

Architecture Framework for Resolution of System Complexity in an Enterprise

Mgbeafulike Ike, J.* and Okonkwo Obikwelu, R.**

*Dept of Computer Science, Anambra State University, Uli, Anambra State, Nigeria

**Dept. of Computer Science, Nnamdi Azikiwe University, Awka, Anambra State, Nigeria

Abstract: Software engineers and architects have been faced with the problem of IT system complexity for some years. The challenge to build highly complex IT systems, ensure that these systems meet the needs of increasingly complex business processes and do all this in a way that allows everything to adapt quickly to changing market condition coupled with rapid development in information technology has been a problem facing the development and building of workable and functional information systems. Research on this problem has been conducted and it was found to cause large cost of software development project, schedule overrun and outright software project failure. A novel architecture framework was developed as a tool for resolving and managing IT system complexity in the enterprise as the end objective. Comprehensive business system frameworks are necessary to capture the entire complexity of such systems. The framework provides the conceptual foundation necessary for building and managing the integral business system and all its components and also provides an integrated description of enterprise information systems that comprises of the back-end systems, front-end systems, management tools and communication system. The framework provides a detailed process of information system development and defines the necessary subsystems that make up the integrated enterprise information system.

Keywords and phrases: Complexity; Information System; Enterprise Architecture Frameworks Enterprise Architecture, Information Technology.

I. Introduction

Almost everything in the Enterprises today is becoming more and more complex. On the business side, regulating requirements, partnerships, relationships, mergers and acquisition all serve to make business process more complex. On the technology side, new distribution models, heightened demand for interoperability and workflow automation all serve to make IT systems more complex. In the midst of all these serve to make IT systems more complex. In the midst of all these complexity comes a requirement for agility – rapid changes in business models, customer expectations, market conditions all put inexorable demands on enterprise to keep its system agile. Most existing enterprise architecture and methodologies evolved in a simpler era. Thus while they are focusing on addressing traditional problem of Information Technology/ business alignment, they do not address the much more difficult problem of managing today's complexity. Complexity is one of the major factors in the cost of developing and maintaining software. The resolution of system complexity has been of great interest to researchers in software engineering for some time. One of the key reasons for this interest is the potential to use the measurements in procedures to control costs of a system over its lifetime. According to Coad and Yourdon (1991). The resolution of system complexity in an enterprise or even a project is a challenging endeavor. The complex interrelationships between technical, business oriented and organizational factors combined with the pace of change in today's organisations make the analysis of the enterprise information systems in general and with regards to its complexity a non trivial task. The need for a more business oriented view to justify investments in information system and to report and analyze the related costs for the enterprise is recognized by a variety of researchers (Zuse, 1990, Card and Glass, 1990). To support a business oriented approach to system complexity a variety of stake-holders in the enterprise have to cooperate and communicate. This poses additional problems to be solved as the various stake-holders have a differing background and view on these issues. The problems that have to be addressed in system complexity resolution can be summarized as (i) handling the complexity of business supporting information processes and understanding their dependencies and interrelationships (ii) The resolution of system complexity using enterprise architecture

In this paper we propose a novel approach to the resolution of system complexity in organisations and projects. Our approach tries to utilize the advantages offered by the discipline of enterprise architecture to support an enterprise-wide holistic system complexity resolution. The motivation for the approach is to show how enterprise architecture can be used to overcome the previously discussed problems. Enterprise architecture is an approach that provides an integrated description of an organisation's structure, processes and its underlying information technology landscape and reduces complexity by providing specific viewpoints on an integrated entire model. By combining the enterprise architecture with a system driven complexity resolution process we

are able to address the previously discussed problems. The resolution process itself contains classic actions which includes (i) Identify the subsystems which have to be together for the functional 'cohesion'. (ii) Form a cluster of these subsystems and identify interconnections in this cluster (iii) Form clusters of the remaining subsystems (iv) Connect the clusters with an interface. All these actions are utilizing the information contained in the enterprise architecture as input and are producing as a result changes in the architecture.

System complexity is an area of software engineering concerned with the measurement of factors that affect the cost of developing and maintaining software (Zuse, 1990). System complexity can be said to be the degree of difficulty in predicting the properties of the system. It is also said to be the composite measure of complexity inside procedures and between them in a system. System complexity is defined by the number of asynchronous subsystems and the amount of their different communication and synchronization needs: every exchange of information between two activities that run in parallel disturbs in one way or another both running processes, and the handling of this disturbance in a predictable way is not at all obvious. More and more organizations are turning to the evolving science of enterprise architecture to develop solutions to these system complexity and challenges.

Contemporary businesses have of necessity installed IT infrastructure in order to operate, compete and grow in the current business climate. Highly abstracted, enterprises can be represented as the complex interplay of people, processes and technologies in achieving business objectives. These objectives are typically articulated as financial, operational and market success measures. While IT infrastructure has obviously accelerated the pace, reach and agility of business, it has also introduced a huge amount of complexity into enterprise fabrics. Typically, roughly 80 percent of an IT budget is spent on maintenance and development of existing enterprise systems while 10 percent of the entire workforce is devoted to IT operations; in addition, associated costs are difficult to control and maintain. Therefore, a radical shift is necessary whose principal goal is "to kill the complexity." Whichever road we take, we must cope with huge volumes, rising dynamics, conflicting constraints and self-defeating uncertainty. These are the principal contributors to system complexity in an enterprise (kamal, 2002). Historical developments created at least three sediments of computing systems: mainframes, server-client systems and Web systems. Each had very specific creation factors and driving causes. Only recently has it become clear that integration and interoperations are keys for the flexible, evolving and reliable enterprise architecture consisting of these sediments. Competition among vendors and inability to agree or impose common standards led to heterogeneous systems that talk to each other only through adapters and operational bridges. Eventually, this stimulated the rise of enterprise middleware systems whose only function is to mediate among incompatible systems. Development style, budget pressure and project deliverables followed by the necessity to deliver short-term results in an over-constrained environment led to the rapid creation of huge amounts of proprietary code. Maintenance and operation of this "mountain-of-spaghetti" code base represent a logistic nightmare. The volume of enterprise data has risen sharply also, as productivity tools enabled enterprise employees to create quickly huge numbers of big documents leading consequently to lower content quality. This was followed by sharpened expectations about responsiveness and latency within and out of the enterprise. All of the factors mentioned above contribute to the huge growth in enterprise IT system complexity. Four types of complexity that affect software and systems designers was captured as (i) **Requirement Complexity** which deals with the complexity inherent in the fact that stakeholders may be numerous and distributed with varying and conflicting goals. Eliciting stakeholder's requirements is a challenge in itself, however once collected there is the task of articulating the requirements to be considered. Once collected, the requirements must be expressed into a common language understandable to both the project team and the stakeholders. (ii) **Organizational complexity** which deals with the barriers that exist both internally (among divisions, locations, disciplines, and teams) and externally (with suppliers, customers, and other organizations) that influence the products, software, and systems that are being built. (iii) **Process complexity** deals with the inherent complexity in the process for creation or evolution of products, whether software only or software-intensive systems involving hardware and software. From agile development, to hybrid iterative, to heavily regulated waterfall approaches, teams involved in design, development, delivery and maintenance activities face challenges with the process. Moreover, various types of processes need to integrate across organizations. Business processes, software and systems development processes, and operational processes all need to flow seamlessly into each other. Striking a balance between simplicity on one end and compliance and reliability on the other end can make it difficult to pin down the best process. (iv) **Design complexity** deals with the increasing demands being placed on the software, systems and products themselves, whether features and functions or connections to external systems. This complexity is experienced by large and small teams alike. It shows itself in the inability to rapidly assess the impact of change and act on it, as well as in a limited understanding of the design by those who are often most deeply affected by the result: the stakeholders that are external to the team. The architecture of software and systems, as well as the behavioral characteristics, are increasingly difficult to understand. Each new requirement for increased functionality increases the complexity. Add to this the need to interconnect software and systems to other systems, and the design complexity increases

even further. Trying to design these systems through code-centric approaches, without modeling and a means to visualize the design at all levels of abstraction, and without employing automation, leads to project delays, uncertain feature compliance, and broken interfaces and systems.

II. Materials And Methods

Enterprise architecture is generally defined as a model-based management and planning instrument for the evolution of enterprise-wide information systems. Enterprise architecture provides information needed to manage enterprise-wide information system effectively. Enterprise architecture involves four primary resources: Business, Application, Information and Technology. These primary resources are the sources of challenges and serious treats to the cost and complexity of information systems. The challenges facing all these resources are rapid development that affects technology. The Business Architecture aligns an organization's operating model, strategies, and objectives with IT; it also creates a business case for IT transformations and provides a business-centric view of the enterprise from a functional perspective. The Application Architecture provides an application- and services-centric view of an organization that ties business functions and services to application processes and services to application components in alignment with the application strategy. The Application Architecture's scope, strategy, standards are a consequence of the Business Architecture. The Information Architecture describes all of the moving pieces and parts for managing information across the enterprise, and the sharing of that information to the right people at the right time to realize the business objectives stated in the business architecture. The Technology Architecture describes how the infrastructure underlying the business, application, and information architectures is organized. This is shown in figure 1.

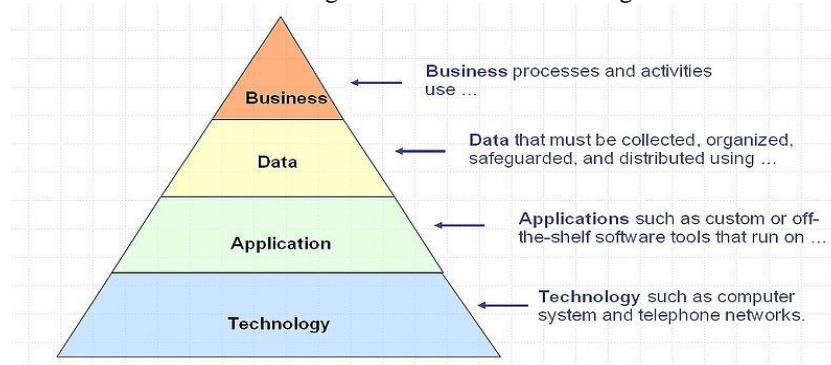


Fig. 1: Enterprise Architecture Components

The method adopted in the development of this model is the separation of concerns principle: The design principle for separating a computer program into distinct sections, such that each section addresses a separate concern. A concern is a set of information that affects the code of a computer program and Single Responsibility principle: The principle that each component or module is responsible for only a specific feature or functionality, or aggregation of cohesive functionality.

III. Results

The conceptual enterprise applications architecture has been designed and developed for managing system complexity in the enterprise. The enterprise-wide information system is broken down into sub-systems of similar type. This is based on the concept of breaking down a complex problem into two or more sub-problems of a similar type until they become simple enough to be solved directly. The enterprise applications are classified as Back end systems, Front end systems, Management tools, Engineering systems and Communication systems.

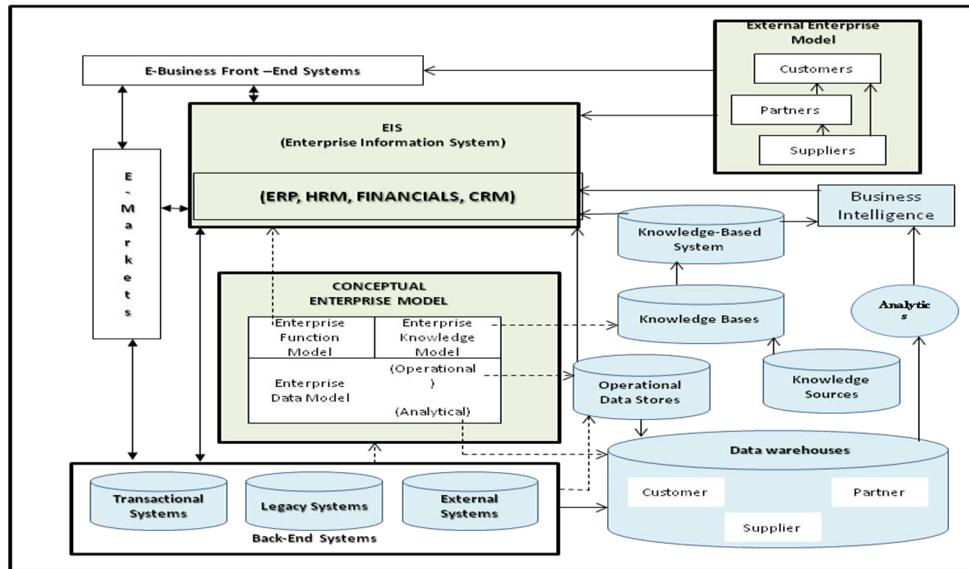


Fig. 2: Conceptual Enterprise Applications Architecture

IV. Discussions

The conceptual enterprise application architecture has been designed as a tool to manage system complexity in the enterprise. In resolving IT system complexity using enterprise architecture, the enterprise is seen as a whole and is based on the principle of separation of concerns. This architecture is divided into five levels: the external enterprise model, the conceptual enterprise model, the front-end systems, the back-end systems and the management tools.

The External Enterprise Model for EIS

The external enterprise model in the architecture for EIS (Figure 2) consists of organizations for customers, channel partners and suppliers, the processes for operational EIS and e-marketplaces. A customer, channel partner or supplier may interact with a firm or a 3rd party e-marketplace via the e-business front-end. Operational EIS processes are the implementations of business functions described in the conceptual enterprise model. Transactions in operations are supported by the back-end transactional systems. Business intelligence from analytical processing can be used to enhance operations, whereas operational results captured by the transactional systems provide the data feeds to data warehouses for further analysis to create new business intelligence.

The Conceptual Enterprise Model for EIS

The conceptual enterprise model consists of logical representations of enterprise data, knowledge and functional requirements for operations and analytics. It provides the enterprise-wide architectural blue-print for EIS. The enterprise data model consists of operational and analytic data models representing enterprise operational and analytical data requirements. The enterprise data model is the information blue-print that provides the linkage for various data structures across the enterprise. It serves as the information roadmap for system integration, new system deployment and system maintenance. The enterprise function model is the representation of business functions supporting operational and analytic functional requirements for the enterprise. It is the functional blue-print and provides the linkage within and across the external enterprise of organizational processes and the internal enterprise of computer systems. It is not uncommon to find different processes implementing the same business function to yield incompatible results. For example, a customer may receive different price quotes from different channels such as the Web, call centers or at retail outlets. Using the enterprise model framework, these processes are linked to the business function of providing price quote, which is mapped to the enterprise data model that contains the single source of pricing information (Chan 2008).

Front-End Systems

Front end systems are used in customer interface. Front end applications are classified applications, which are used in customer interface or are utilized to create or distribute value-added services for customers.

Back-End Systems

Back end systems are transactional applications that perform transactions and operations, which are not directly in the customer interface.

Management Tools

Management tools are applications used to support management activities in all levels of the organisation. Management tools include applications such as *operational data stores, analytics*, data warehouse, business intelligence applications, budgeting and reporting, *Knowledge Bases and Knowledge-Based Systems*. These applications are often integrated with backend, front end and engineering applications and they collect data from these applications and refine it as useful information. Communication systems are used for collaboration and communication within the company and with external parties. Communication applications include applications such as e-mail and groupware.

V. Conclusion

Advance in development and rapid technological innovations with software crisis has been identified as being major contributing factors to IT system complexity. The enterprise architecture in this case becomes very useful in the aspect of managing system complexity by structuring data in a common language and format. Complex information system development intentions often result in a large number of outright failures which are sometimes economical such as budget and schedule overruns. Also due to poor product quality and insufficient user satisfaction, the information system faces great challenges. It is only by adopting enterprise architecture can enterprises be able to manage this complex system situation. Enterprise architecture is a blueprint of an organization's vision and a management framework that includes various business methods and tools to understand and document the structure of an enterprise, and help the enterprise to organize its business processes and Information Systems (IS) / Information Technology (IT) resources to achieve strategic alignment between its business processes and IS/IT capabilities.

References

- [1]. Oracle (2009). An Oracle White Paper in Enterprise Architecture October.
- [2]. Booch G (2005). "The Complexity of Programming Models". Keynote talk at AOSD 2005, Chicago, IL, March 14-18, 2005.
- [3]. Card D. N. and R. L. Glass (1990.). *Measuring Software Design Quality*, Englewood Cliffs, NJ:Prentice Hall.
- [4]. Coad P. and E. Yourdon (1991). *Object-Oriented Design*, Englewood Cliffs, NJ:Prentice Hall.
- [5]. Chan, J.O (2004). The Analytic Enterprise Model for the Value Chain. *Communications of the International Information Management Association*, 4(4), 59-69.
- [6]. Chan, J.O (2008). An Integrated Architecture for Enterprise Relations Management. *Communication of International Information Management Association*, 8(2), 55-66.
- [7]. DoD Architecture Framework Working Group. DoD Architecture Framework, Feb 2004.
- [8]. Kemal A. Delic (2002) .Enterprise It Complexity Zuse H. (1990). *Software Complexity Measures and Models*, New York, NY: de Gruyter & Co.