A technical discussion of FEAF using RUP and UML
05/20/03

Development of Federal Enterprise Architecture Framework using the IBM Rational Unified Process and the Unified Modeling Language

Allen Sayles
Senior Systems & Software Engineer
Rational Brand Services
IBM Software Group

Development of Federal Enterprise Architecture Framework using the Rational Unified Process and the UML

The IBM® Rational Unified Process® (RUP®) is uniquely qualified to support Enterprise Architecture (EA) efforts of groups and agencies that are following the guidelines of the Federal Enterprise Architecture Framework (FEAF). The Rational Unified Process can help customers successfully capture, manage, and use their Enterprise Architectures. In this paper we will explore how RUP and the UML can be used to build and manage Enterprise Architectures. Specifically, we will examine the FEAF level IV matrix to discuss how RUP facilitates capturing various FEAF models.

Background

The Clinger-Cohen Act of 1996 mandated that federal agencies develop and maintain an enterprise IT architecture in order to promote information sharing and organization among federal agencies. In 1999, The Federal Chief Information Officers (CIO) responded to this mandate by establishing the **Federal Enterprise Architecture Framework (FEAF)** *http://www.cio.gov/documents/fedarch1.pdf*. The purpose of the FEAF is to establish an agency-wide roadmap to achieve an agency's mission through optimal performance of its core business processes within an efficient information technology (IT) environment. Enterprise architectures (EAs) help agencies accomplish this; simply stated, they are blueprints for systematically and completely defining an organization's current (baseline) or desired (target) environment. EAs are essential for evolving information systems and developing new systems that optimize their mission value. This is accomplished in logical or business terms (e.g., mission, business functions, information flows, and systems environments) and technical terms (e.g., software, hardware, communications), and includes a Sequencing Plan for transitioning from the baseline environment to the target environment.

If defined, maintained, and implemented effectively, these institutional blueprints assist in optimizing the interdependencies and interrelationships among an organization's business operations and the underlying IT that support operations. The experience of the US federal Office of Management and Budget (OMB) and General Accounting Office (GAO) has shown that without a complete and enforced EA, federal agencies run the risk of buying and building systems that are duplicative, incompatible, and unnecessarily costly to maintain and integrate.

Frameworks

Within the government, there are several different Enterprise Architecture Frameworks, including FEAF, DoD Architecture Framework, and others designed by specific agencies. They all share a common goal: to reduce substantially the inconsistency of architectural descriptions across the federal government. Therefore, an EA framework allows more efficient analysis of duplication and redundancies of business processes and systems both within and across agencies.

According to the FEAF[1], the framework enables the federal government to:

- Organize federal information on a federal-wide scale
- Promote information sharing among federal organizations
- Help federal organizations develop their architectures
- Help federal organizations quickly develop their IT investment processes
- Serve customer needs better, faster, and more cost effectively.

> *"If you fail to invest in a well-defined information architecture, you will cripple the knowledge infrastructure that is the foundation for the intelligent learning organization."* Larry P. English Federal Enterprise Architecture Framework 1.1

The Federal Enterprise Architecture Framework Overview

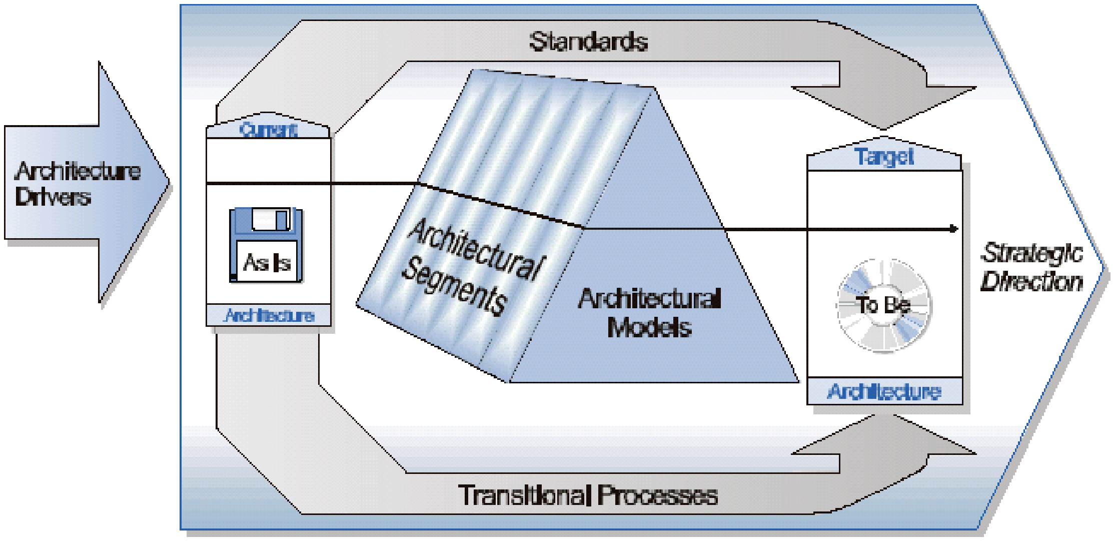
The Federal Enterprise Architecture Framework is an organizing mechanism for managing the development and maintenance of architecture descriptions. The FEAF also provides a structure for organizing federal resources and describing and managing Federal Enterprise Architecture activities. The framework does this by organizing information about the enterprise into various levels, or frames of reference. The top level, Level I, is the highest-level view of the enterprise. The bottom level, Level IV, contains the most detailed information about the enterprise. It partitions the Enterprise Architectures into business, data, application, and technology architectures. The FEAF also takes into account elements of the Zachman Framework[1] and uses the Spewak[2] EA planning methodology.

FEAF Levels

The FEAF identifies eight components needed to develop and maintain a Federal Enterprise Architecture. A decomposition of the eight components provides further granularity resulting in the FEAF, which contains four levels. The first three levels illustrate the progression of eight increasingly detailed components leading to a structure for classifying and organizing the descriptive representations of the Federal Enterprise in level IV. After a brief discussion of Levels I-III below, this paper will discuss Level IV in detail.

Level I

Level I is the highest level of the Federal Enterprise Architecture Framework; it introduces the eight components needed for developing and maintaining the Federal Enterprise Architecture. As shown in figure 1, the flow of the framework, from left to right, represents the continuous process of the Federal Enterprise Architecture.



[1] Zachman, John A. *A Framework for Information Systems Architecture*. IBM Publication G321-5298. 914-945-3836. IBM Systems Journal. Vol. 26, No. 3. 1987.

[2] Spewak, Steven H. with Steven C. Hill. *Enterprise Architecture Planning, Developing aBlueprint for Data, Applications and Technology*. John Wiley & Sons, Inc., 1992.

Figure 1. Federal Enterprise Architecture Framework, Level I

Level I of the FEAF is described by the following eight elements:

- **Architecture Drivers** – Represents an external stimulus that causes the Federal Enterprise Architecture to change
- **Strategic Direction** – Ensures that changes are consistent with the overall government direction
- **Current Architecture** – Represents the current state of the enterprise or agency. Full characterization may be significantly beyond its worth and maintenance.
- **Target Architecture** – Represents the target state for the enterprise within the context of the strategic direction.
- **Transitional Processes** – These processes apply the changes from the current architecture to the target architecture in compliance with the architecture standards, such as various decision making or governance procedures, migration planning, budgeting, and configuration management and change control.
- **Architectural Segments** – These focus on a subset or a smaller enterprise within the total enterprise.
- **Architectural Models** – Provide the documentation and the basis for managing and implementing changes in the enterprise.
- **Standards** – Include agency adopted standards (both mandatory and voluntary) including best practices and various open standards, all of which focus on promoting interoperability.

Level II

Level II shows, at a greater level of detail, the business and design aspects of the Federal Enterprise Architecture and how they are related. The relationship of business and design architectures is push/pull – the business pushes design to meet its needs, and design (i.e., new developments in data, applications, and technology) pulls business to new levels of service delivery in support of business operations.

The same eight elements described for Level I are elaborated in Level II to provide additional granularity of business and design. For example, at Level II, when looking at the Current Architecture component, we would be concerned with the Current Business Architecture, which identifies the current business needs that are supported by the current design, and the Current Design Architectures, which define the currently implemented data, applications, and technologies used to support the current business needs. A similar perspective can be observed for the other components in Level II.

Level III

Level III expands the design pieces of the framework to show the three design architectures: data, applications, and technology as shown in figure 2.

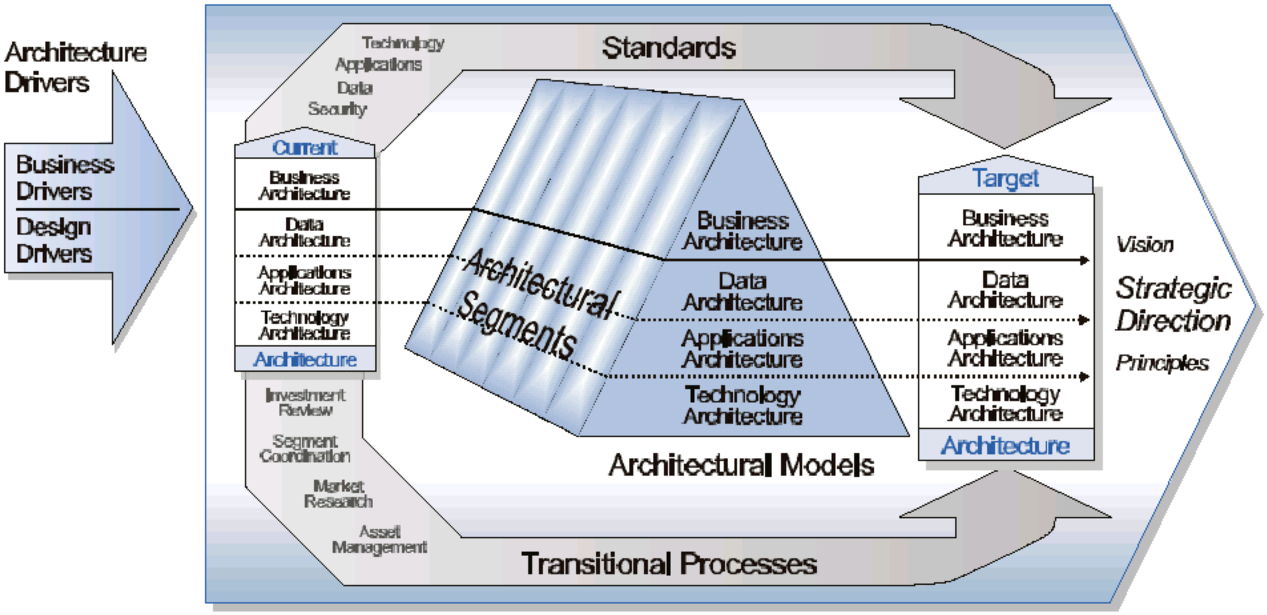


Figure 2. Level III of the Federal Enterprise Architecture Framework

The design architecture within Level III further elaborates on the design details outlined at level II. Below are samples of three of the six components that are further elaborated at Level III.

- **Current Design Architectures** - The currently implemented designs used to support the current business needs. The current design architectures consist of the following three architectures.
  - Current Data Architecture - Defines what data is in place to support the business (i.e., data models).
  - Current Application Architecture - Defines what applications are in place to manage the data and support the business functions (i.e., application models).
  - Current Technology Architecture - Defines what supporting technology is in place to provide an environment for applications that manage the data and support the business functions (i.e., technology models).
- **Target Design Architectures** - The future designs to be used to support the future business needs. The target design architecture consists of the following three architectures.
  - Target Data Architecture - Defines the data needed to support the business (i.e., data models).
  - Target Applications Architecture - Defines the applications needed to manage the data and support the business functions (i.e., applications models).
  - Target Technology Architecture - Defines the supporting technology needed to provide an environment for applications that manage the data and support the business functions (i.e., technology models).
- **Design Models** - Three types of models used to define the enterprise.
  - Data Models - Define the enterprise.
  - Application Models - Define the applications that control the data.
  - Technology Models - Define the current and target technology.

Level III also provides additional detail for Architectural Segment, Transitional Processes, and Standards components.

Level IV

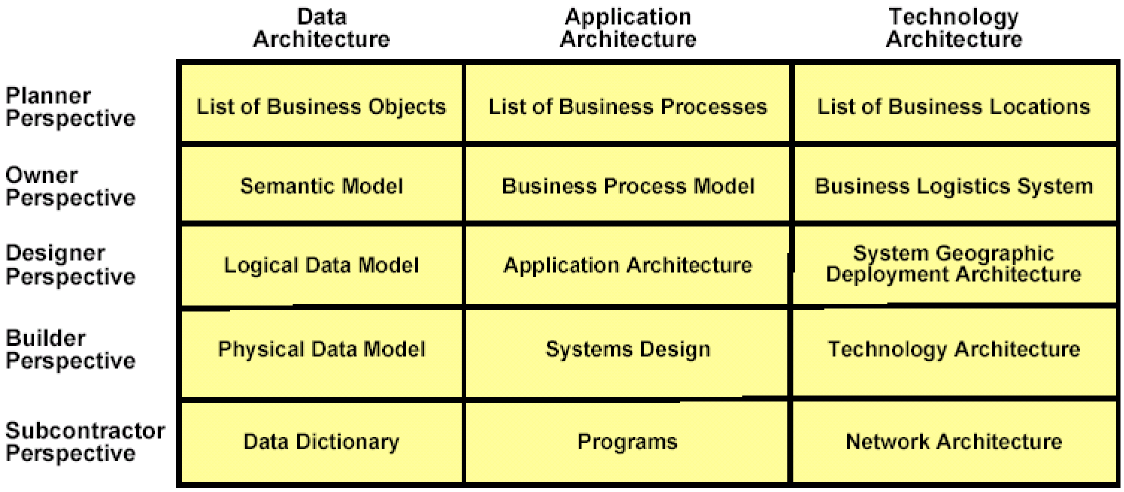
Level IV (the view from 1,000 to 500 feet) identifies the kinds of models that describe the business architecture and the three design architectures: data, applications, and technology. It also defines enterprise architecture planning. At level IV, how the business architecture is supported by the three design architectures begins to evolve and be made explicit. At this level, the FEAF identifies two mechanisms, the FEAF matrix and the Enterprise Architecture Planning (EAP) Methodology. The FEAF matrix is used to organize the architectural information and the EAP helps to define what architectures are appropriate for the specific enterprise.

Below we will examine the FEAF matrix via a general overview of FEAF architecture and its components and an orientation to the IBM Rational Unified Process, or RUP, in the context of FEAF architecture. This paper will then present the FEAF matrix in more detail, showing how RUP can be used to support the various roles required from the FEAF matrix.

FEAF Matrix Overview

The FEAF provides a structure to develop, maintain, and implement top-level operating environments and support implementation of IT systems. The structure classifies and organizes the significant models of an enterprise, based on the Zackman framework. The Zachman Framework was developed in 1987 by John Zachman as a means for organizations to assess the completeness of software development process models in terms of their overall information requirements. The framework provides multiple perspectives on the complete architecture and a categorization of the artifacts of the architecture. The Zachman Framework is actually a matrix of 36 cells covering the *who, what, where, when, why*, and *how* of an enterprise. The framework splits the enterprise into six perspectives, starting at the highest level of business abstraction all the way down to implementation. The framework can contain global plans as well as technical details, lists, and charts. Any appropriate approach, standard, role, method, or technique may be placed in it.

FEAF focus on three aspects of the Zachman Framework, data (the "what"), process or application (the "how"), and location or technology (the "where"). As shown in Figure 3, the FEAF is graphically represented as a 3x5 matrix with architecture types (Data, Application, and Technology) on one axis of the matrix, and perspectives (Planner, Owner, Designer, Builder, and Subcontractor) on the other. The corresponding EA products are listed within the cells of the matrix. Later in this paper we will go into detail on structure of the FEAF matrix.



	Data Architecture	Application Architecture	Technology Architecture
Planner Perspective	List of Business Objects	List of Business Processes	List of Business Locations
Owner Perspective	Semantic Model	Business Process Model	Business Logistics System
Designer Perspective	Logical Data Model	Application Architecture	System Geographic Deployment Architecture
Builder Perspective	Physical Data Model	Systems Design	Technology Architecture
Subcontractor Perspective	Data Dictionary	Programs	Network Architecture

Figure 3. FEAF Architecture Matrix

Enterprise Architecture Planning Overview

The Enterprise Architecture Planning methodology helps define what data, application, and technology architectures are suitable for supporting the enterprise. The EAP is distilled into 7 components (or steps). Figure 4 shows the seven components of EAP for defining these architectures and the related migration plan. The seven components are in the shape of a wedding cake, with each layer representing a different focus of each major task (or step).

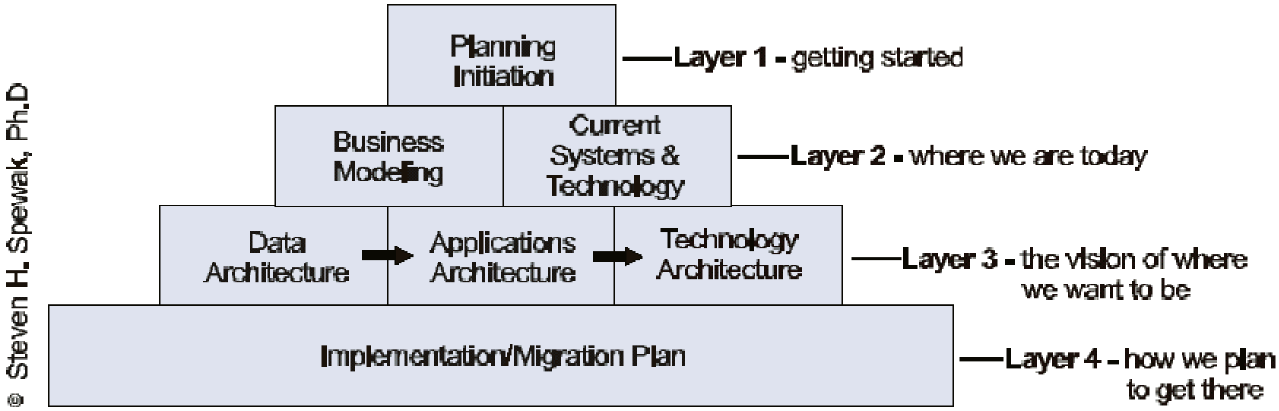

Planning Initiation
Layer 1 - getting started
Business Modeling
Current Systems & Technology
Layer 2 - where we are today
Data Architecture
Applications Architecture
Technology Architecture
Layer 3 - the vision of where we want to be
Implementation/Migration Plan
Layer 4 - how we plan to get there
© Steven H. Spewak, Ph.D

Figure 4. Components of Enterprise Architecture Planning[3] by Steven Spewak.

The Federal Enterprise Architecture Framework recognizes that architecture development and maintenance requires a process that continually evaluates current conditions and potential solutions. Key aspects[4] of the process include:

- Obtaining executive buy-in and support,
- Establishing a management structure that outlines various roles and activities to facilitate the development of the EA,
  Defining an Architecture process and approach,
- Developing both baseline and target EAs,
- Developing a gap analysis to create a sequencing plan to transition systems, applications, and business processes,
- Using the Enterprise Architecture to prioritize implementation decisions and investments in organizational change, and
- Managing the change of the Enterprise architecture over time as the agencies needs are continuously changing and evolving.

So far, this paper has provided definitions of enterprise architecture, a background on its driving factors, and some of the processes for building an EA as described by the CIO Council. The IBM Rational Unified Process supports these same key points. We will now focus on defining an architectural process and approach, establishing roles, identifying RUP disciplines for EA, and identifying RUP activities to build an EA as prescribed by the FEAF matrix.

[3] The Chief Information Officers Council, "Federal Enterprise Architecture Framework.", Version 1.1, September 1999.

[4] The Chief Information Officers Council, "A Practical Guide to Federal Enterprise Architecture.", Version 1, February 2001.

The Rational Unified Process

The IBM Rational Unified Process (RUP) is a Web-enabled set of software and system engineering best practices that can help guide a team's Enterprise Architecture development activities. As an industry-wide process platform, RUP makes it easy for practitioners to choose and customize the set of process components that are right for specific needs. A team will achieve more predictable results when it is unified with common processes that improve communication and create a common understanding of all tasks, responsibilities, and artifacts. The RUP is mated with the Unified Modeling Language (UML) to provide a mechanism for visualizing, specifying, constructing, and documenting the artifacts of system architectures. The RUP is a good choice for guiding EA development because it places significant focus on establishing an architecture that is well defined, meaningful, and useful. In addition RUP ensures that the users and stakeholders of the enterprise are taken into account through the development of the architecture. This approach blends well with the OMB's emphasis on "service to the citizen". Finally, RUP supports an iterative process, which takes into account the evolving nature of architecture, from "as-is" to "to-be" and the incremental steps between.

In RUP, the architecture of an enterprise is the organization or structure of the enterprise's significant components. The intent of defining the architecture is not to be complete, but rather to cover the breadth of the organization. RUP provides nine different disciplines to facilitate best practices across the system development lifecycle. For Enterprise Architecture development, we focus on six of them: Business Modeling, Requirements, some aspects of Analysis & Design, Configuration Management, Project Management, and Environment.

To define any given Enterprise Architecture, you must first define an architectural representation — that is, a way of describing important aspects of architecture. The FEAF uses a matrix to provide multiple views or perspectives of the enterprise. Each architectural view addresses some specific set of concerns, specific to stakeholders in the process: for example, end users, designers, managers, system engineers, maintainers, and so on. These various architectural views serve as communication media between the architect and other project team members regarding architecturally significant decisions. In a similar fashion, RUP also promotes different views of architecture based on the stakeholders and their needs.

Roles

To enable teams to build FEAF artifacts, the Practical Guide to building Enterprise Architectures[5] defines a set of roles that map directly to roles within the RUP. The RUP can be customized to include detailed responsibilities, actions, and artifacts for each FEAF role. Figure 5 shows a mapping of roles between FEAF and RUP.

EA Program Management Office Role	Primary RUP Role
Chief Architect	Project Manager
Senior Architecture Consultant	Architecture Reviewer & Systems Analyst
Business Architect	Business Designer & Business-Process Analyst
Applications Architect	Software Architect & Systems Analyst
Information Architect	Database Designer & Business Process Analyst
Infrastructure Architect	Systems Engineer & Systems Administrator
Security Systems Architect	Systems Architect (Security Expert)
Technical Writer	Technical Writer
Quality Assurance	Test Manager & Configuration Manager
Risk Management	Project Manager
Configuration Control	Configuration Manager

Figure 5. Roles Described by the Enterprise Architecture Program Management Office Mapped to Roles in the Rational Unified Process

[5] Ibid.

For this paper, we have graphically outlined each role in RUP to provide a better understanding of the responsibilities, actions, and artifacts. Figure 6 provides an example of how the project manager role is defined. In addition we have found that a role called systems architect,[6] or solution architect, is increasingly necessary to facilitate the development of enterprise architectures. This role integrates many of the roles above in more of a jack-of-all-trades approach. This works well for Enterprise Architecture, because we do not need detailed understand of all the different architectural models; rather, what we need is sufficient understand of the different architectural areas.

Role: Project Manager

The project manager role allocates resources, shapes priorities, coordinates interactions with customers and users, and generally keeps the project team focused on the right goal. The project manager also establishes a set of practices that ensure the integrity and quality of project artifacts.

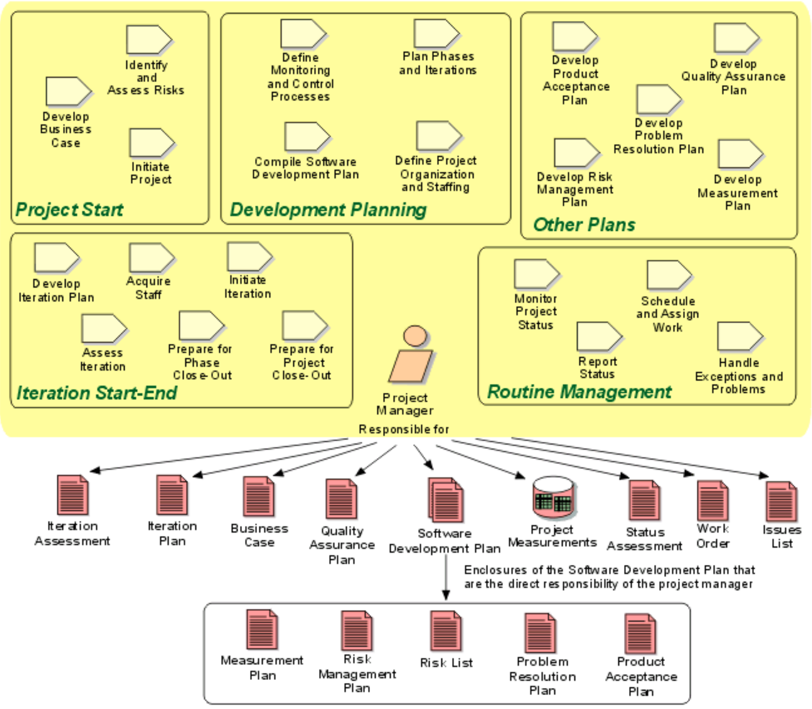


Figure 6. Project Manager Role in RUP

The FEAF Matrix

Figure 3 above offered an overview of the FEAF matrix that describes the FEAF at Level IV detail. The matrix incorporates five perspective rows (i.e., views) Planner, Owner, Designer, Builder, and Subcontractor, and the first three architectural artifacts or product abstraction columns (i.e., what, how, and where) of the Zachman Framework.[7] The

[6] "New Goals in Systems Development: Big Ideas for Better Business" by Dave West and Mike Perrow, in The Rational Edge, January 2003. http://www.therationaledge.com/content/jan_03/feature_article.jsp

[7] The Zachman Framework includes three other columns not incorporated into the Federal Enterprise Architecture Framework at this time, although some agencies may find these other aspects useful for gaining some notional understanding the who, when, and why.

FEAF matrix also refers to the perspectives, or rows, as views to denote the various levels of abstraction. In addition the intersection of perspectives and focuses (columns) are called “models” in the FEAF. The IBM Rational Unified process also incorporates the best practice of providing various levels of abstraction for different stakeholders and needs. In RUP, architecture is defined through various views, each depending on the level of detail necessary for a particular stakeholder. The critical architectural decisions are presented in each view. Models in RUP document all of the decisions made, including the architecturally significant decisions. For example the use case model may include 25 use cases, only 10 of which are architectural significant. The use case view would then just represent those use cases that are important for the Architecture. For the purposes of this paper, FEAF models and RUP architecture views are equivalent. In addition, RUP provides a consistent set of models that tie together all of the architectural elements in the various views.

The Planner and Owner rows focus on the business architecture definition and documentation. When completed, these rows make explicit what the enterprise business is and what information is used to conduct it (i.e., the business models). These first two rows are considered essential and must be completed to develop an architecture description that can be commonly understood and integrated across the Federal Enterprise.

The third, fourth, and fifth rows (i.e., Designer, Builder, Subcontractor) define the design architectures (i.e., data, applications, and technology) that support the business architecture. Appropriate models from these rows are developed depending on the purpose and objectives of the specific architecture effort.

The models defined for each perspective and design architecture intersection are the basis for managing and implementing change in the enterprise in a timely manner. The Framework provides a logical structure for classifying and organizing the kinds of enterprise models that are significant for management and development of the supporting systems.

Perspectives (Rows)

In Figure 3 above, each row represents a total view of the solution from a particular perspective. An upper row does not necessarily represent a more comprehensive understanding of the whole than a lower row. Nor does an upper row decompose into greater detail in a lower row. Each row represents a distinct, unique perspective; however, the deliverables from each row must provide sufficient detail to define the solution at the level of abstraction and must translate to the next lower row explicitly.

Each perspective must take into account the requirements of the other perspectives and the constraints those perspectives impose. The constraints of each perspective are additive. For example, the constraints of higher rows affect the rows below. The constraints of lower rows can, but do not necessarily affect the higher rows. Understanding the requirements and constraints necessitates communication of knowledge and understanding from perspective to perspective.

Planner’s View (Scope) - This represents the first architectural sketches, which depict at the highest level of abstraction the size, shape, partial relationships, and basic purpose of the enterprise. It corresponds to an executive summary for a planner or investor who wants an overview or estimate of the scope of the system, what it would cost, and how it would relate to the general environment in which it will operate.

Owner’s View (Enterprise or Business Model) – The next level of abstraction are the architect's drawings that depict the enterprise from the perspective of the owner. They correspond to the enterprise (business) models, which constitute the designs of the business and show the business entities and processes and how they relate.

Designer’s View (Information Systems Model) – At this level of abstraction the architect’s plans are translated into detail requirement representations from the designer's perspective. They correspond to the system model, designed by a systems analyst who must determine the data elements, logical process flows, and functions that represent business entities and processes.

Builder's View (Technology Model) - The contractor must redraw the architect's plans to represent the builder's perspective, with sufficient detail to understand the constraints of tools, technology, and materials. The builder's plans, constitute yet another level of abstraction and correspond to the technology models, which must adapt the information systems model to the details of the programming languages, input/output (I/O) devices, or other required supporting technology.

Subcontractor View (Detailed Specifications) – The final perspective is the subcontractor's, who works with specifications at the lowest level of abstraction. These correspond to the detailed specifications given to programmers who code individual modules without being concerned with the overall context or structure of the system. Alternatively, they could represent the detailed requirements for various commercial-off-the-shelf (COTS), government-off-the-self (GOTS), or components of modular systems software being procured and implemented rather that built.

Focus (Columns)

The Framework is designed as a matrix. Down the left side are the perspectives denoting the levels of abstraction, across the top are the different focuses or products (i.e., Entities = *what*, Activities = *how*, Locations = *where*) of these perspectives. Each focus asks a question. The way in which the questions are answered depends heavily upon the perspective. In essence, at the intersection of each perspective and focus is a particular view of the enterprise architecture.

Models (RUP Architectural Views)

The success of the Federal Enterprise Architecture depends on managing (enforcing) the development process and implementing the architecture descriptions. Business rules must be enforced consistently from implementation to implementation to coordinate and/or change behavior throughout the enterprise. Models must be defined logically, independent of technology constraints, such that the implementation technology can be changed with minimum disruption and cost. Change must be incorporated as a design and management criteria, such that any aspect of the enterprise can be maintained relevant in a dynamic environment.

Support for FEAF using the IBM Rational Unified Process

It is important to point out a reoccurring theme of the design-oriented architectures. The focus is on three separate areas: Data, Application, and Technology. This idea of separating data from function is not new and has been the standard protocol when using structural analysis and design techniques such as Data flow diagrams, hierarchical decomposed processes, and data matrices that decompose function or process separate from data. Although functional decomposition can be an effective method for analyzing requirements, it can lead to problems when applied to system architecture and design — for example, it often leads to systems that do not scale, have brittle architectures, contain modules that are redundant, and are not reusable across the enterprise.

Today, the vast majority of software systems are constructed using object-oriented methods and programming languages. A gap exists in how systems are implemented versus how business processes are identified and communicated to the system development team. In our experience working with clients, information supplied that is the result of functionally decomposed methods is not usable for systems development as the mapping and context are difficult, at best, to understand and maintain. As a result, we have been left with an ineffective bridge for communication between the development teams and the enterprise architecture team. The Rational Unified Process and the UML offer a bridge over this communication gap. They provide a standard set of processes and notation for describing the high-level business domain as well as the detailed design issues. The techniques and methods are similar for each team and the result is communication that can be interpreted and understood by various stakeholders and team members for their particular needs. This communication bridge across the team is facilitated by a single set of models that are consistent and tie together the architectural views.

IBM Rational Unified Process for Systems Engineering

The Rational Unified Process emphasis is primarily on software systems. Enterprise Architectures involve software, but also hardware, people, and information. This is recognized in the FEAF with its emphasis on data, application, and technology design architectures. Essentially the enterprise organization can be thought of as a system that contains other systems. While RUP does discuss how to represent hardware, people, and information for software applications, it requires enhancements when addressing system concerns. To meet this need, RUP for System Engineering is a RUP plug-in that enhances RUP with a combination of new and improved activities and artifacts. It also provides a set of techniques that reduces the need for functional decomposition, thus leading to system and sub-system specifications that meet the needs of the entire development team. We have not gone into detail in this paper on how to employ some of the RUP SE techniques for EA, rather we have identified the RUP and RUP for System Engineering workflow details and activities that would be used to build an EA.

The table below begins to provide guidance on which aspects of RUP and RUP for System Engineering to use to construct the various models (or RUP Architectural views) of the FEAF matrix. The matrix below provides a brief definition of the architectural view to be captured, how RUP and UML might be used to capture the view, and a RUP workflow and activity reference for more detailed information on using RUP. The architectural views are not disconnected, but are views into a consistent, and implementable set of models.

Federal Enterprise Architecture Framework

Perspectives	**Data Architecture** ***(entities = what*)**	**Application Architecture** ***(activities = how*)**	**Technology Architecture** ***(locations = wher*e)**
Planner ***(scop*e)**	**List of Business Objects** **Definition:** A high level list of business objects (or things, or assets) in which the enterprise is interested. The model defines the scope of subsequent enterprise object models. **IBM Rational Approach:** The Rational Unified Process Business Modeling discipline provides for creating a Domain model, focusing on explaining "things" and products important to the business domain. This in a sense creates a data dictionary to capture all of your business objects as modeling elements for use and reuse. This can be captured using UML as a simple object or class diagram (Appendix Figure 1) without relationships, and if necessary generated into documents. **RUP Reference**: Review the	**List of Business Processes** **Definition:** A high level list of processes that the enterprise performs. The model defines the scope of subsequent enterprise process models **IBM Rational Approach:** Business modeling is an important discipline within the Rational Unified Process. This discipline describes how to develop a vision or mission statement for the organization, and to define the processes, roles, and responsibilities of that organization in a business use-case model and a business object model. The list of business processes (Appendix Figure 2) can be presented using UML with business use case diagrams. A business use case is a sequence of actions a business performs that yields an observable result of value to a particular business actor.	**List of Business Locations** **Definition:** A high level list of locations in which the enterprise operates. The model defines the scope of subsequent location models that are connected by the enterprise **IBM Rational Approach:** This list of business locations (Appendix Figure 3) is captured and presented as a set of localities, which are defined in RUP SE. Localities represent notional locations where processing occurs without tying it to a specific location or piece of hardware. Locality diagrams are depicted as UML deployment diagrams where the nodes are stereotyped as locality. In this particular view connections between localities are not necessary and a list could be generated from the model for reporting. **RUP for Systems**

Perspectives	Data Architecture *(entities = what)*	Application Architecture *(activities = how)*	Technology Architecture *(locations = where)*
	Business Modeling Discipline: Develop a Domain Model Workflow Detail for additional information.	**RUP Reference**: Review the Business Modeling Discipline: Describe Current Business Workflow Detail for additional information.	**Engineering Reference**: Review Analysis & Design Discipline: Synthesize System Architecture Workflow Detail for additional information.
Owner ***(Enterprise)***	**Semantic Model** **Definition:** The Semantic Model is a model of the actual enterprise business objects (i.e., things, assets) that are significant to the enterprise. **IBM Rational Approach:** The Semantic model is essentially a refinement of the planner perspective List of Objects. The owner perspective refines the domain model to include the relationships between business objects. The semantic model (Appendix Figure 4) can be captured using the same types of UML diagrams. **RUP Reference**: Review the Business Modeling Discipline: Develop a Domain Model Workflow Detail for additional information.	**Business Process Model** **Definition:** The Business Process Model shows the actual business processes that the enterprise performs, independent of any system or implementation considerations and organizational constraints. **IBM Rational Approach:** In this cell we further analyze the identified business processes above. This is documented using UML activity diagrams or sequence diagrams (Appendix Figure 5) to model the flows of events, or tasks, performed by various workers. The elements included in the sequence or activity diagrams reflect how the various enterprise resources collaborate to achieve the business use case goal. The elements would be combinations of people, applications, hardware, and data. In addition to the visual UML models, Business use cases maintain a textual specification that provides further understanding of the business process. **RUP Reference**: Review the Business Modeling Discipline: Refine Business Process Definitions Workflow Detail for additional information.	**Business Logistics System** **Definition:** The Business Logistics model captures the locations of the enterprise and their connections (i.e., voice, data, post or truck, rail, ship, etc.). It identifies all of the types of facilities at the nodes like branches, headquarters, warehouses, etc. **IBM Rational Approach:** The localities in the planner perspective are refined with connection information. This is done in using a Locality Diagram (Appendix Figure 6) to show the various localities and their connections. The connection lines are annotated to show how they are achieved (i.e., voice, data, post or truck, rail, ship, etc.). In addition an intra-nodal perspective can also be accomplished with a locality diagram to describe the facilities at each node. **RUP for Systems Engineering Reference**: Review Analysis & Design Discipline: Synthesize System Architecture Workflow Detail for additional information.

Perspectives	**Data Architecture** ***(entities = what)***	**Application Architecture** ***(activities = how)***	**Technology Architecture** ***(locations = where)***
Designer *(information systems)*	**Logical Data Model** **Definition:** The logical data model is a logical representation of the objects of the enterprise about which it records information. It is represented as a fully attributed, keyed, normalized entity relationship model reflecting the intent of the Semantic Model. **IBM Rational Approach:** The Logical Data model is captured by further refinement of the semantic model. UML class diagrams (Appendix Figure 7) are used to further refine the semantic model above. The logical data model class diagram displays the data entities and relationships as well as the data entities attributes with key designations. **RUP Reference**: Review the RUP Analysis & Design Discipline: Analyze Behavior and Database Design Workflow Details for additional information.	**Application Architecture** **Definition:** The Application Architecture model presents the logical systems implementation that supports the business processes. It expresses the human and machine boundaries to the system. **IBM Rational Approach:** The application architecture now develops the architecture for individual applications, or systems, that support business processes. The artifacts presented in the application architecture are those that are architecturally significant. The RUP provides guidance for developing the application architecture in various disciplines and activities. In particular in the Requirements and Analysis & Design disciplines. The application architecture will contain system use cases, and their corresponding analysis realizations. The analysis realizations provide high-level descriptions of the interactions and relationships between application elements. The interactions and relationships are documented using UML interaction diagrams (Sequence Diagrams (Appendix Figure 8) or Collaboration Diagrams) and class diagrams (Appendix Figure 8). The realizations are further developed and detailed in Systems Design. **RUP Reference:** Review the Requirements Discipline: Define the system and Refine the System Definition Workflow Details. Analysis & Design Discipline: Define a	**System Geographic Deployment Architecture** **Definition:** The System Geographic Deployment Architecture is a logical model describing the system implementation of the business logistics system. It describes the types of facilities and controlling software (applications) at nodes and lines of communication between them (examples are processors, operating systems, storage devices, DBMS's, and peripherals/drivers). **IBM Rational Approach:** In this model we now define components derived from various details in other views. The localities are realized by a set of components that consist of hardware, software (applications), or workers. The components are depicted as nodes stereotyped as descriptor-nodes and viewed on UML deployment diagrams (Appendix Figure 9). **RUP for Systems Engineering Reference**: Review Analysis & Design Discipline: Synthesize System Architecture Workflow Detail for additional information.

Perspectives	**Data Architecture** ***(entities = what*****)**	**Application Architecture** ***(activities = how*****)**	**Technology Architecture** ***(locations = where*****)**
		Candidate Architecture and Analyze Behavior Workflow Details	
Builder ***(technology*****)**	**Physical Data Model** **Definition:** The physical data model represents the data model that has been refined to account for the actual database implementation. The physical data model describes structure necessary to support the logical model and is dependent upon the select technology. **IBM Rational Approach:** The creation of a physical data model maps the logical data entities and attributes to physical tables and columns. This is supported by the UML so that a single modeling language is used. The physical data model is expressed using the UML data modeling profile. (Appendix Figure 10) RUP is flexible in allowing you to model the physical data model. Relational models can be captured using the UML profile for data modeling; Object Oriented data stores can be captured using a fully attributed Class diagram. In addition, XML schemas can also be modeled with UML. **RUP Reference:** The RUP Analysis & Design Discipline: Database Design is applicable here.	**Systems Design** **Definition:** The Systems Design defines the methods and their realizations. **IBM Rational Approach:** The System Design further elaborates the analysis realizations in the Application Architecture to provide all of the detail necessary for implementation. The RUP provides detailed guidance how to capture the System design in the Analysis & Design discipline. Specifically Use-Case Design, Subsystem Design, Class Design activities. The artifacts are depicted in terms of sequence and/or collaboration diagrams to describe dynamic interactions between design elements, class diagrams for architecturally significant design classes (Appendix Figure 11), state machines for classes that have significant stateful behavior, component diagrams for architecturally significant software components. **RUP Reference:** The RUP Analysis & Design Discipline: Refine the Architecture and Design Components Workflow Details is applicable to this activity.	**Technology Architecture** **Definition:** The Technology Architecture is the physical representation of the technology environment for the enterprise. It shows the actual hardware and software (Appendix Figure 12) systems at the nodes and lines, including operating systems and middleware. **IBM Rational Approach**: The technology architecture describes the actual physical hardware in the enterprise that will be use to implement the system. It also shows the software systems in the system design allocated to hardware. The RUP provides guidance how to capture this activity in a UML deployment diagram. **RUP Reference**: The RUP Analysis & Design Discipline: Refine the Architecture Workflow Detail is applicable to this activity.
Subcontractor (detailed specifications)	**Data Definition** **Definition:** The definition of all data objects specified by the physical model and would include all the data definition language required for	**Programs** **Defintion:** The application implementation that realizes the System Design.	**Network Architecture** **Definition:** The Network Architecture consists of the specific definition of the node addresses and the line identification.

Perspectives	Data Architecture *(entities = what)*	Application Architecture *(activities = how)*	Technology Architecture *(locations = where)*
	implementation. **IBM Rational Approach:** The data definition is the actual implementation of the Physical model. UML specifications can be translated directly into the implementation (DDL or directly to the database management system). Often the implementation is automatically generated from the physical model.	**IBM Rational Approach:** Each element in the system design is implemented by coding it in a programming language or by using a pre-existing component. Exactly what an element in design corresponds to depends on the programming language. UML specifications used for system design can be translated in to various program languages including: Java, Visual Basic, C++, C#, XML, and various others. In addition, patterns can be employed to help ensure consistency in the implementation. A pattern codifies specific knowledge collected from experience. Patterns provide examples of how good modeling solves real problems, whether you come up with it yourself or you reuse someone else's.	identification. **IBM Rational Approach:** The network Architecture is a refinement of the Technology Architecture UML deployment diagram showing specific addresses and line identification.

Conclusions

The business and design models necessary in establishing and managing enterprise architectures may be accomplished using various techniques and approaches. The IBM Rational Unified Process provides a cohesive set of best practices and methods to build and maintain Enterprise Architectures. The Rational Unified Process ties together the different perspectives with a set of practical activities and artifacts result in a creation of a consistent set of models. Architectural views of the models can be organized into the FEAF matrix. The great advantage of using RUP is that the underlying set of models is consistent and provide for communication across the organization. In addition this set of models is implementable. In essence, using RUP as a process framework for developing enterprise architectures, organizations can effectively capture, review, manage change, and communicate enterprise architectures across the different perspectives and across the organization.

Additional References

Rational Software, “Rational Unified Process for Systems Engineering 1.1”

Rational Software, “Rational Unified Process” Version 2002.05.00

U.S. Department of Defense. *C4ISR Architecture Framework Version 2.0*, December 18, 1997.

Web Sites

1. General Services Administration (GSA), Office of Information Technology

http://www.itpolicy.gsa.gov

2. U.S. Chief Financial Officers (CFO) Council

http://www.financenet.gov/fed/cfo

3. U.S. Chief Information Officers (CIO) Council

http://cio.gov

Appendix

Planner Perspective

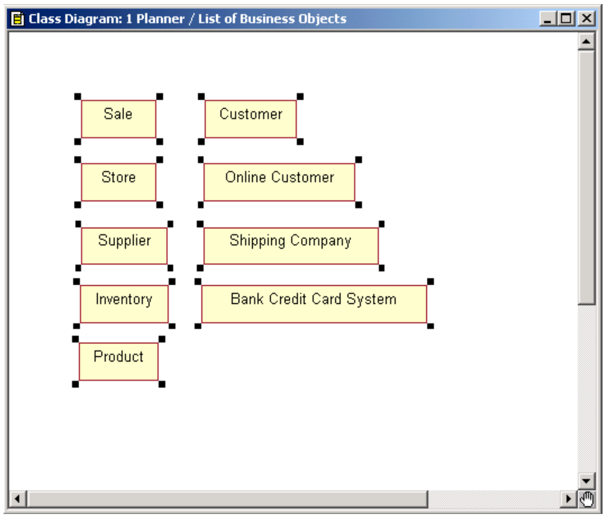


Figure 1. Visual Depiction Of Business Objects

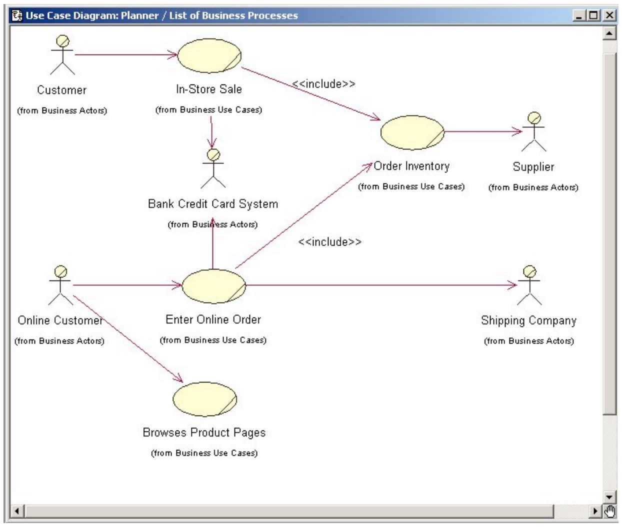


Figure 2. Use Case Model Of Business Processes

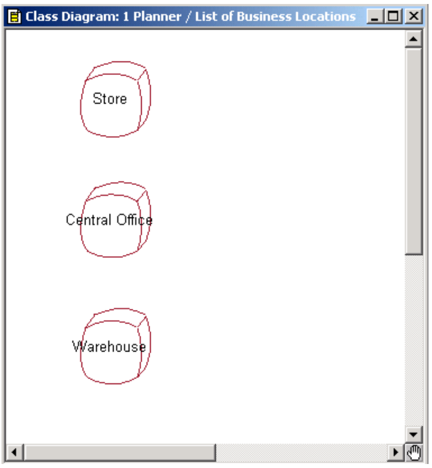


Figure 3. UML Visualization Of Business Locations Using A Custom Icon

Owner Perspective

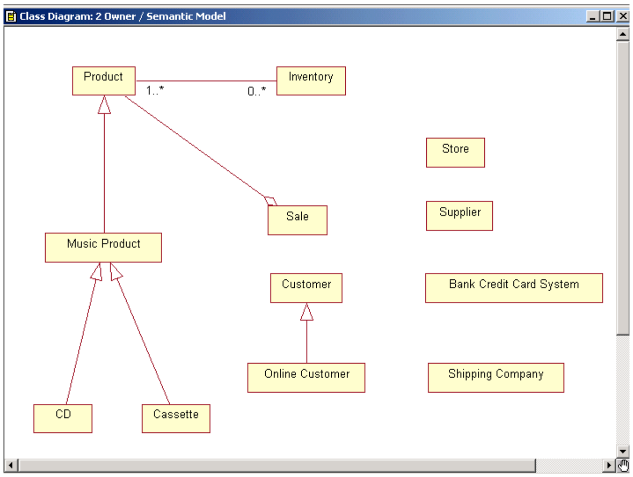


Figure 4. UML Visualization Of A Semantic Model

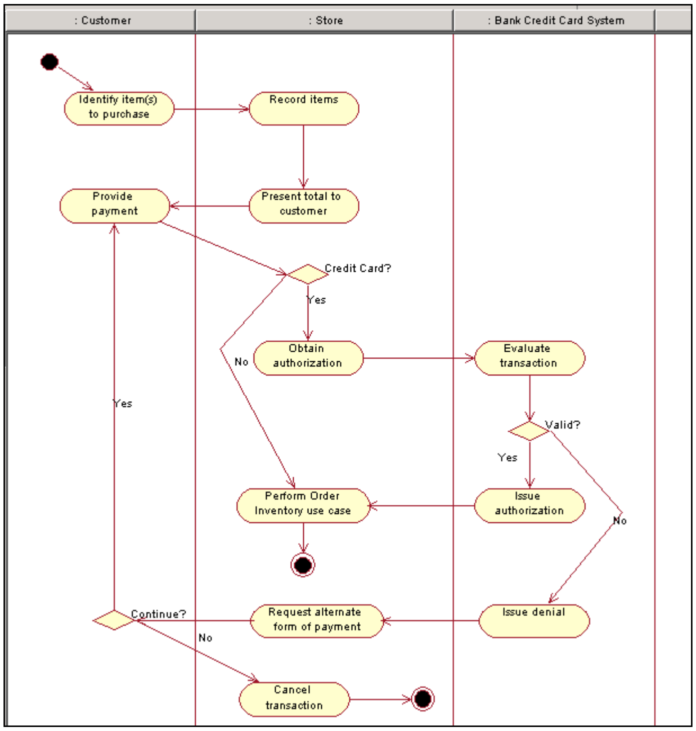


Figure 5. UML Activity Diagram With Vertical Swimlanes

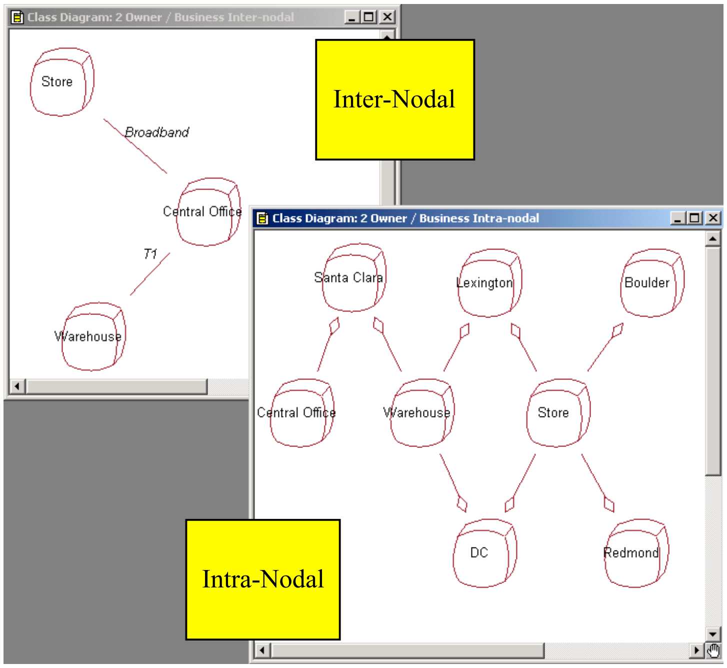


Figure 6. UML Diagram Showing Business Logistics with Inter-Nodal and Intra-Nodal views

Designer Perspective

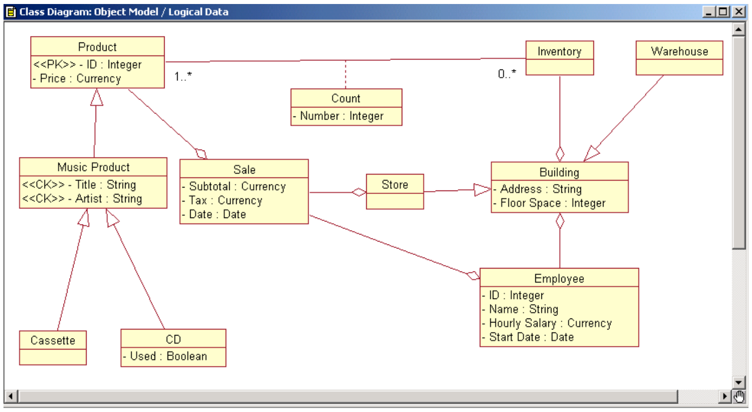


Figure 7. Logical Data Model

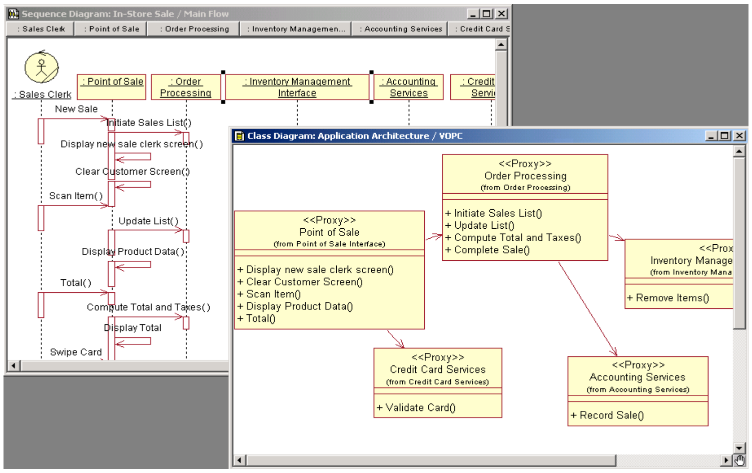


Figure 8. UML Sequence and Class Diagrams

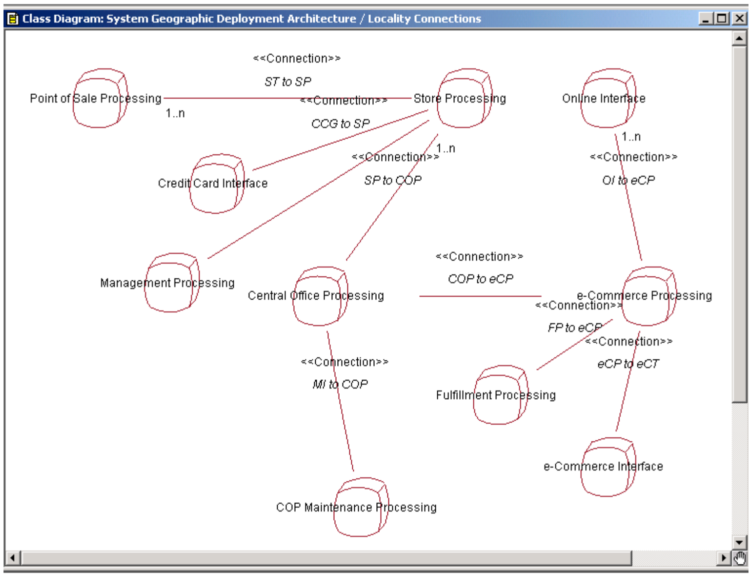


Figure 9. System Geographic Deployment Architecture

Builder Perspective

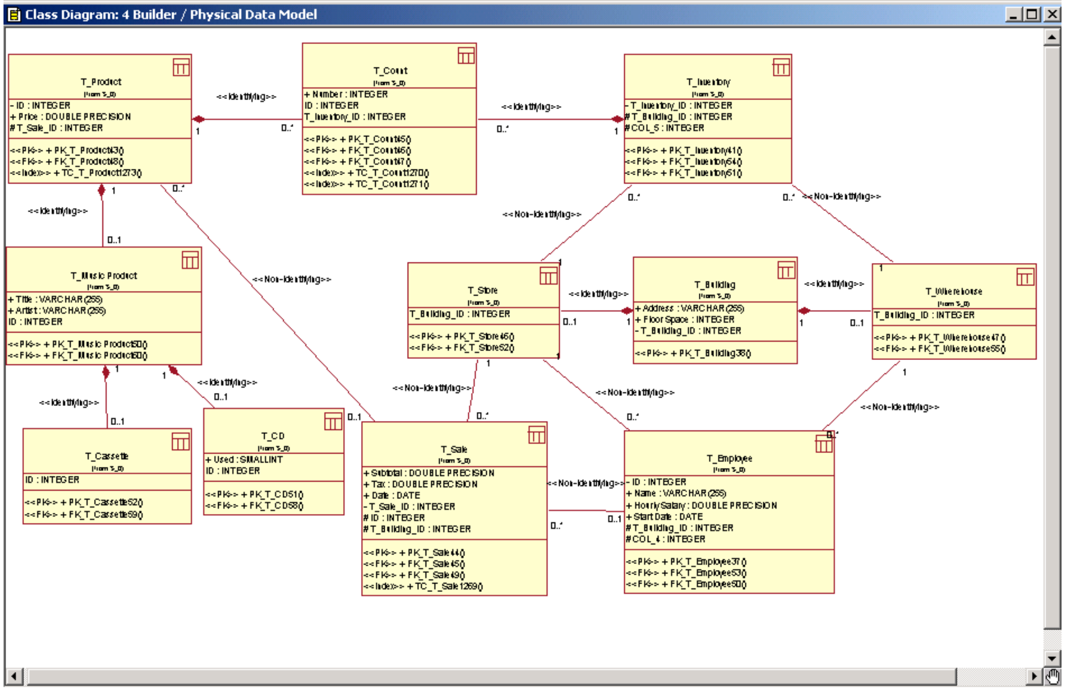


Figure 10. Physical Data Model

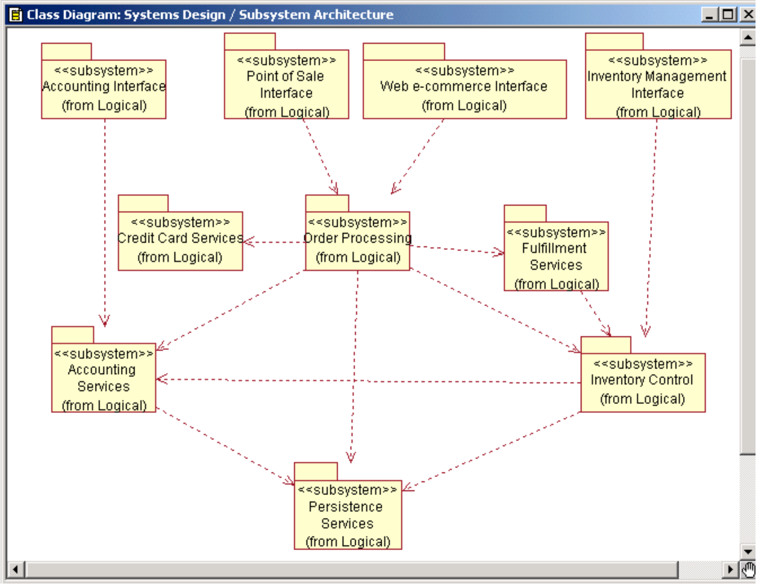


Figure 11. Systems Architecture Displayed Using UML Sub Systems

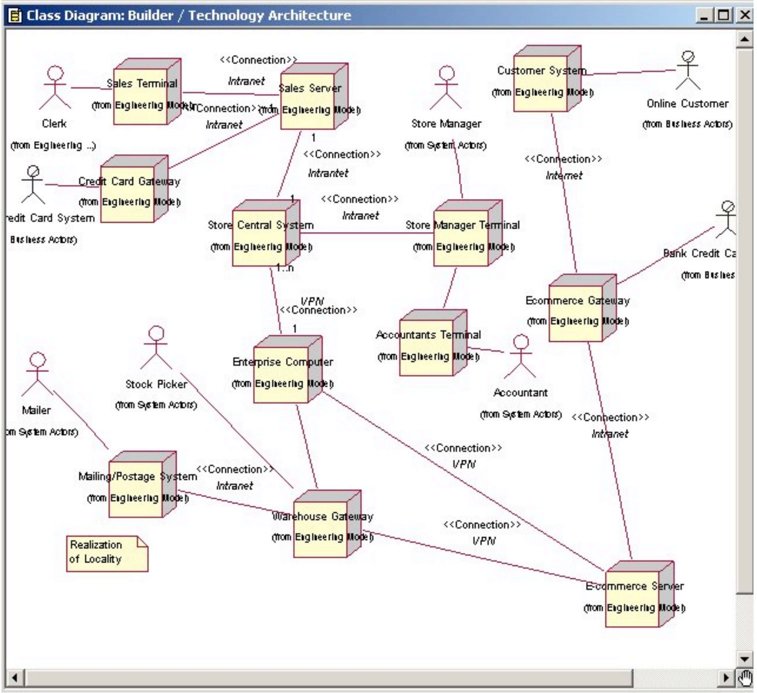


Figure 12. Technology Architecture

IBM software integrated solutions

IBM Rational supports a wealth of other offerings from IBM software. IBM software solutions can give you the power to achieve your priority business and IT goals.

- *DB2® software helps you leverage information with solutions for data enablement, data management, and data distribution.*
- *Lotus® software helps your staff be productive with solutions for authoring, managing, communicating, and sharing knowledge.*
- *Tivoli® software helps you manage the technology that runs your e-business infrastructure.*
- *WebSphere® software helps you extend your existing business-critical processes*
  to the Web.
- *Rational® software helps you improve your software development capability with tools, services, and best practices.*

Rational software from IBM

Rational software from IBM helps organizations create business value by improving their software development capability. The Rational software development platform integrates software engineering best practices, tools, and services. With it, organizations thrive in an on demand world by being more responsive, resilient, and focused. Rational's standards-based, cross-platform solution helps software development teams create and extend business applications, embedded systems and software products. Ninety-eight of the Fortune 100 rely on Rational tools to build better software, faster. Additional information is available at www.rational.com and www.therationaledge.com, the monthly e-zine for the Rational community.

Rational Software Corporation is a wholly owned subsidiary of IBM Corp. (c) Copyright Rational Software Corporation, 2003. All rights reserved.

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Printed in the United States of America.
All Rights Reserved. Made in the U.S.A.

IBM the IBM logo, DB2, Lotus, Tivoli and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Rational, Rational Unified Process, and RUP are trademarks or registered trademarks of Rational Software Corporation in the United States, other countries or both.

Other company, product or service names may be trademarks or service marks of others.

The IBM home page on the Internet can be found at **ibm.com**