Understanding the Role of Enterprise Architecture
towards Better Institutionalization

Lawrence Chung       Hyun-Kyung Song       Yeong-Tae Song       Nary Subramanian
University of Texas at Dallas       Towson University       University of Texas at Tyler

Abstract.
In order for a software system to be more beneficial to an organization, its services should be laid out in a careful consideration of the various constituents of the organization, or the architecture of the organization – hereafter enterprise architecture. A software system and the enterprise architecture then ultimately help achieve the goals that the organization has, while helping to solve some critical problems the organization might be faced with. In this paper, we present our initial understanding of the role of an enterprise architecture in relation to such goals and problems, as well as to the requirements on, and the specification of, a software system. Based on such understanding, we then present our preliminary work on how to proceed towards institutionalizing an enterprise architecture for an organization, with an emphasis on the high-level concerns that an enterprise architect should have during the process of institutionalization.

1. Introduction
As software systems provide more services to various parts of an organization, so is there the need for a seamless integration of, and inter-operation among, such services. It is expected that a seamlessly integrated system of both homogeneous and heterogeneous services, as an integral part of the organization, will help achieve the goals of the organization, while at the same time alleviating or solving some critical problems the organization faces.

How well such a seamlessly integrated system can help, or even hurt, will inevitably depend on how well the system works with its environment, namely the various constituents of the organization, or the architecture of the organization – hereafter enterprise architecture. As a blue print of an organization, an enterprise architecture is comprised of important assets of the organization – people, hardware, software, resources, organizational structures, etc. that are vital to the effective functioning of the organization. An enterprise architecture model then is an abstraction of such assets, namely, highlighting only essential assets while describing only those externally visible, or attributable, characteristics of them (as black-boxes so to speak). In a layman’s sense, such a model should help prevent such episode as a blind man and an elephant where everyone insists on his/her own view of what the object is.

In this paper, we present some of our initial understanding of the nature of an enterprise architecture and its role in relation to the goals an organization tries to achieve, the problems the organization tries to avoid, and the requirements on, as well as the specification of, a seamlessly integrated software system whose parts communicate, coordinate and collaborate with each other. Based on such understanding, we then also present our preliminary work on how to
proceed towards institutionalizing an enterprise architecture for an organization, with an emphasis on the high-level concerns that an enterprise architect should have during the process of institutionalization.

For our work, we adopt, integrate, and extend ideas largely from two areas, which have been developed somewhat independently of each other: Requirements Engineering and Enterprise Architecture. From the area of Requirements Engineering, we adopt ideas on the relationship among a software system specification, requirements and the environment of a software system [GGJZ2000], ideas on treatment of non-functional requirements as softgoals and rationalizing design decisions using them [CNYM2000], ideas on formal treatment of goals and satisfying them [van Lamsweerde2000], and ideas on agent-orientation for modeling organizations [YM1994] [MC2000]. From the area of Enterprise Architecture, we adopt ideas from a number of so-called frameworks, including the Zachman framework, Capgemini’s Integrated Architecture Framework, Microsoft Enterprise Architecture, TOGAF - the open group architecture framework, and FEAF - the federal enterprise architecture framework ([AKL1999] [HV1993] [Luftman2000] [SK2007] [WF2006] [Zachman2003]).

In Section 2, we present our initial thoughts on the role of an enterprise architecture. In Section 3, we present the process of enterprise architecture institutionalization. At the end, we summarise our contributions and describe future plans. In Section 3, we present the process of enterprise architecture institutionalization. At the end, we summarise our contributions and describe future plans.

2. Understanding the Role of Enterprise Architecture


In the area of Requirements Engineering, there have been some major developments. One such development is the reference model [GGJZ2000], which states that the requirements (R) is satisfied through the collaboration between the behavior of a software system (S) and its environment/domain phenomenon (D):

\[ S, D \models R. \]

Another major development has to do with the notion of softgoals [CNYM2000], which non-functional requirements have been treated as and used in rationalizing just about any kind of decisions in selecting among alternatives – namely, a good, better or best option among functional operationalizations. Goal-orientation was pioneered initially in dealing with functional goals and making refinements, through AND/OR decompositions, as earlier in [Nilsson1971] and later in [van Lamsweerde2000]. A formal characterization of the relationship between goals (Gh) and the concepts in the reference model, with the additional goal of the accuracy of the information maintained by the software system (Ac) and domain expectations or assumptions (As), is also given in [van Lamsweerde2000] as:

\[ S, Ac, D \models R \text{ with } S, Ac, D \not\models \text{false, and} \]

\[ R, As, D \models Gh \text{ with } R, As, D \not\models \text{false.} \]

Similarly to the description in [SC2000], these notions can be further generalized to

\[ S, AnySoftwareSoftgoal, D \models R \text{ with } S, AnySoftwareSoftgoal, D \not\models \text{false, and} \]
\[ R, As, D \models Gh \text{ with } R, As, D \neq \text{false}, \text{ plus} \]

\[ Gh \text{ make } Gs \text{ or } Gh \text{ help } Gs \text{ or } Gh \text{ hurt } Gs \text{ or } Gh \text{ break } Gs \]

The last part is somewhat of a rough formalization about the notion of satisficing for softgoals (Gs) as in [CNYM2000] – functional operationalizations can make, help, hurt or break softgoals (or non-functional goals and requirements). For more details on this, see [CNYM2000]. An earlier work on dealing with a goal-oriented enterprise architecture can be found in [SCS2006].

Yet another development in the area of Requirements Engineering is agent–orientation (see [YM1994] [MC2000] for more details), which provides for dealing with multi-agents, in terms goals, softgoals, resources, tasks and various types of dependencies among them.

**Understanding the Role of Enterprise Architecture in Context.**

In relation to these latest advancements in the area of Requirements Engineering, an enterprise architecture is largely about the environment of a software system, although it becomes a bit blurry when it comes to the hardware device or infrastructure a software system has to run on whether an enterprise architecture is exclusively about only the environment of a software system. Not barring any sharper characterization, it suffices to say that an enterprise architecture is about important assets of an organization and relationships between them. Important assets would include people in the environment who may or may not directly interact with the software system, various hardware devices (e.g., printers, sensors, etc.), technology infrastructures (e.g., computer network, both wired and wireless, a distributed operating system, data repositories, etc.), organizational structures (e.g., departments, divisions, regional vs. international offices, etc.), resources (e.g., monetary budget, office spaces, meeting rooms, etc.). These assets and dependencies between them are essential parts of the environment/domain (D), where some of them collaborate with the software system (S) in order to achieve the requirements (R), which in turn require various phenomena (As) induced by agents in the environment/domain (D) to achieve a set of the organizational goals – both hardgoals (Gh) and softgoals (Gs), while solving some key organizational problems (P).

A key question that remains then would be ultimately how well, or how poorly, the set of organizational goals, Gh and Gs, can be achieved, and how well, or how poorly, organizational problems (P) can be solved. But since

\[ S, \text{AnySoftwareGoal}, D \models R \]

\[ R, As, D \models Gh \]

\[ Gh \text{ make } Gs \text{ or } Gh \text{ help } Gs \text{ or } Gh \text{ hurt } Gs \text{ or } Gh \text{ break } Gs \]

the “how well, or how poorly” questions have to do with the “how well, or how poorly” questions about the environment/domain (D), various domain phenomena (As) – this is to be realized by some agents in D, and the requirements (R) – the achievement of these also relies on D.

Since enterprise architecture is a blue print of the organization, the way the software system (S) and its environment/domain (D) collaborate with inevitably depends on such enterprise architecture. A better enterprise architecture is likely to result in a better achieved goal, and vice versa. In other words, one enterprise architecture can better satisfice some key softgoals than another enterprise architecture.
Of particular interest here about enterprise architecture is some of the commonly recognized benefits of having a clear understanding, and description, of an enterprise architecture. In other words, some of the key softgoals that have been associated with the notion of enterprise architecture include, among other things:

- IT-related benefits:
  - Better complexity management
  - Better technical resource oversight
  - Better knowledge management
  - Better IT visibility
- Business-related benefits:
  - Reduction in impact of staff turnover
  - Faster adaptability
  - Operating procedures improvement
  - Better decision making

In the next section, we describe how to proceed with institutionalizing an enterprise architecture, which is likely to resulted in better achieved goals – both hard and soft.

### 3. A Process for Institutionalizing an Enterprise Architecture

In proposing a process for institutionalizing an enterprise architecture, we adopt, among other things, the spirit of the Zachman Framework for Enterprise Architecture [Zachman2003], which offers a normalized semantic structure, freeing its user from low-level implementation concerns, such as top-down, bottom-up, left-to-right or right-to-left methodologies.

![Business-IT Alignment Problem](image)

**Figure 1. Business-IT Alignment Problem**

For the illustration of our approach, we will be using a well-known methodology, called 5W1H, as seen below:

![5W1H Table]

Those are the questions that we will be facing in describing our process for institutionalization. The following describes the steps for the institutionalization.
1. Initiate the process

- Most of organizations are currently in support of both legacy and new information systems. To better support of those heterogeneous systems and achieve organization’s goals, structural enhancement needs to be considered, i.e., institutionalization of well-known enterprise architecture to the organization. For that, the organization must decide the scope of the project such as what to include, who to be involved and how to do it. Next step is to build a team and establish a formal target vision. When building a team, the team should include stakeholders such as customers, users, system architects, system developers, system maintainers, business managers, IT managers because each sees the architecture differently. These views categorize Enterprise Architecture as in the following:

- **Business architecture**: represents the fundamental organization of the corporation from a business strategy viewpoint.
- **Process architecture**: service development, service creation, service distribution. Focus on effectiveness and efficiency.
- **Integration architecture**: information system components in the relevant enterprise context. Focus on agility, cost efficiency, integration, and speed.
- **Software architecture**: the fundamental organization of software artifacts.
- **Technology (or infrastructure) architecture**: computing, telecommunications, hardware and networks.

The above architectures are considered as fundamental decomposition of enterprise architecture. When adopting a well known enterprise architecture to an organization, documentation and publishing standards must be in place so that the adopted architecture can be institutionalized. Those can be categorized as:

- **Core artifacts**

  - **Strategy specification** (“what” questions): Hierarchy of organizational goals and success factors, Product/service model (including partners in value networks), Targeted market segments, Core competencies, Strategic projects, Business principles, Dependencies between these artifacts
  - **Application specification** (business IT alignment questions): Specification of applications and application Components, Specification of enterprise services and service components
  - **Software specification**: Specification of software components, Specification of data resources, Specification of interfaces among software components, Dependencies between these artifacts
  - **Technical infrastructure specification**: Specification of IT components, Dependencies between these artifacts

2. Characterize the baseline architecture (describe where you are now)

   This step describes the baseline architecture. Without identifying the present state of the information system, design of new architecture may not be implemented properly. To describe the baseline architecture, this step needs to identify hidden assets, gaps and redundancies. The best approach is to organize information according to architectural view.

- **Characterizing Business architecture**: The business architecture represents the fundamental organization of the corporation (or government agency) from a business strategy viewpoint. Design and evolution principles for business architecture can be derived according to the market based approach or the resource based approach to strategic management.
- **Characterizing Process architecture**: The process architecture represents the fundamental organization of service development, service creation, and service distribution in the relevant enterprise context. Design and evolution principles for this layer focus on effectiveness (creating specified outputs) and efficiency (meeting specified performance goals).
- **Characterizing Integration architecture**: The integration architecture represents the fundamental organization of information system components in the relevant enterprise context. Design and evolution principles for this layer focus on agility (e.g. by service orientation), cost efficiency (e.g. by reduction of interfaces), integration (e.g. by analysis of data coupling), and / or speed (e.g. by straight through processing).
• **Characterizing Software architecture**: The software architecture represents the fundamental organization of software artifacts, e.g. software services and data structures. A broad range of design and evolution principles from computer science is available for this layer.

• **Characterizing Technology architecture**: The technology architecture represents the fundamental organization of computing / telecommunications hardware and networks. A broad range of design and evolution principles from computer science is available for this layer too.

• **Survey the users**: Measure of user satisfaction with the existing system for developing the target architecture.

3. **Designing target architecture**

• This step defines the target business architecture according to business strategic viewpoint. The target architecture describes organization’s future information systems that aligned with their business strategy. In other words, the target architecture reflects where the organization wants to be, not where it is now.

• One of the widely accepted models of alignment , as in [HV1993]. Their model, also known as the Strategic Alignment Model, describes the alignment between the architectures along two dimensions as in the Figure 2. The dimension of strategic fit differentiates between external foci and directs towards administrative structures. The other dimension of functional integration separates business and IT. The model defines four domains that have been harmonized in order to achieve the alignment.

![Strategic Alignment model](image)

Figure 2. Strategic Alignment model

• In this step, the analysis of architectural differences between the baseline and target business architecture related to business strategy must be performed. A gap analysis identifies the differences between the baseline and target architectures. The gap analysis results should be applied to design the other target architectures namely, Process, Integration, Software, and Technology architecture. The target business architecture should include the strategy and goals of the organization.

• The characteristics of target architecture must be defined and implemented. The following may be considered as representative target architecture characteristics:

  - **Modifiability**: the easiness of change of business objectives and operations, the evolution of infrastructure components.
  - **Traceability**: the level of visibility of the relationships among components that constitute the architecture across the views.
  - **Conformity**: to the principles and standards defined in the architectural framework.

4. **Plan architecture transition**
- This step in EA process assesses organizational technology maturity. Assessment of the maturity of the technology available should be done to implement target architecture.
- The design constraints should be identified and applied to the implementation. The constraints need to make sure the quality of the implementation. Some of the constraints are:
  - **Performance**: How often or how fast does the system need to run?
  - **Interoperability**: Under what conditions does the system need to work?
  - **Reliability**: The ability of a system or component to perform its required functions under stated conditions for a specified period of time.
- Analysis of the dependencies and design constraints should be performed.
  - The architecture team should understand the dependencies of applications and perspectives and review design constraints across all projects.
- The architecture transition plan should be done based on the above processes’ outcome.

5. **Assess the enterprise architecture** (criteria for evaluation) Institutionalizing Enterprise Architecture is an incremental and continuous process. In this paper, we are limiting the scope of our discussion to the quality of EA itself. The differences from the gap analysis and organizational culture may be incorporated in the future work. The following analyses are used to measure the quality of EA within an organization.

- **Dependency analysis** exploits the associations between the various EA artifacts to derive direct and indirect dependencies these artifacts. Which business processes are affected if we switch-off a certain server? Which applications are required to facilitate the value chain of a product offered at the marketplace?
- **Coverage analysis** represented as matrices relating the two dimensions of interest. Redundancies and gaps can be identified.
- **Interface analysis** focuses on the interfaces within a class of EA artifacts. Minimizing the coupling, maximizing the cohesion.
- **Complexity analysis** calculates a complexity measure based on the number of architectural components and the dependencies between those components. The design goal is to reduce the overall EA complexity.
- **Cost analysis** is the calculation of IT-related costs, the cost for the allocation of products/process/organizational units.

4. **Conclusion**
The notion of enterprise architecture has been gaining significant interests from both academia and perhaps more from industry, thanks to the benefits associated with it. In this paper, we have taken a fresh look at the role of enterprise architecture from a Requirements Engineering perspective, in order to better understand the value of an enterprise architecture in the context of organizational goals, problems, and their solutions. We then have given a preliminary version of our ongoing work towards institutionalizing an enterprise architecture that can help create the expected value as much as possible. We understand that still much remains to be done before every organization can institutionalize highly beneficial enterprise architecture. We plan to investigate ways to better understand a seamless integration of an enterprise architecture with various types of software services embedded in it. As another line of future work, we would like to apply our proposal to several organizations and collect the results that can be used as metrics that represent the degree of institutionalization.
References


