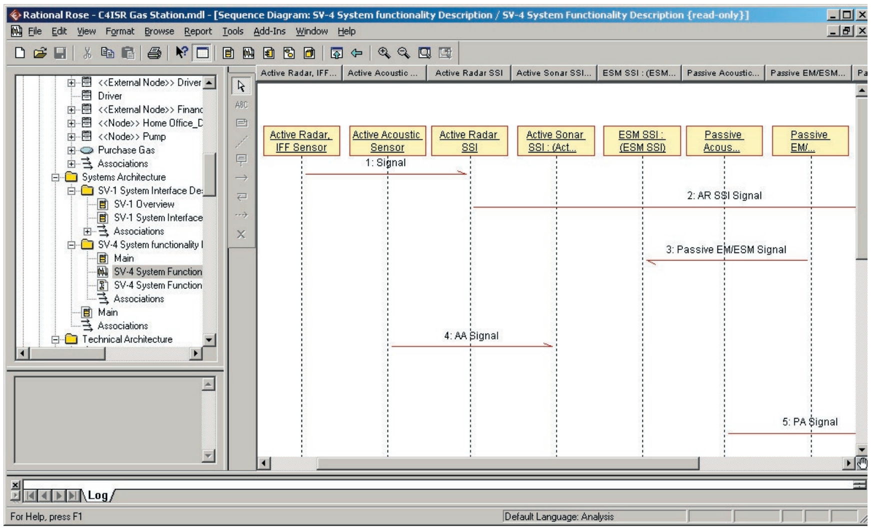


UML state diagram (in this screen shot, with IBM Rational Rose RealTime)

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Operational	OV-6c	Operational Event/Trace Description	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)
• Shows the tracing of actions in a scenario or critical sequence of events				

IBM Rational Suite Support

UML and IBM Rational Suite include support for sequence diagrams. Sequence diagrams can be built manually or automatically (when applicable) if the model can be executed, for example, with IBM Rational Rose RealTime.



UML sequence diagram (in this screen shot, with IBM Rational Rose)

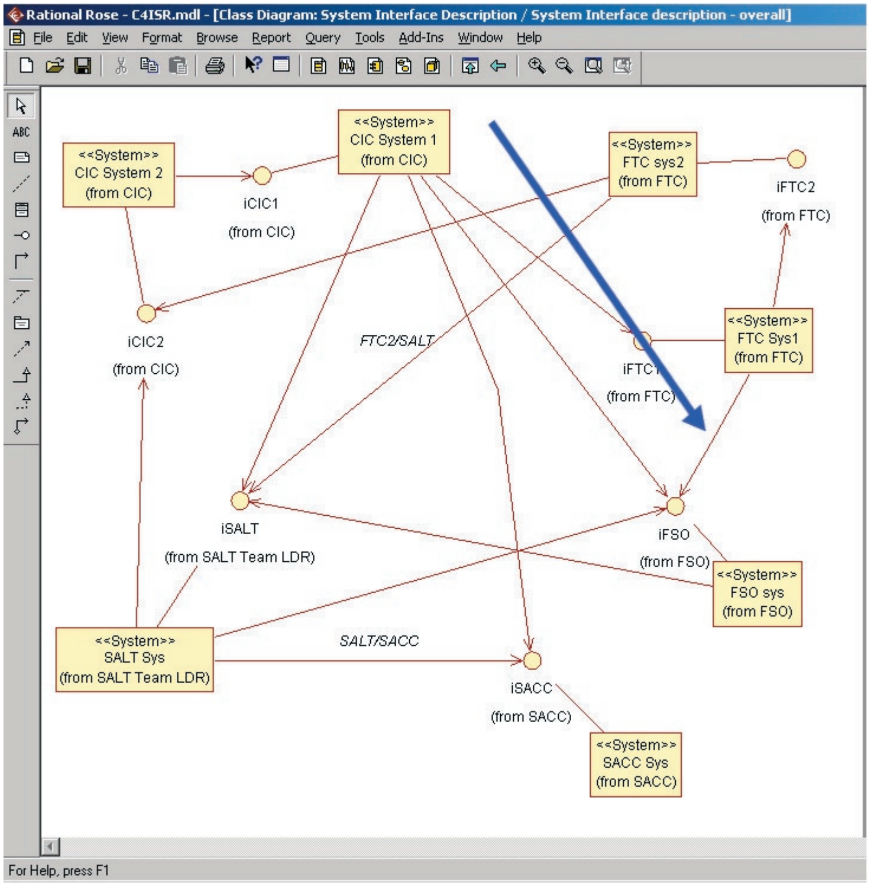
Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Operational	OV-7	Logical Data Model	Supporting	Documentation of the data requirements and structural business process rules of the operational view. (4.2.2.4)
• Documents the data requirements and structural business process rules of the architecture's operational view				

IBM Rational Suite Support

The UML Data modeling profile provides the capability to create a logical data model. The logical data model can be created from analysis classes that are designated as persistent or from a domain analysis. The logical data model can be created in IBM Rational Suite by building a conceptual data model with analysis classes. In addition, powerful forward and reverse engineering capabilities allow automatic synchronization to and from the logical data model, which can significantly facilitate the physical database code writing.

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Systems	SV-1	System Interface Description	Essential	Identification of systems and system components and their interfaces, within and between nodes (4.2.1.6)

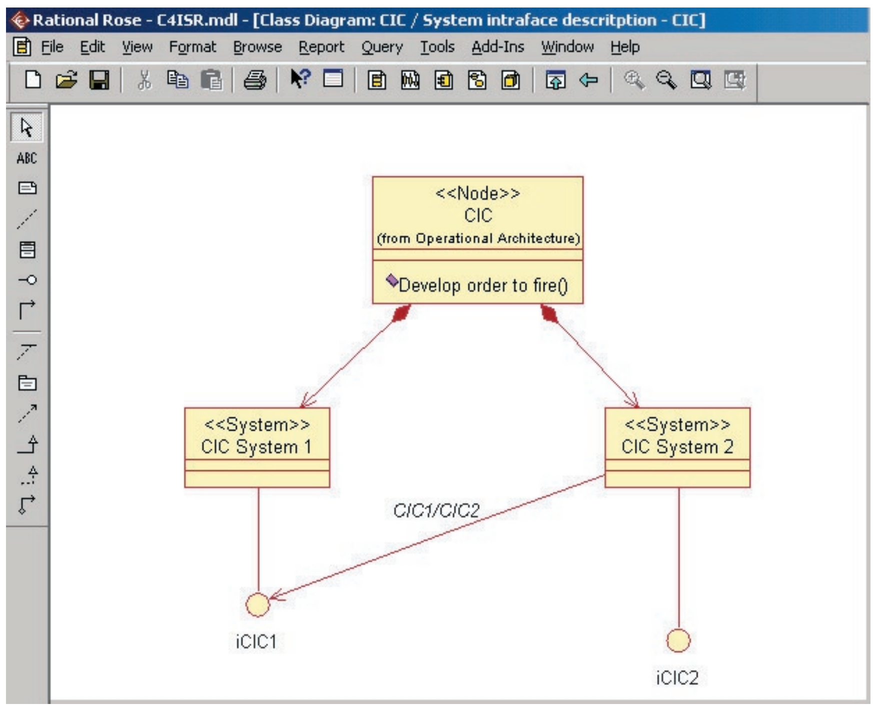
- Identifies systems and system components and their interfaces, within and between nodes
- Links the operational and systems views by describing the systems allocated to nodes and needlines in the Operational Node Connectivity diagram



Intrasystem communication

IBM Rational Suite Support

This view is supported in IBM Rational Suite. UML provides the capability to model systems communications and to link the systems to the needlines described in the Operational Node Connectivity diagrams. This model can be created and maintained in IBM Rational Suite. Additionally, the view can be extracted automatically based on the work performed when specifying the system nodes and components.



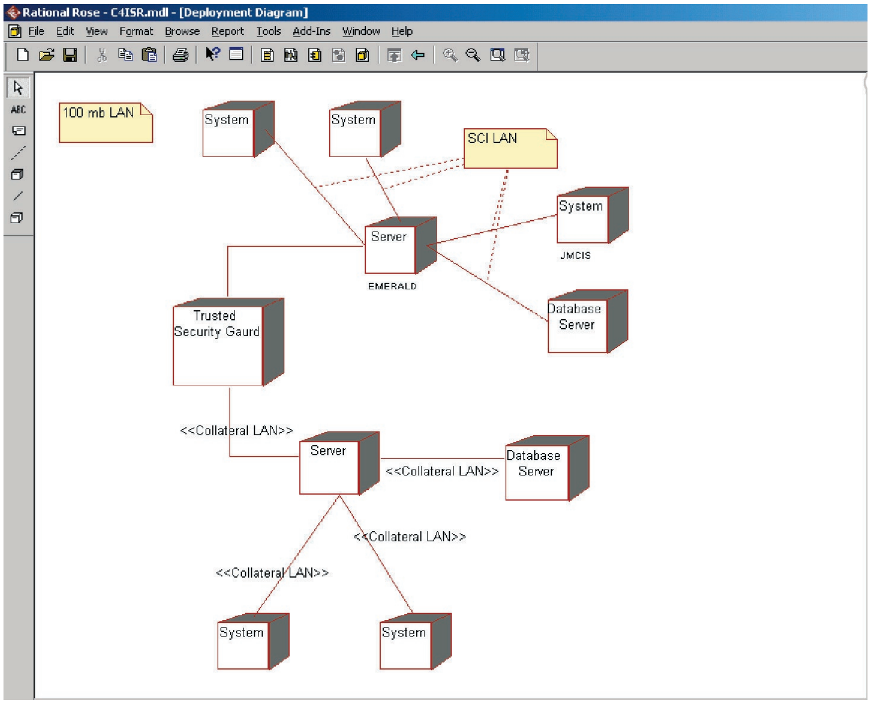
System Interface Description class diagram with IBM Rational Rose

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Systems	SV-2	Systems Commu-nications Description	Supporting	Physical nodes and their related commu-nications laydowns (4.2.2.5)

- Defines physical nodes and their related communications layouts
- Focuses on specific communication pathways or networks and the details of their configurations through which the physical nodes and systems communicate
- Represents the physical allocation of needlines in the Operational Node Connectivity diagrams

IBM Rational Suite Support

UML provides the capability to model systems communications and link the systems to the needlines described in the Operational Node Connectivity diagrams. This model can be created and maintained in IBM Rational Suite.



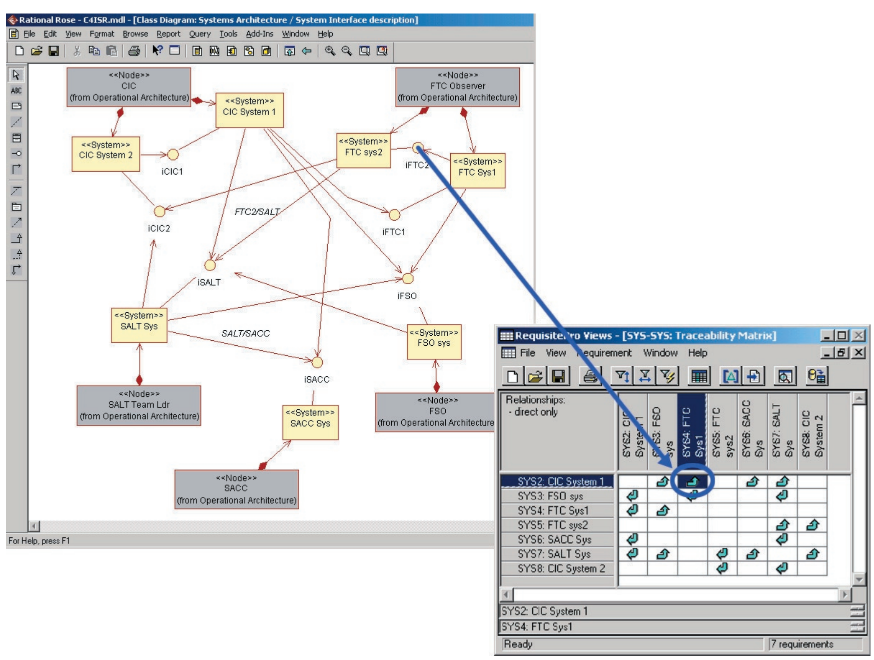
Deployment diagram with IBM Rational Rose

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Systems	SV-3	Systems Matrix	Supporting	Relationships among systems in a given architecture; can be designed to show relationships of interest—for example, system-type interfaces or planned versus existing interfaces (4.2.2.6)
• Represents the system-to-system relationships identified in the System Interface Description • Examines relationships among systems in a given architecture; can be designed to show relationships of interest (for example, system-type interfaces or planned versus existing interfaces)				

IBM Rational Suite Support

The systems-to-systems view is supported in IBM Rational Suite. This matrix can be automatically constructed from graphical work done to specify intersystem relationships, especially because of the use-case flow down approach described in RUP SE to determine systems and related subsystems along with aggregated services or interfaces (see RUP SE).

A matrix can be created automatically to list all the systems and the relationships between them. The traceability of these relationships and their consistency when changes occur are maintained within Rational Suite.



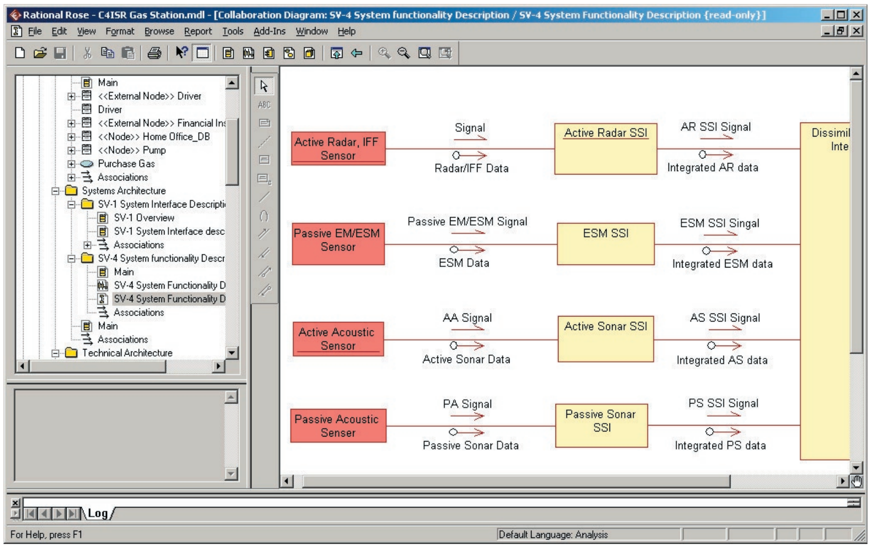
Traceability matrix within IBM Rational RequisitePro built from IBM Rational Rose diagrams

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Systems	SV-4	Systems Functionality Description	Supporting	Functions performed by systems and the information flow among system functions (4.2.2.7)

- Examines functions performed by systems and the information (data) flow among system functions

IBM Rational Suite Support

IBM Rational Suite provides comprehensive capabilities for logical decomposition. The ability to show blocks (functions) and data flowing among them is supported in Rational Suite. UML provides the capability to extend its notation through stereotypes. Data flow analysis can be accomplished in several UML diagrams including the collaboration diagram, activity diagram and state diagram. In each diagram, the data is defined as the arguments of a particular message or transition.

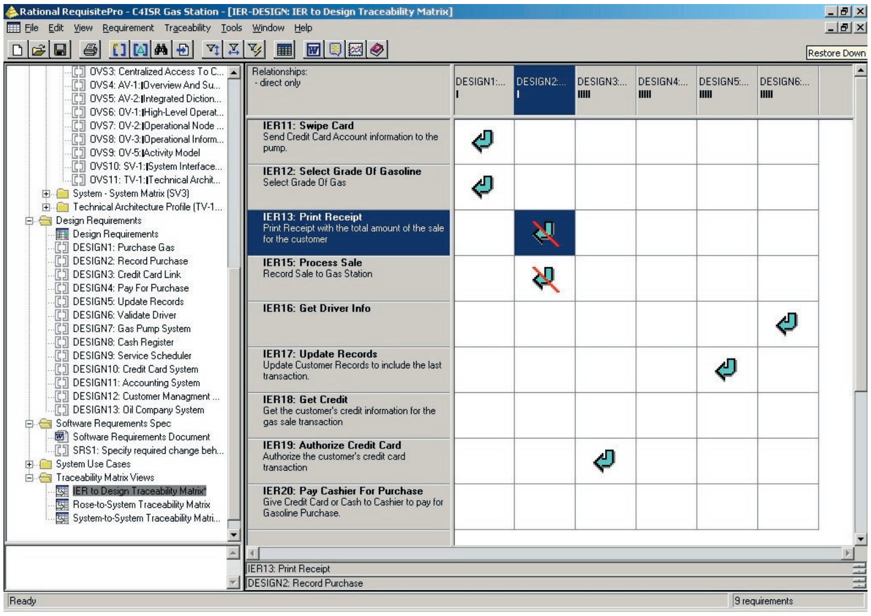


System Functionality Description class diagram in IBM Rational Rose

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Systems	SV-5	Operational Activity to System Function Traceability Matrix	Supporting	Mapping of system functions back to operational activities (4.2.2.8)
• Examines the mapping of system functions back to operational activities				

IBM Rational Suite Support

This traceability matrix is supported in IBM Rational Suite. After establishing the matrix, the impact of change is automatically tracked; if an operational activity is changed, the impact across system functionality is easily assessed.



Traceability matrix and automatic impact analysis with IBM Rational RequisitePro

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Systems	SV-6	System Information Exchange Matrix	Supporting	Detailing of information exchanges among system elements, applications and hardware allocated to system elements (4.2.2.9)
• Details information exchanges among system elements, application and hardware allocated to system elements				

IBM Rational Suite Support

This system information exchange view can be expressed as a matrix. After establishing the matrix, the impact of change is automatically tracked (see SV-5 in Table 1).

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Systems	SV-7	System Performance Parameters Matrix	Supporting	Performance characteristics of each systems' hardware and software elements, for the appropriate timeframe(s) (4.2.2.10); planned incremental steps toward migrating a suite of systems to a more efficient suite
• Details performance characteristics of each system's hardware and software elements, for the appropriate timeframe(s)				

IBM Rational Suite Support

The system performance parameters matrix is supported in IBM Rational Suite. Each performance characteristic can be tracked, for example, as formal descriptions in Rational Suite within UML diagrams, the requirements database or formal test cases. Where technically feasible, measurements against the parameters can be scripted and automatically remeasured and then compared with requirements. Characteristics of throughput, performance, functionality and reliability can be tracked and automated in this manner.

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Systems	SV-8	System Evolution Description	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)
• Details planned incremental steps toward migrating a suite of systems to a more efficient suite, or toward evolving a current system to a future implementation				

IBM Rational Suite Support

IBM Rational Suite supports creating a system evolution description within the RUP core disciplines. Maintenance, evolution or migration activities are typically part of the project life cycle. Furthermore, a program may use third-party project management tools to create the timeline and to tie system requirements to different points on the schedule.

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Systems	SV-9	System Technology Forecast	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)
• Details 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				

IBM Rational Suite Support

The technology forecast can be created in IBM Rational Suite through the use of additional specifications that meet a prospective purpose. Additional requirements are maintained in Rational Suite to satisfy this need. Technology entries can be individually tracked and attributes associated therewith.

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Systems	SV-10a	Systems Rules Model	Supporting	One of three products used to describe systems activity sequence and timing — constraints that are imposed on systems functionality because of system design or implementation (4.2.2.13.1)

- Is one of three products used to describe systems activity sequence and timing
- Details constraints that are imposed on systems functionality because of system design or implementation

IBM Rational Suite Support

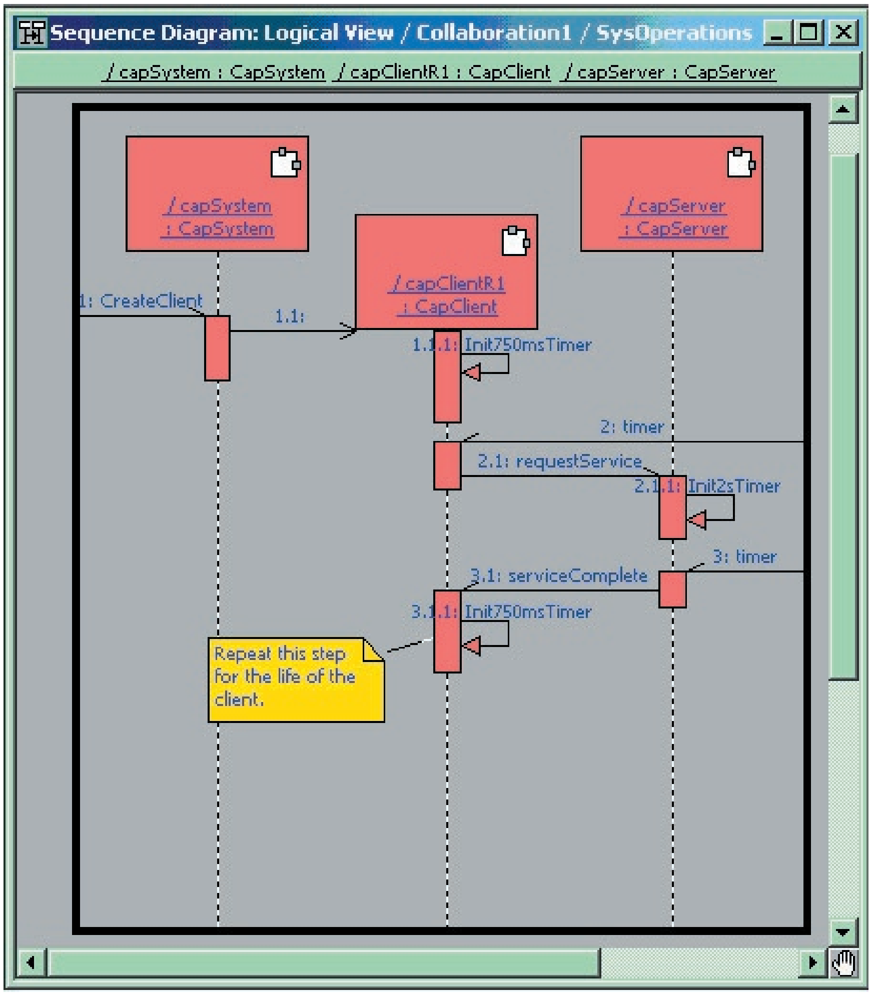
Tracking the textual description line by line is achievable within IBM Rational Suite. Advanced concepts such as state nesting, complex transitions and annotated scenario traces are also supported. For example, because constraints come from different domains (the business, the architecture, quality objectives and so on), each type of constraint is best expressed using different UML mechanisms in their contexts/diagrams.

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Systems	SV-10b	Systems State Transition Description	Supporting	One of three products used to describe systems activity sequence and timing —responses of a system to events (4.2.2.13.2)

- Is one of three products used to describe systems activity sequence and timing
- Details the responses of a system to events

IBM Rational Suite Support

Tracking the textual description line by line is achievable within IBM Rational Suite. Advanced concepts such as state nesting, complex transitions and annotated scenario traces are also supported. Timing information can be either captured or specified within a formal UML description.



Annotated sequence diagram with timing information in IBM Rational Rose RealTime

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Systems	SV-10c	Systems Event/Trace Description	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)

- Is one of three products used to describe systems activity sequence and timing
- Details system-specific refinements for critical sequences of events described in the operational view

IBM Rational Suite Support

Tracking the textual description line by line is achievable within IBM Rational Suite. Advanced concepts such as state nesting, complex transitions and annotated scenario traces are also supported.

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Systems	SV-11	Physical Data Model	Supporting	Physical implementation of the information of the logical data model—for example, message formats, file structures and physical schema (4.2.2.14)

- Shows the physical implementation of the information provided in the logical data model (for example, message formats, file structures and physical schema)

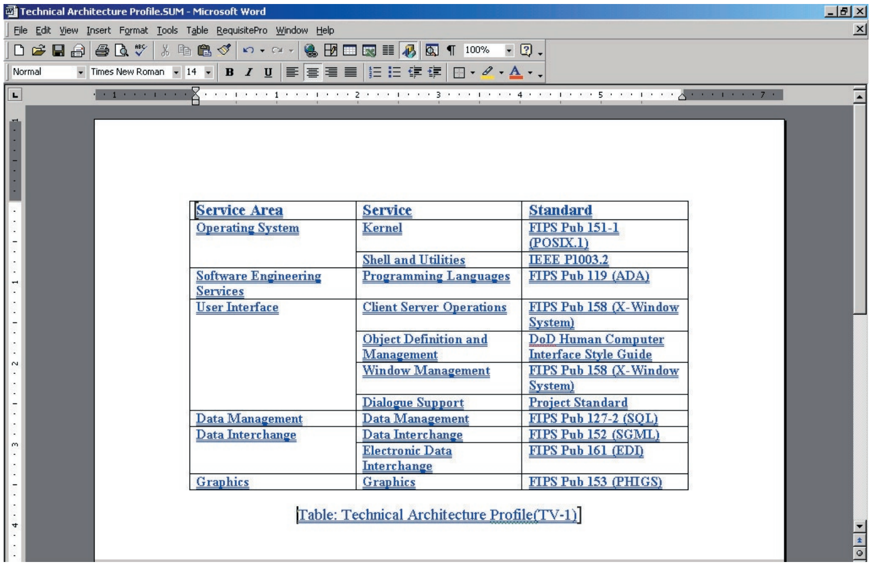
IBM Rational Suite Support

This view of the data model is supported in IBM Rational Suite. Relationships to the higher-level requirements can be maintained with the logical data model through IBM Rational Rose, IBM Rational Rose RealTime and XDE modeling tools and IBM Rational RequisitePro.

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Technical	TV-1	Operational Activity to System Function Traceability Matrix	Essential	Mapping of system functions back to operational activities (4.2.2.8)
• Examines standards that apply to the given architecture				

IBM Rational Suite Support

IBM Rational Suite provides the capability to create Microsoft Word documents and maintain the documents either with IBM Rational RequisitePro or IBM Rational ClearCase® integrated software configuration management solution. IBM has developed close partnerships with many government programs. The technical staffs of the distributed teams are in the unique position to advise and help with standard qualification and architecture processes.



Service Area	Service	Standard
Operating System	Kernel	FIPS Pub 151-1 (POSIX.1)
	Shell and Utilities	IEEE P1003.2
Software Engineering Services	Programming Languages	FIPS Pub 119 (ADA)
User Interface	Client Server Operations	FIPS Pub 158 (X-Window System)
	Object Definition and Management	DoD Human Computer Interface Style Guide
	Window Management	FIPS Pub 158 (X-Window System)
	Dialogue Support	Project Standard
Data Management	Data Management	FIPS Pub 127-2 (SQL)
Data Interchange	Data Interchange	FIPS Pub 152 (SGML)
	Electronic Data Interchange	FIPS Pub 161 (EDI)
Graphics	Graphics	FIPS Pub 153 (PHIGS)

Table: Technical Architecture Profile(TV-1)

Technical Architecture Profile document maintained with IBM Rational Suite

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
Technical	TV-2	Standards Technology Forecast	Supporting	Description of emerging standards that are expected to apply to the given architecture, within an appropriate set of timeframes (4.2.2.15)
• Describes emerging standards that are expected to apply to the given architecture, within an appropriate set of timeframes				

IBM Rational Suite Support

The standards forecast is supported in IBM Rational Suite. Forecast changes can be tracked and managed. IBM technical staffs are in the unique position to advise and help with creating this forecast.

Conclusion

According to a recent study,[12] the total time required to create the enterprise architecture at Cisco Systems was three years. This three-year effort included a 15-month project focused on business process reengineering, a nine-month project dedicated to EA design and a one-year phase for building and deploying the system. The same government report states that the planned timeframe established for the DoD effort is one-year for EA design and development of a transition plan, followed by validation and a prototype. An aggressive rollout is planned for 2005, with completion in 2007. Currently, according to the same source, the industry[13] average for EA implementations is five years.

A review of U.S. federal agencies' use of architectures began in 1994, initially focusing on those agencies pursuing high-risk major system modernization programs. These included the National Weather Service system modernization, the Federal Aviation Administration air traffic control modernization and the Internal Revenue Service tax systems modernization. Generally, it had been reported that the EAs serving these agencies were incomplete, along with some recommendations that they develop and

[12] Chief Financial Officers Council (CFOC), April 16, 2002, http://www.cfoc.gov/documents/doc_cfo_mins04_16_02.doc.

[13] Benchmarked EAs included Cisco Systems, Northrup Grumman, the United Kingdom Ministry of Defence, Hershey Food Corporation, Canada's National Defence, Department of Commerce, National Aeronautics and Space Administration (NASA), the Commonwealth of Kentucky and The Gillette Company.

implement complete EAs to guide their modernization efforts. Since then, numerous other federal organizations—such as the Department of Education, Customs Service, Immigration and Naturalization Service and Centers for Medicare and Medicaid Services—have undertaken EA programs in conjunction with large-scale modernization.

The analysis showed that few organizations have developed and implemented complete EAs.[14] Because few EAs were developed using a single framework or because little consideration was given to business processes, policies and people, many organizations failed to accomplish the intimidating unification process of EA implementation. Lessons learned indicate that the following best practices that should be considered when planning and implementing an EA:

- Manage the project as an iterative process and leverage iteration achievements to gain momentum.
- Enforce compliance through the use of existing standards in conjunction with proven methodologies.
- Incorporate, plan and manage change-management efforts throughout the project.
- Staff the project with top talent and ensure long-term commitment to the project.
- Implement knowledge-transfer procedures and consider how the talent will be reintegrated into the organization once the EA project is completed.
- Leverage COTS software and minimize customization.

As described in this white paper, IBM Rational can provide the tools and resources to meet these success factors. RUP SE, delivered as a RUP plug-in, is an application of the Rational Unified Process framework to support the development of large-scale systems that are composed of software, hardware, workers and information components. RUP SE can provide system architects with the advantages of RUP best practices while providing a setting for addressing overall system issues. Some of the benefits of RUP SE include:

---

[14] CFOC, April 16, 2002, http://www.cfoc.gov/documents/doc_cfo_mins04_16_02.doc.

- **System Team Support –** Enables ongoing collaboration of business analysts, architects, systems engineers, software developers, hardware developers and testers
- **System Quality –** Provides the views to address system quality issues in architecture-driven systems development
- **System Visual Modeling –** Provides UML support for systems architecture
- **Scalability –** Scales from medium to very large systems
- **Component Development –** Provides workflows for determining hardware and software components
- **System Iterative Design and Development –** Supports concurrent design and iterative development of hardware

For more information on IBM Rational offerings and capabilities in the area of enterprise architecture, please visit **ibm.com**/software/rational.

www.ibm.com

IBM Software Group
Route 100
Somers, NY 10589

G507-1005-00