# Creating an Enterprise Architecture on a Shoestring: A Light Weight Approach to Enterprise Architecture

James J. Cusick
Wolters Kluwer, Corporate Legal Services
New York, New York
j.cusick@computer.org

William Tepfenhart
Monmouth University
Monmouth, NJ
wtepfenhart@monmouth.com

## *Abstract*

*The creation of an Enterprise Architecture requires dedicated resources, or does it? At CT Corporation an Enterprise Architecture was developed using part time resources over a one year period. This architecture fulfilled the goals set out in the team's charter and covered the applications, functionality, and data that were embodied in the systems that run the business. This paper describes the approach taken to develop this architecture, its uses, and the remaining gaps in its development.*

## 1 Introduction

When most people think about an Enterprise Architecture (EA) they sigh and say that would be nice but who has the time to do everything required to build one. Especially in these days of reduced IT budgets, off-shoring, and lean and mean business environments an EA can be seen as a luxury. At CT Corporation, a legal services firm owned by the publisher Wolters Kluwer, our technology leadership saw it as necessary to build an EA and to put in place a process to keep applications compliant to its framework. This effort was done on a very limited resource budget using managers and architects across the company to collaborate on the creation of the EA and the analysis and solution of related architecture problems in a part-time committee format. The efforts of this committee translated into several key architecture artifacts that illuminated the systems architecture of the company and provided valuable insights into relationships among these systems and allowed for forward planning.

This paper covers the genesis of this effort, some key best practices used in putting the work together, the major outputs of the process, and some gaps and next steps for the effort. The approach used is simple, flexible, and has provided useful knowledge for the company and can be replicated by anyone seeking similar results.

# 2 Starting Points

The Enterprise Architecture effort began when the CTO and his senior managers developed the idea of forming an Application Architecture Committee (AAC). This architecture committee was given a charter to work on cross systems problems in a rapid response mode and to develop long range systems planning and an Enterprise Architecture. The motivations for this work came from the notion that a well constructed systems architecture increases the level of flexibility and control over the applications, allows for greater flexibility in providing solutions, and reduces time to market for our electronic products and their supporting systems. The goal of the architecture committee was to extend our architecture to support those new efforts, and to establish stable, supportable, versatile, compatible, and consistent base structures upon which to build our applications.

This committee was chartered with defining key information architectures focusing on application architecture and interoperability, as well as the strategy for their implementation within and external to our ongoing project commitments. The major deliverables of this committee included the establishment and communication of a consistent architecture on which all applications will be built and maintained. This committee was also empowered to establish and execute a process to ensure that this goal will be met.

In regards to the Enterprise Architecture, the architecture committee was specifically asked to develop a strategic architecture for components shared by various business and IT functions which ensured:

- A long term application architecture design strategy;
- Accurate and consistent data across the enterprise;
- Secure and protected information;
- Data and technology shared across the enterprise and facilities business communication and integration;
- Architecture solutions which are widely accepted and/or are industry standards.

All of the above were to be accomplished in a cost-effective manner proving value to the business

In addition, the architecture committee was asked to fulfill the following points:

- Short term design definition, including potential incremental steps to achieve long term strategy and/or expediencies as demanded by business needs.
- Transition steps for implementing the long term strategy as and when it differs from the short term objectives.

The architecture committee at its inception consisted of a group of Technical Managers representing the key IT functions including web development, ERP development, database, and infrastructure. The team was immediately given multiple technical problems to work on and resolve. These cross systems issues were tackled by pulling together key requirements, developing a proof of concept and turning over the approach to a development team to carry through to production. This pattern repeated for many problems given to the committee while some problems required other solutions such as a white paper or a technical brief. However, on the EA side of the equation we began by studying best practices in the field and developing a plan for creating our architecture. These best practices are briefly reviewed below and the plan we adopted is presented.

# 3 Best Practices

Enterprise Architecture provides a common basis for understanding how systems are structured to meet strategic objectives across organizational boundaries. Typically, EA is organized into layers (Spewak, 1992):

- Business Architecture
- Application Architecture
- Technical or Infrastructure Architecture
- Information Architecture

While software architecture has been described as the composition of components, connections, and their constraints (Garlan, 1993) which define a system, Enterprise Architecture reflects this composition at a system-of-systems level. The process of conducting architecture describes the components that make up a system, the mechanisms for communication between processes, the information content or messages to be transmitted between processes, the input into the system, the output from the system, and the hardware that the system will run on. For Enterprise Architecture the architecture is the high-level definition of the structure of a system, which is comprised of parts, their interrelationships, and externally visible properties (Bredemeyer, 2005).

The goal of developing an Enterprise Architecture is to define a vision that bridges the extant status of the firm ("Where it is?") and its projected future status ("Where it wants to be?") (Malhotra, 96). To get to this vision you need a plan of attack. A typical approach that comprises two phases of Planning and then Execution is provided by Spewak (1992):

1. Planning the work
    a. Determine Scope and Objectives
    b. Create a vision
    c. Adapt a methodology
    d. Assemble the team
    e. Prepare project plan
2. Conducting the work
    a. Develop Business Model
    b. Conduct Enterprise Survey
    c. Document Current State of Systems and Technology
    d. Develop Data Architecture
    e. Develop Applications Architecture
    f. Conduct Migration Planning

While this planning structure is sensible and appropriate to the work, we modified the approach to meet our local needs and to account for practicalities of time and access to information. As a result, the first major deviation we made to this standard approach was to skip the business process layer analysis. We did not feel that this step was attainable with the team of technical individuals we had in place and asking for business experts to be involved in the process up front when we had nothing else in place did not make sense to us. Our assumption was that we should document the existing systems and their capabilities first and then develop the forward looking business needs later. Thus our actual plan developed as follows:

1. Postpone the business process layer analysis
2. Document the existing systems
3. Create Input/Output characterization for all production systems
4. Develop high level context diagram of system interaction
5. Document functional and component composition of all major applications
6. Develop baseline cross-systems functional architecture

7. Analyze existing architecture against forward looking requirements
8. Develop target architecture(s)
9. Develop migration plans
10. Implement migration plans

In addition, we decided to build out our forward looking models incrementally by first producing a 6 month view, then a 12 month view, and finally an 18 month view.

# 4 The Application Catalog

The first step in the process was to develop an application catalog. We developed a listing of all production systems by taking our operational metrics reporting tool and creating a list of all systems currently in production and being tracked. Using this list we then assigned each system to one of the team members who was to develop a brief description of the system that included an input/output diagram such as the one in Figure 1. This diagram includes the name, connection type, and periodicity of each information transmission for all interfaces of the system. These diagrams are simple, straight forward to develop and very powerful. People can quickly understand the diagram and it can provide valuable discussion points for those interested in the system. As will be seen in the next step, these diagrams also allow for the creation of a key follow on artifact in the process of establishing an Enterprise Architecture. Our Application Catalog contained a one page entry for all 31 major applications in the company. To develop the catalog took roughly 6 weeks calendar time and required about a dozen people who were expert on these systems to document this high level view. This usually took about an hour. We conducted multiple reviews on the catalog to make sure everyone agreed on the content.

Today we perform ongoing maintenance on the catalog whenever there is an addition, change, or retirement of any system. This maintenance work normally takes just a few minutes time by the architect or a committee member to perform.

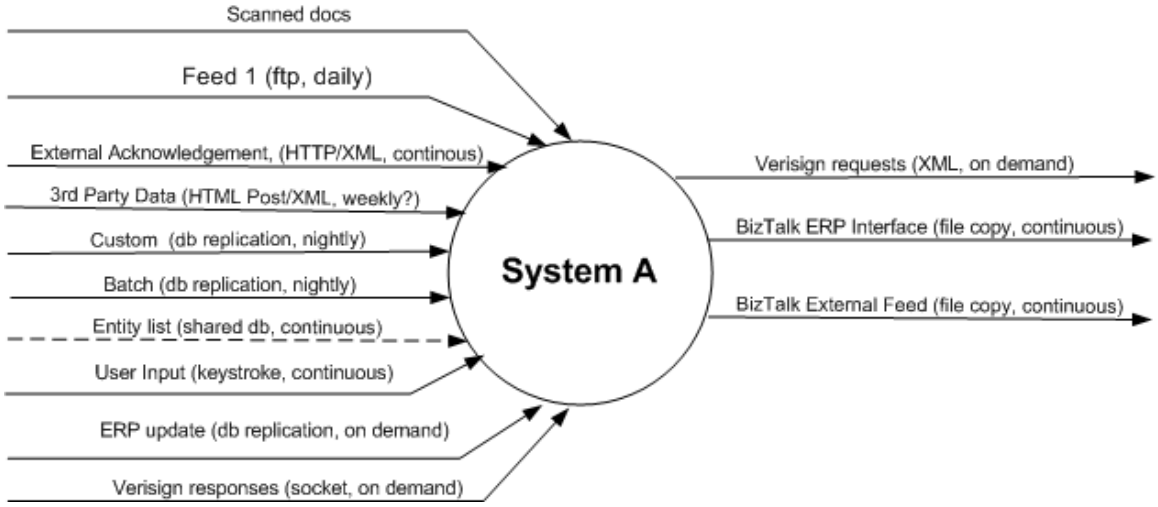


*Figure 1 – Sample Input/Output Diagram for an Application*

# 5 The Enterprise Map

Once the Application Catalog was complete we simply analyzed all the ins and outs of each application and began to develop a holistic system-of-systems view of the application suite. Quite literally, we laid out all the major applications on a white board and then went through each interface and drew them in so as to connect all the ins and outs of the catalog. The resultant drawing is seen in Figure 2. This diagram shows external and internal users and which systems they access. It also segregates customer facing systems from backend systems as well as external systems. The external systems are not found in our application catalog but could be added.

The benefit of this diagram should be obvious to anyone who has ever walked into a company and tried to get a handle on how things were connected. As soon as we published this diagram we saw it printed out and pinned up on cube walls over the desks of project managers, testers, and programmers alike. In addition, as the company engaged in various acquisitions, partnerships, and outsourcings, this diagram and its associated material became vital means of communicating about what systems did what and where new connections could or should be made. Creating this diagram was an “ah ha” moment and it continues to provide revelations to people. Along with the application catalog this diagram and its variants have proved irreplaceable and invaluable.

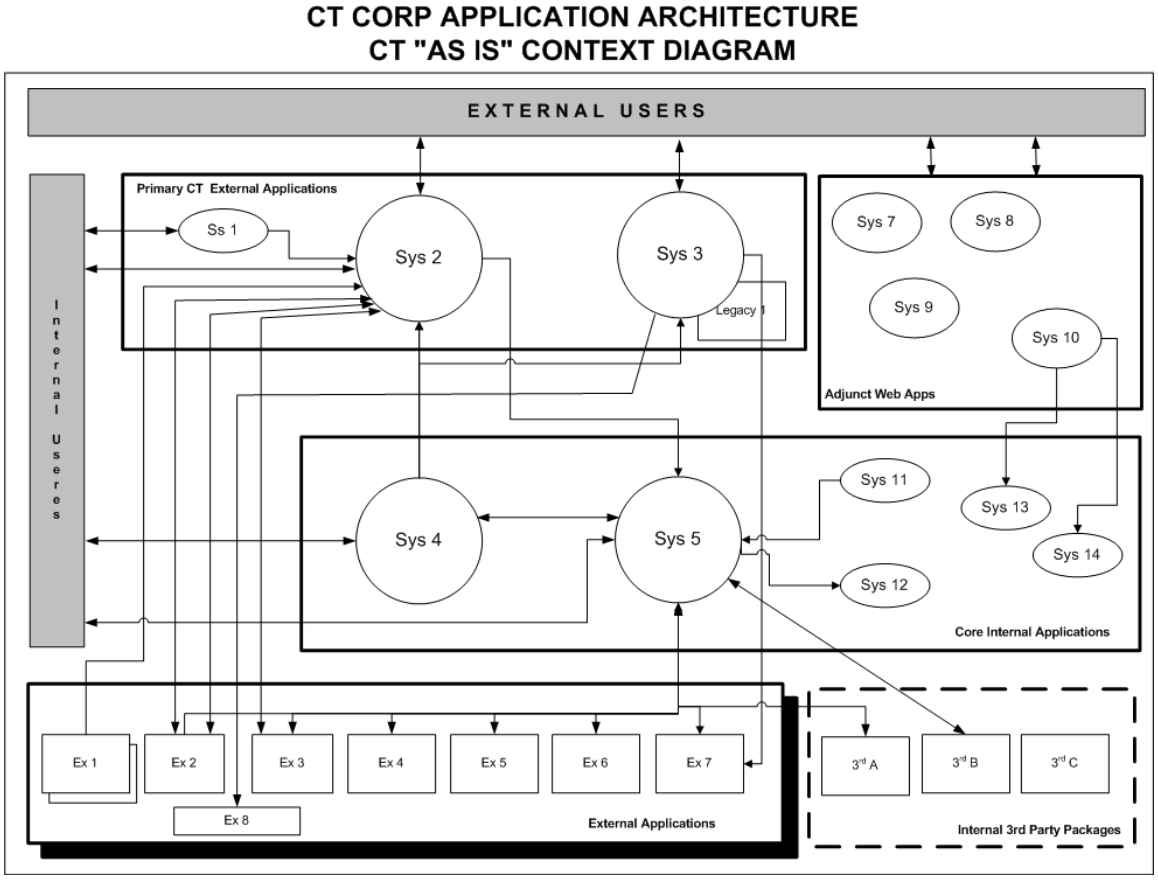


*Figure 2 – The Enterprise Architecture Context Diagram*

# 6 The Functional Description

After developing the Application Catalog and the Enterprise Architecture Context diagram we turned our attention to the Functional Architecture. Most of our systems were rather well documented so getting the information required was not an expensive proposition but there was time involved in gathering the required data. First we created a template for capturing and organizing the data. This template included the following categories:

- Application
- Functional
- Components
- Technical Classification
- Type Of Technology
- Technology
- Version Of Technology
- Comments
- System
- Author

We then assigned each system to a particular application expert to document the functions, components, and related information. We tracked the name of the author in case we needed to go back for clarifications later. As a result of this process we had a detailed listing of all the functions within our suite of applications. This comprised numerous spreadsheets with hundreds of rows of information. We then analyzed the functional makeup of these various systems and looked for themes or patterns or overlaps. When the analysis was complete we ended up with a diagram that held 50 functional categories in 10 groups. This resulted in another "ah ha" moment for many people on the team as the analysis really succeed in boiling down the vast array of complex capabilities into a very succinct set of discrete functionalities. We continued our analysis by developing a detailed document that textually described each of these 50 functional areas and went into the next level of detail. This document is currently only 25 pages long and provides a top to bottom description of the sum total of our enterprise system capabilities from an end user perspective.

Developing this functional analysis has been the most time consuming effort we have attempted. Once again, we utilized about a dozen key technical and application specialists to develop the initial functional listings using the templates. This took approximately 3 months calendar time. The extended time reflects the complexity of the task and also the priority. Everyone involved had full time jobs working on production applications and could only spend sporadic time on the documentation effort. Similarly, the summary document describing all these functions required roughly 4 months to get to its current version which is about 95% complete. Likewise, the same experts were involved in this writing task and progress has often been slowed by other commitments. Nevertheless this document is a valuable touchstone for anyone interested in how these applications work and what they do.

# 7 Solving Problems using the EA

In addition to the Enterprise Architecture development the architecture committee was constantly called on for problem resolution. These problems came from 3 sources, senior management, the committee itself, or the development community. Problems were outlined on a problem statement form and reviewed by management to determine if the problem made sense for the committee to work on. Such a problem might be how to accomplish new functionality on a system with new technology or it might be an interoperability question that needed a solution. Once the problem

was defined the committee would establish a “strike team” to attack the problem and the goal would be to quickly produce a solution.

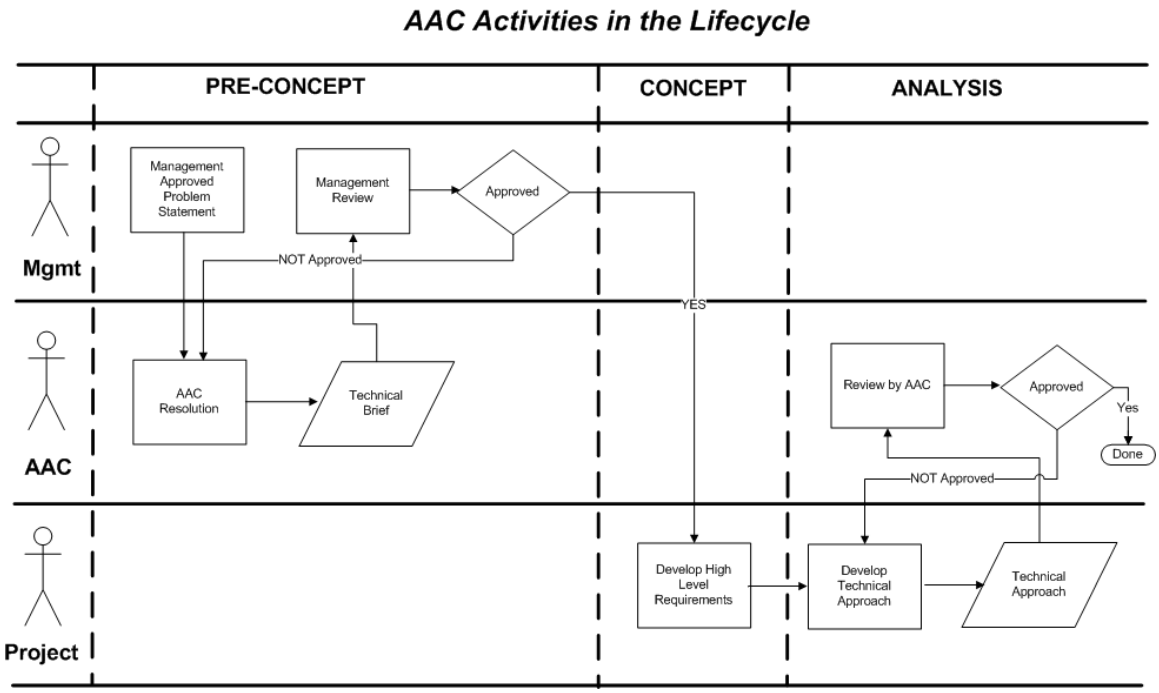


*Figure 3 – Architecture Committee Process for Handling Architecture Development*

This work of problem solving relates directly to the Enterprise Architecture. The process depicted in Figure 3 below indicates the approach taken to insure that solutions adhere to the Enterprise Architecture. When the architecture committee would be given a problem they would solve it in compliance with the Enterprise Architecture and its goals and future defined state. Thus no rogue solution can enter the enterprise environment. As the process shows, the solution pattern is handed off to the development team to take to production as the committee is only a prototyping body. However, once the development team defines the detailed technical approach it comes back to the architecture committee for review and comparison with the Enterprise Architecture.

# 8 Gaps

There are some clear gaps in our approach when compared with industry best practices for Enterprise Architectures. We have noted one key limitation earlier concerning not attacking the business model as of yet. The result of this is an emphasis on the technical elements of the business at the expense of the business structure and processes.

We also have compared ourselves to a set of Enterprise Architecture Best Practices provided by the Systems and Software Productivity Consortium (www.software.org). Table 1 below indicates the best practices recommended by the Consortium and we have indicated which of those, by self-evaluation, we believe we have accomplished with our approach. This table points out gap areas.

Two gap areas, understanding customer needs and including all relevant stakeholders in the architecture process, can be filled-in over time without expensive effort. The documented architecture reflects the results of an organic, rather than planned, growth of the enterprise in response to individual customer needs. Changes to the architecture as a result of new customer

needs or refinements of existing services shall provide the understanding of customer needs over time. Tracking the sources of requests for changes shall identify all relevant stakeholders as well.

We stated that we have not gained an explicit commitment by all individuals involved and, hence, identified that as a gap in meeting best practices. However, one could argue that the widespread use of the "as is" context diagram by programmers, testers, and project managers indicates the grass-roots level has recognized the value of the results produced. It is expected that they will support the maintenance of this effort in the future.

The last major gap concerns evaluation of the architecture. The "as is" architecture exists and is fulfilling the functionality required of it. As new needs for additional capabilities emerge over time, the strengths and weaknesses of this architecture shall be exposed. Admittedly, this is a "trial by fire" approach for evaluating an architecture. However, the existence of the architecture allows forward-looking developers to assess the difficulty of incorporating new capabilities.

According to our rough analysis we have achieved 58% compliance with this scorecard. We find that to be within acceptable limits when compared with the limited resources we dedicated to the effort. This also gives us a yardstick to improve against.

| EA Best Practice | Accomplished |
|---|---|
| Obtain management commitment and support | Yes |
| Understand customer needs | No |
| Ensure architecture is business-driven | Yes (by default) |
| Describe the architecture in multiple views related to stakeholder concerns | Yes |
| Include all relevant stakeholders in the architecture process | No |
| Gain commitment at the grass-roots level | No |
| Preserve flexibility Simplify the architecture | Yes |
| Unify multiple enterprise architecture efforts | N/A |
| Communicate architectural decisions | Yes |
| Evaluate the architecture | No |
| Use frameworks wisely | N/A |
| Build in evolvability | Yes |
| Commit resources to develop and maintain the Enterprise Architecture | Sort of |
| Use the EA to guide and constrain projects | Yes |
| Perform EA in a manner consistent with corporate culture | Yes |
| Balance scope of the enterprise versus level of detail versus resources committee | Yes |

*Table 1 – Enterprise Best Practices Scorecard*

# 9 Next Steps

There are a variety of areas we plan to pursue in the future using our current work as a spring board. Among them are:

- Development of an enterprise data model
- Application of Product Line software planning
- Defining and maintaining Application Architecture standards and ensuring full compliance
- Measure design time by making use of standard architecture patterns
- Measure the reduced architecture conflict among parallel projects

# 10 Conclusions

This effort started with a management challenge to do two things in parallel: solve real architecture problems and develop an Enterprise Architecture. We accomplished both in a very cost effective manner. The approach we used is simple, direct, and effective. Applying an approach like the one we have presented can allow a small organization or one with few resources to create a useful, flexible, and extensible Enterprise Architecture in a matter of months. While there are some limitations to this approach, as outlined above, the benefits of knowing the pattern of one's applications, how they interrelate, and what they are capable of represents a powerful set of abstractions. We look forward to extending and improving our work on Enterprise Architectures and finding new ways to return value to our business.

# 11 Acknowledgements

This work was made possible by the leadership provided by Chris Jutkiewicz in forming the architecture committee and requesting the Enterprise Architecture. Along with him, his senior staff developed the charter for this work and oversaw its success, they include Venkatesh Ramakrishnan, Al Esposito, Galina Vaynerman, and Paul Wanuga. In addition, the members of the committee and the many experts who provided input are all responsible for this work especially Kevin Lane, Alpana Prasad, Vic Tagliaferro, Peter Backes, Michael Kezerashvili and John Fiorillo.

# 12 References

1. Spewak, S., **Enterprise Architecture Planning, Developing a Blueprint for Data, Applications, and Technology**, Wiley-QED, New York, 1992.

2. Garlan, D., and Shaw, M., *"An Introduction to Software Architecture*", **Advances in Software Engineering**, Volume I, eds., Ambriola, V., and Tortora, G., World Scientific Publishing Co., New Jersey, 1993.

3. Bredemeyer Consulting, "*Enterprise Architecture Basics"*, [WWW document], http://www.ewita.com/, viewed on March 25, 2005.

4. Malhotra, Yogesh. (1996). "Enterprise Architecture: An Overview", [WWW document], BRINTInstitute. URL http://www.kmbook.com/enterarch.htm, viewed on March 25, 2005.

5. Systems and Software Consortium, "*Enterprise Architecture Best Practices*", **Architecture Line**, www.software.org, 2003.