Rational Software White Paper

From Inception to Implementation: A Life-cycle Approach to Enterprise Architectures

TP901, 09/02

Rational®
the software development company

Table of Contents

Introduction

In 1996, the U.S. Congress passed the Clinger-Cohen Act, which, among other things, mandated that federal agencies develop and maintain an enterprise information technology (IT) architecture. This act brought focus to a growing concern that government organizations had developed applications that, while partially successful in meeting a narrow set of needs, could not be tied together to meet the needs of a broader set of constituencies. Enterprise Architecture (EA) efforts are intended to enable the interoperability of a broad range of systems to accomplish expanding missions by defining those missions clearly, and by describing how the set of interoperating systems accomplish those missions.[1]

We begin with an overview of what an EA is, including an examination of its benefits, and then we summarize several federal EA frameworks. Next, we discuss why UML (the Unified Modeling Language) is uniquely positioned to describe EA. Finally, we discuss ways that organizations can not only build a good EA, but also use it to drive the software and systems development projects.

What is meant by Enterprise?

An enterprise is a set of resources applied toward a business purpose or mission. Enterprises may be a business or government agency, a part of business or agency, or in some cases they may consist of several businesses or agencies (e.g., the intelligence community, or a bank participating in a broad credit card enterprise). Recently, many managers have begun to recognize the importance of understanding the architecture of their enterprise in a much more precise manner than they have done in the past. This has given rise to a number of different techniques for capturing and modeling enterprises as well as the lines of communication these techniques have formed between their various entities. The trend has also led to new tasks, activities, and functions these entities perform, and the relationships between all these entities must be understood and managed.

What is "Enterprise Architecture"?

An EA is a "blueprint" for an organization that describes all the information used within the enterprise, their relationships, and how they interact to fulfill the mission of the enterprise. Documenting the EA provides the ability to understand and analyze the operation of the business, allowing shortcomings or inconsistencies to be identified and addressed, eliminating costly errors, redundancies, and inefficiencies.

Why are Enterprise Architectures important?

Even people, who have worked in an enterprise for many years may not know how it actually works, let alone how it is supposed to work. Naturally, if an organization does not understand its mission and how that mission is carried out, it will never be successful in achieving the mission. An EA offers a remedy because it documents what the organization does and who this activity benefits.

An EA is especially important for understanding those information systems that directly support the organization's mission -- for example, an insurance company has many descriptions of the customer throughout the policy, claims and renewals systems. Understanding the implications of adding a new customer system, and deciding what the dependencies are between those systems provides the insurances company management with the information necessary to make the right decisions. Without an EA, it is difficult or impossible to tie systems together in a meaningful way. Consequently, newly developed systems form new "stovepipes" rather than being properly integrated into the whole systems architecture. Eventually, as many organizations have come to realize, it becomes impossible to maintain a large number of different systems, each of which perform similar functions. There are many negative implications of having overlapping stovepipe systems including cost of maintenance, increased complexity and cost of testing. Without an EA, there is no way to sensibly understand how to integrate the data and business logic from these disparate systems into an intelligent body of information on which corporate decisions can be based. Moreover, any attempt to do so quickly uncovers the fact that much data is duplicated across these several stovepiped systems, or even worse, it may be subtly inconsistent as well.

Examples of these kinds of problems abound. Any organization that merges with another typically must merge many similar but incompatible systems, ranging from accounting to payroll to customer relationship management.

[1] "Enterprise Architecture Use across the Federal Government can be Improved," General Accounting Office, February 2002, GAO-02-6.

Our federal and state governments, currently challenged to streamline the communication of intelligence information across agencies, face a problem of unprecedented complexity. And in healthcare, the need to share information between insurers, patients, employers, and healthcare providers drives up costs even while these parties try to tie their incompatible systems together to bring costs down.

An EA, combined with processes for its development and evolution, provides a mechanism that enables communication about the essential elements and functioning of the enterprise. It yields centralized, stable, and consistent information about the enterprise environment. In the car rental agency example, an EA would provide information about it's the agency's more lucrative markets and help executives understand how the company's resources are meeting the customer needs in those locations. The existence of the EA makes it much easier for an organization to respond to the forces of change. Reducing the duplication and inconsistencies in the information dramatically improves the ROI made on the organization's various IT implementations. When the information about the enterprise is more precise and of higher quality, better decisions can be made.

Enterprise Architecture in the United States Federal Government

While federal EA efforts are by no means indicative of all the industry efforts to standardize and benefit from EA, federal EA efforts are 1) important because of the large scope of the targeted enterprises and 2) useful because these projects offer good object lessons for further analysis.

DoDAF

The **Department of Defence Architectural Framework (DoDAF)** is focused on enabling the interoperation of disparate systems used in support of military missions. The framework is intended to clearly relate the objectives of information technology with the performance of the core mission. As part of the development of DoD enterprise architecture, information technology objectives will be correlated with business cases for IT investments.

DoDAF and in particular C4ISR which provides specific support for DoDAF with respect to command, control, communications, computers, intelligence, surveillance and reconnaissance has a number of characteristics that make it useful :

- Create mission systems that work effectively together.
- Enable owners and investors to make buy, build and reuse decisions.
- Make design decisions about the where functionality resides throughout the systems architecture.
- Provide a common language to describe the system for inter mission integration.
- Allow larges groups of stakeholders to work effectively on large complex systems.

For more information about DoDAF, including how Rational Software uses UML to support the DoDAF, visit www.rational.com.

FEAF

The **Federal Enterprise Architecture Framework (FEAF)** was established in 1999 by the Chief Information Officers (CIO) for each federal agency in response to the Clinger-Cohen legislation. The purpose of the FEAF is to facilitate shared development of common processes and information among federal agencies and other government agencies.

According to FEAF, it allows 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 cost effectively.

The FEAF, like other frameworks, is essentially a guide for collecting common architecture information and building a repository to store this information.

For more information about FEAF, including how Rational Software uses UML to support the FEAF visit www.rational.com.

TEAF

Similar to the objectives of FEAF, the **Treasury Enterprise Architecture Framework (TEAF)** is a framework developed by the Department of the Treasury and its agencies (e.g. IRS, Customs, etc). It has goals similar to FEAF, but is more specialized to the Department of the Treasury's domain.

Enterprise Architectures in other contexts

EA is emerging as a significant IT trend in the government sector, including federal, state, and local governments. In addition, a similar trend is emerging in industry as companies recognize that the IT spending of the past was not well-aligned with organizational objectives, creating islands of information and systems that fail to meet their intended objectives.

Representing Enterprise Architectures

Having decided that an EA is necessary, how does one go about creating and maintaining one? Traditionally, first steps involve writing and preparing slide presentations, and this approach is certainly better than nothing. However, such documents are surprisingly difficult to maintain: there is no easy way to keep the EA documents up to date as the business and the elements within it change. As soon as they become out of date, the questions these documents were designed to answer are no relevant to the EA initiative under way. Consequently, the documents lose value, and interest in maintaining them wanes.

Moreover, most companies who set out to document their EA find no standard methodologies to help them form a consistent and understandable notation. Finally, they find pictorial descriptions to be essential for effective communication of concepts, but the pictures that are developed are imprecise and vague. A better way is needed.

The Unified Modeling Language (UML)

The UML is a visual modeling language developed initially to combat these kinds of difficulties in the general arena of software development. Soon after the initial release of the UML, the software development community realized its potential for much more than visual descriptions of software. In fact, its applicability extends to many domains. An early discovery was its ease of use for business modeling, data modeling, and system modeling -- in other words, for developing precise and complete visual descriptions of the elements of an EA. A principal reason for this was the availability of good graphical modeling tools based on the UML that may be used to create and maintain the architecture and its semantics in a central, browsable framework.

UML is a standard specification language maintained by the Object Management Group profiles. One of the principle strengths of the UML is its definition of a single meta model. When models are developed using the UML, they are consistent in their definition, and thus support standard traceability and linking. This eliminates the time required to understand relationships between things created in different contexts.

UML-based tools also present the user with another key capability: formal traceability, or linking, of various artifacts to one another. This is perhaps these tools most important feature, as it permits a formal process to be developed for maintaining the various documents describing the EA and the business requirements that EA realizes. The same modeling technique can be used throughout the enterprise to link process, data and control model elements together. When external factors demand changes to a business level requirement, one can trace the change through the interlinked system of models, starting with the EA, to learn 1) what would have to be modified in individual organizations or systems; 2) how much these changes might cost; and 3) what the impact of this change would be on other systems, markets, and customers. This traceability brings a level of predictability to enterprise development that was not possible before. Stockholders and other stakeholders who are used to seeing vague business predictions change every quarter can begin to hold the company to a higher standard as its ability to maintain a high-quality EA model matures. Once you link the EA to the actual software systems development projects, it is possible to keep the EA up-to-date and measure organization progress against this organization "yard stick".

Key Benefits of the UML

As a language designed for visual representation of models, the UML offers business an excellent means to assess, build, and deploy IT systems that support the core mission of an organization in the most cost effective and accurate way possible. Consider the following strengths of the UML:

- UML is an industry standard notation for developing and documenting systems of all kinds, all the way from capturing the business or operational process to the EA to the specification of the systems that support the EA.
- By providing a single rigorous notation for documenting all the disparate views of the EA, UML enables relationships to be easily represented, communicated and understood. This facilitates communication and uncovers errors in understanding easier than other approaches.
- Since UML can be used to define the EA, as well as the supporting system, it supports the ability to assess compliance of a system with the defined EA.

From Inception to Implementation: Rational's Life-cycle Approach

Defining an Enterprise Architecture

As the original inventor and advocate developer of the UML and the pioneer of its application to business process modeling and systems engineering, Rational Software is uniquely positioned to deliver EA design and implementation techniques to today's businesses. The increasing interest in EA as a blueprint for enabling improved system interoperability suggests a strong need for best practices and knowledge transfer in the EA domain. To date, however, EA has been represented using an odd assemblage of inconsistent techniques that result in inconsistent and erroneous enterprise architectures that contribute little to an organization's real mission.

Use cases provide a powerful antidote to this problem. A proven technique for representing the structure of interaction and results within a software system, use cases have helped Rational's customers understand the organizational mission and what the organization does to achieve it.

Use cases can help an organization understand how best to allocate mission-fulfilling behavior to elements within the organization. This provides a consistent thread through the entire organization, a way of tying actions to value provided. This use-case driven approach breaks through the barriers created in a typical functional decomposition approach, where the actions become ends to themselves, disconnected from the mission, and the result is often a precise collection of flawed processes.

Rational's solution for EA is also centered around an incremental, iterative approach to developing the EA. Often, EA efforts fail to achieve their objectives because organizations try to define everything, all at once. Many organizations in both government and civilian sectors have discovered that an iterative, incremental approach that is driven by risk-mitigation achieves better results by being more adaptive to inevitable change. The Rational Unified Process® platform codifies these best practices, allowing the whole development team to work from a single, documented development framework.

Furthermore, Rational's tools are designed to automate the RUP, and they are ideally suited to EA development. Using Rational RequisitePro® to manage EA requirements and Rational Rose® to visually represent elements in the EA, Rational provides support for all aspects of EA. In addition, Rational provides benefits in managing versions of the EA as it evolves throughout the project via Rational ClearQuest® and Rational ClearCase® Rational's toolset also functions across geographically disperse worksites, enabling multi-site collaboration of colleagues in a controlled and consistent way.

Rational® ProjectConsole measures the progress of teams as they collaborate on the development of the EA. Rational ProjectConsole collects both standard and custom metrics from your Rational Suite® development platform and third-party products, and presents the results graphically so that you can easily assess project progress and quality. This allows you to better predict which areas will require special attention and where to focus scarce resources for staying on schedule. ProjectConsole also enables you to make decisions based on quantitative analysis, including accurate and accessible project status information for better anticipating -- and avoiding -- problems or delays. Better project predictability helps you to keep projects on schedule and on budget.

Implementing an Enterprise Architecture

An Enterprise Architecture serves a far greater purpose than simple architecture domentation. As companies continue to seek creative methods for "doing more with less," a properly implemented EA will drive prudent investments in information systems and technology. Rational Software enables a smooth transition from the EA to the definition and development of systems that support the organizational mission that the EA describes.

Rational Software has more than 22 years of experience helping customers deliver complex, enterprise-class applications using an iterative, risk-mitigating approach that improves the ability to predict the delivery. Rational's development tools amplify results using this approach, providing support for requirements capture and management, change management, visual modeling, project and program status assessment and control, and automated testing. Integration between tools and process enables greater predictability in delivering EA-driven results at lower cost.

For more information on how Rational Software can help you develop and implement your EA visit www.rational.com

Rational Software
Dual Headquarters:
Rational Software
18880 Homestead Road
Cupertino, CA 95014
Tel: (408) 863-9900

Rational Software
20 Maguire Road
Lexington, MA 02421
Tel: (781) 676-2400

Toll-free: (800) 728-1212
E-mail: info@rational.com
Web: www.rational.com
International Locations: www.rational.com/worldwide