Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 53 (2016) 158 – 165

www.elsevier.com/locate/procedia

The 10th International Conference on Axiomatic Design, ICAD 2016

A systematic approach to enterprise architecture using axiomatic design

Farhangmehr Behrouz[a,*], Mehdi Fathollah[b]

[a] Department of industrial engineering, Islamic Azad university, Karaj branch, Alborz 31485- 313, Iran
[b] PhD, Assistant Professor, Department of industrial engineering, Islamic Azad university, Karaj branch, Alborz 31485- 313, Iran

* Corresponding author. Tel.: +98-912-363-7782; fax: +98-263- 441 -8156. E-mail address: f.mapna@yahoo.com

Abstract

Today, the organizations have complex structures and therefore, Enterprise Architecture (EA) could provide them with solutions to describe, coordinate, and align their business elements in order to achieve the strategic goals and deploy organizational governance. In this regard, various frameworks offered according to enterprise activities field. Multi-layered and pyramidal structure is the common feature of most frameworks, from strategic planning on top of the pyramid to information technology infrastructure at the bottom. So far, several models and methods are developed to specify the architecture requirements of each layer and trace architectural components at different layers (often with different substances), mainly just by descriptive and graphic tools. Translating and converting strategic requirements to processes, data, and technology, providing the organization big picture in detail and handling change management are the main purpose of EA. These cannot be achieved unless the requirements are accurately and systematically determined from the top to the bottom of the pyramid. Also, the architecture of each layer is designed to respond the requirements of the upper layer, while specifying the exterior and outward relationship between heterogeneous architectural components, not only does not cover all the needs, but also could be misleading for the organization. This paper attempts to deploy a methodology based on Axiomatic Design (AD) by using two axioms to systematically analyze the current enterprises capability and map the requirements of each layer of EA as the design domain into other domains.

Peer-review under responsibility of the scientific committee of The 10th International Conference on Axiomatic Design

Keywords: Axiomatic Design; Enterprise Architecture; Information Technology Analysis

1. Introduction

Most of the enterprises are encountering business changing, for instance a development of products and services or economic situations. According to these situations, they have to absolutely improve their business processes in order to be able to survive. In this regards, these enterprises, should adapt themselves to these changes effectively [1], [2]. In view of the increased business and organizational extension and dynamics, integration, agility and the ability to change, are becoming more and more important. Enterprises should thus pay considerable attention to their enterprise architecture [5]. Enterprise Architecture (EA) is the process of translating business vision and strategy into effective enterprise change by creating, communicating, and improving the key principles and models that describe the enterprise’s future state and enable its evolution. [Gartner, 2008]

Large and medium-sized organizations regard the alignment of business and IT as the most important motive for working on an EA. Other important reasons for putting EA on the agenda are support for change processes and strengthening the flexibility of the company. [Roeleven, 2010]. Since EA artifacts are not sufficient for make alignment between business and IT within enterprises, enterprises are looking to find a method to address theirs challenges on competiveness by implementing EA artifacts [7]. So far Several EA Implementation Methodologies or EAIMs have been proposed by academics and practitioners in literature [8].

Peer-review under responsibility of the scientific committee of The 10th International Conference on Axiomatic Design
doi:10.1016/j.procir.2016.07.012

Although they are different in implementation practices and development phases, they are common in the concepts, principles of transition from current architecture (As-Is) to desire architecture (To-Be) [9].

In spite of the huge interest in EA it turns out that 66 percent of programs did not fulfill expectations [Roeleven, 2010]. This study intends to analyze the applications of AD theory to the Enterprise Architecture (EA), in order to provide a methodology for dealing with existing challenges. Axiomatic Design (AD) is distinguished from other systematic design methods by having design axioms that guide good design decisions [Suh, 1990]. In "literature review" section key concept of AD, EA and complexity are described. Also, in "research gap and need for action" section, current challenges in EA, and corresponding solutions by AD are presented. "Proposed EAIM" section describes proposed algorithm for EAIM based on AD and last section summarizes the work and suggests future research works.

2. Literature review

2.1. Key Concepts of Axiomatic Design

Axiomatic Design (AD) principles have been expanded and applied to numerous engineering and non-engineering applications and proved to provide structured implementation procedures [Kulak et al., 2010]. AD Theory was proposed by Nam Pyo Suh. The goal of AD is to establish a scientific basis for design and to improve design activities by providing a theoretical foundation based on logical and rational thought processes and tools [Suh, 2001 p.5].

The AD framework divides the design process into 4 domains [Suh, 2001 p.11]: the customer domain, the functional domain, the physical domain and the process domain. In each domain, there is a characteristic vector. Respectively, they are customer attributes (CAs), functional requirements (FRs), design parameters (DPs) and process variables (PVs). As shown in Figure 1, the domain on the left relative to the domain on the right represents "what we want to achieve", whereas the domain on the right represents the design solution of "how we choose to satisfy the needs (i.e., the what)" [Suh, 2001 p10].

The process of matching variables in one domain (e.g., FRs) with other variables in another domain (e.g., DPs) is called mapping: to go from WHAT to HOW [Cochran et al., 2000]. Therefore, when mapping the right domain to the left domain, "zigzagging" decomposition is used. Designers are requested to create a design hierarchy. FRs and DPs, PVs must be decomposed into a hierarchy respectively until a complete detailed design or until the design is completed [Suh, 2001 p21].It is noted that DPs are defined according to FRs in the same level and FRs of the lower level are defined based on the characteristics of DPs in the upper level. This decomposition process continues until the leaf (bottom) level is reached. The domains may have several levels of abstraction that jointly describe the technical system architecture [Marques etal., 2009].

During the mapping processes, the designer is guided by two fundamental axioms to produce a robust design: the Independence Axiom and the Information Axiom [Suh, 2001 p.16].

- Independence Axiom: Maintain the independence of the functional requirements (FRs).
- Information Axiom: Minimize the information content of the design.

In particular, the axioms provide criteria for distinguishing bad designs from good ones [Suh, 1990]. One important point to note is that Axim 2 is only applied when Axiom 1 has been satisfied. In most design tasks, it is necessary to decompose the problem hierarchically. The FRs, DPs, and PVs mapping process can mathematically be described as vectors [Suh, 2001 p18] in the design matrix. A design equation should be written for each transition between domains and at each decomposition level. Detailed information and elaborations on the scientific background of AD are provided by Suh [2001].

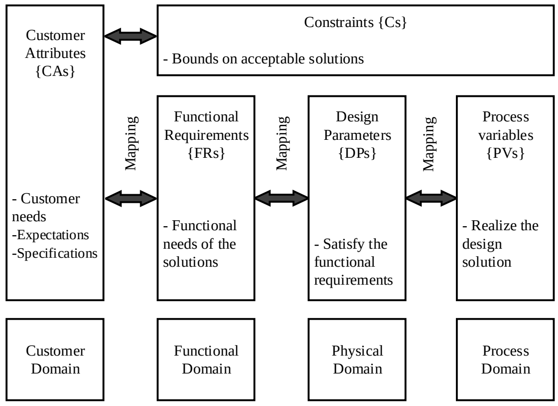

Fig. 1. Axiomatic design domains [Suh, 2001]

2.1.1. The Independence Axiom

Using vector notations for FRs and DPs, the relationship is expressed as the following design equation:

$$FR = ADP$$

Matrix A is called a design matrix. The characteristics of matrix A determine if the Independence Axiom is satisfied. If the design matrix is a diagonal matrix, it is an uncoupled design. Because each DP can satisfy a corresponding FR, the uncoupled design perfectly satisfies the Independence Axiom. When the design matrix is triangular, the design is a decoupled design. A decoupled design satisfies the Independence Axiom if the design sequence is correct. When a design matrix is neither diagonal nor triangular, the design becomes a coupled design. In a coupled design, no sequences of DPs can satisfy the FRs independently. Therefore, an uncoupled or a decoupled design satisfies the Independence Axiom and a coupled design does not. If a design is coupled, an uncoupled or decoupled design must be found through a new choice of DPs. It is noted that constraints (Cs) exist in the design. Constraints are generally defined from design specifications and they must be satisfied [14]. As an index for coupling, the following index R called "Reangularity" is defined [14]:

$$R = \prod_{\substack{i=1,n-1 \\ j=1+i,n}} \left[1 - \frac{\left(\sum_{k=1}^{n} A_{ki} A_{kj} \right)^2}{\left(\sum_{k=1}^{n} A_{ki}^2 \right) \left(\sum_{k=1}^{n} A_{kj}^2 \right)} \right]^{1/2}$$

Where the numbers of FRs and DPs are n and each element of the design equation is Aij. Reangularity is not sufficient to show all the cases of coupling. Therefore, another index called "Semangularity" (this means the same angle quality in Latin) is defined as follows:

$$S = \prod_{j=1}^{n} \frac{|A_{jj}|}{\left(\sum_{k=1}^{n} A_{kj}^2 \right)^{1/2}}$$

Table1 shows the conditions of each design for different Reangularity and Semangularity:

Table1. Reangularity and Semangularity for each design

	Uncoupled design	Decoupled design	Coupled design
Reangularity	1	$R=S<1$	$R\neq S<1$
Semangularity	1	$S=R<1$	$S\neq R<1$

2.1.2 The Information Axiom

AD requires satisfaction of the Independence Axiom. Multiple designs that satisfy the Independence Axiom can be derived. In this case, the best design should be selected. The best design is the one with minimum information. Generally, the information is related to complexity. The probability of success has been utilized as an index of the information content. Suppose p is the probability of satisfying FRi with DPi. Then the information content is defined as:

$$I_i = \log_2 1/p$$

In equation 4, the reciprocal of p is used to make the larger probability have less information. Also, the logarithm function is utilized to enhance additively. The base of the logarithm is 2 to express the information content with the bit unit. Information content can be calculated by using the probability density function.

The terminologies are as follows:

The design range is the range for the design target, the system range is the operating range of the designed product and the common range is the common area between the design range and the system range. The design range is defined by lower and upper bounds and the system range is defined by a distribution function of the system performance. The design should be directed to increase the common range. The information content is defined as follows:

$$p_s = A_{cr}/A_{sr}$$

$$I = -\log_2 \left(A_{cr}/A_{sr} \right)$$

Where A_{sr} is the system range and A_{cr} is the common range.

2.1.3 Complexity

The complexity concept is the measure of uncertainty in satisfying FRs, caused by poor design or lack of knowledge (or understanding) about the system under consideration. [12] Complexity is a relative quantity, which is determined by the overlap between the system range (actual performance) of FRs based on the system chosen and the design range (desired performance) of FRs. There are four deferent types of complexity: time-independent real complexity, time-independent imaginary complexity, time-dependent combinatorial complexity and time-dependent periodic complexity.

The relationships between the FRs determine how difficult it will be to satisfy the FRs within the desired certainty and thus complexity. So, coupling increases time-independent real complexity and in general, real complexity of a decoupled system is larger than an uncoupled design.

Imaginary complexity is defined as: Uncertainty that is not real uncertainty, but arises because of the "designer's" lack of knowledge and understanding of a specific "design" itself. Time-dependent complexity consists of combinatorial complexity and periodic complexity, depending on whether the uncertainty increases indefinitely or occasionally stops increasing at certain point and returns to the initial level of uncertainty.

The time-dependency can come from either 1) time-varying system range or 2) unpredictability of FRs in future. The first type is subject to probabilistic treatment, and thus is related to time-varying system range. The second type is explained in terms of the unpredictability of FRs in future. [10]

To create a system with high reliability, the complexity of the system must be reduced. This can be done by eliminating time-independent real and imaginary complexity and by transforming time-dependent combinatorial complexity to time-dependent periodic complexity through the introduction of a functional periodicity. In addition, real complexity may be reduced by eliminating coupling of FRs in an engineering system.

2.2. Key Concepts of Enterprise architecture

Enterprise architecture (EA) is "a well-defined practice for conducting enterprise analysis, design, planning, and implementation, using a holistic approach at all times, for the successful development and execution of strategy. Enterprise architecture applies architecture principles and practices to guide organizations through the business, information, process, and technology changes necessary to execute their strategies."[15] For the first time EA was introduced by John Zachman in 1987. The purpose of the founder of EA was to use architecture like civil inside of enterprises to reduce complexity of developing Information Systems (IS). At first he presented the framework to create skeleton for his purpose.

In EA project, enterprise architect must select a framework and an implementation methodology. EA implementing method or EAIM can be independent or dependent to a framework. While EA framework tries to capture information from enterprise's business and IT, and model them, EAIM tries to utilize models for developing appropriate ISs and IT Infrastructure for enterprise [18], [19].

A Zackman’s Framework (ZF) (as first EA framework) is limited to architecture and does not include a strategic planning methodology [Zachman, 1987]. In 1992 Steve Spewak introduced the first methodology for implementing EA. Spewak presented the EA planning to complete EA lifecycle. In other words, EA methodology complement EA framework. EA contains three principal phases, As-Is architecture, To-Be architecture, and migration plan [20].

In As-Is architecture (also known as baseline, current, and initial architecture), EA will be defined current situation of business and IT of enterprise by means of set of definitions which illustrate the current state of the enterprise's mission, business processes and technology's infrastructure. [9], [19].

In To-Be architecture (also known as desired, future, target architecture) EA will be represented the desired architecture including future of business and IT based on vision of enterprise. This type of architecture is the result of enterprise's long-term strategies and plans [19].

EA frameworks structure architects' thinking by dividing the architecture description into domains, layers or views, and offers models. Since Stephen Spewak's Enterprise Architecture Planning (EAP) in 1993 [22], and perhaps before then, it has been normal to divide enterprises architecture into four architecture domains:

- Business architecture,
- Data architecture,
- Applications architecture,
- Technology architecture.

Note that, the applications and technology architecture is about the choice of and relationships between them in the enterprise's portfolio, not about the internal architecture of a single application or technology (which is often called application or technology architecture). Many EA frameworks combine data and application domains into a single (digitized) information system layer, sitting below the business (usually a human activity system) and above the technology (the platform IT infrastructure).

Enterprise Architecture (EA) optimizes an enterprise’s Information Technology investments and translates business strategies into implementable technology solutions to achieve business competitive environment [24]. In fact, EA have a specific role to evolve Information Systems (ISs), developing new systems and incorporate new technologies to reach enterprise mission optimize [2].

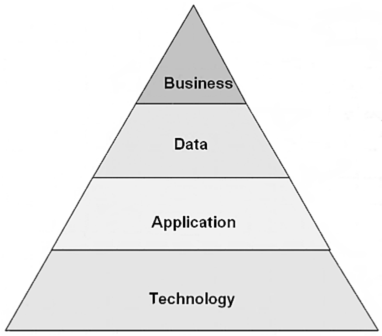

Fig. 2. Layers of the enterprise architecture.[23]

2.3. Research gap and need for action

Several researches indicate that current methods have significant weaknesses in fulfilling EA goals;

- Leist and Zellner [25] stated the current EAIMs for describing or developing an EA have a broad scope and a lack of structure, so usually it causes more complexity and difficulty in carrying it out successfully.
- There are no rigorous methodologies defined for EA projects [26]. Besides, they do not provide a step-by-step method that goes from identifying enterprise business strategies up to process analysis and improvements [27].
- Enterprise Architecture (EA) is employed by enterprises for providing an integrated environment in order to support the alignment of the enterprise’s business and Information Technology (IT) [29], [30]. According to the results obtained from an Evaluation of Enterprise Architecture Implementation Methodologies [22], it must be underlined that current EAIMs are neither complete nor effectiveness in order to support and covers all demands of EA implementation, because most of them do not consider all needed aspects by together. The research [13] indicates alignment is not utilized in most EAIMs. Common EAIMs do not have specific plan for depiction complexity and dynamic aspects of EA and Although EAIMs provide appropriate methods for modeling, they are difficult in learning and using.
- IT and business are even more complex in theirs processes, therefore bringing these two sides together in EA is a critical problem. Current EAIMs do not address complexity in any meaningful manner by providing models for complexity against which architectures can be validated [22], [28].
- Since the elements and their relationships change overtime, EA implementation becomes increasingly complex [27]. Current EAIMs do not deal with complexity of dynamic aspect sufficiently and this leads to inappropriate

understanding of the future changes and difficult deployment of ISs [27].

- Since the involvement of heterogeneous stakeholder groups such as application owners, business developers, software developer, system analyzer, enterprise architect, and the others may create complexity requirements in an enterprise, an appropriate documentation and communication of the enterprise models are vital [21]. Current EAIMs were developed in the context of enterprise integration rather than interoperability [17]. As a consequence, it is difficult to make appropriate communication and decision. It is needed to EAIM consider interoperability in a step- by- step manner [4].
- Today's modeling practices are proprietary, time-consuming, and generally ineffective as tools for communicating strategic-level planning across and down all levels of enterprise. As a consequence, it is difficult to learn and use [30], [22].
- Cuenca et al. [11] highlighted, there are several EA frameworks, and they represent a modeling framework for organizing enterprise model that may have to be created during the EA implementation and all of them contain views; however, they do not define and consider lifecycle, building blocks, and how the building blocks fit together.
- There is ineffectiveness of EA implementation methodology that is used to support Enterprise Architecture Implementation due to the complexities; these complexities come from EAIM's processes, models, methods, and strategy [5], [16]. Consequently, EA projects may be faced with lack of support in the following part of EA: requirement analysis, governance and evaluation, a guideline for implementation, and continuous improvement of EA implementation [9].

Generally, mentioned problems can be grouped into following categories:

1. Complexity:

As described below, there are all four types of the complexity;

- Conditions of design matrices determine time-independent real complexity. So, by determining "decoupled" and "coupled" designs in current EA layer in "As-Is" phase, EA time-independent real complexity can be recognized. Then, by Re-architecting and making "uncoupled" design in "To-Be" phase, they would be eliminated and also the probability of FRs satisfying increases and consequently this type of complexity decreases. On the other hand, according to complexity theory, real complexity is defined as a measure of uncertainty when the probability of achieving the FRs is less than 1 because the common range is not identical to the system range [12]. This definition can be restated as `the complexity caused by system range's being outside of the design range. So, calculating the common range for FRs in each EA layer as described in "proposed methodology" can be another way to determine complexity in current and new EA. Therefore, the complexity theory can be utilized to determine complexity degree of current EA and demonstrate the complexity decrease in new EA mathematically.
- According to complexity theory, inability to depict complexity and ineffectiveness in learning and using the models is related to time-independent imaginary complexity. As demonstrated in complexity theory [10], this type of complexity is also associated with conditions of design matrices, where coupling design leads to imaginary complexity and decoupling design increases the probability of this type of complexity. So, using conditions of design matrices can manage imaginary complexity in current and new EA. Besides, effective communication eliminates ambiguities, and this leads to explicit requirements for the processes as well as positively influencing the acceptance and adoption of the EA [21]. Also, modeling different perspectives of enterprises is a significant part of modeling that need to utilize in EAIM. Consequently, by using an appropriate modeling, the EAIM could effectively reduce the complexities of current and desired architecture, and also the transition plan [6]. Using "zigzagging" decomposition method can be helpful to model EA and make an effective communication between EA components designers in each layer.
- Mentioned insufficiency in dealing with complexity of dynamic aspects and changing elements over time, implicates the time-dependence complexity. Enterprises external situations are quite variable. Such as economic conditions, technology, low, rules and regulations environment, and so on. So enterprises strategic goals must be redefined continually to deal with the changes. According to "proposed methodology" section, the goals represent CNs in top layer or domain. So, EAIMs encounters variable strategic goals, which represent customer needs in AD. Unpredictable FR (or CN in this case) inevitably increases uncertainty in achieving FR. Achieving FR with high certainty requires complete knowledge on FR. Thus, if FR is not predictable, it necessarily incurs uncertainty in achieving them. Therefore, by changing unpredictability to predictability, we can eliminate some uncertainty that is due to not knowing FR. We can prevent the system from going into certain states, which in turn can lead to fatal situation [10]. Scenario building techniques can be utilized to predict possible goals in future. In this case, each scenario has separate CNs and dedicates AD calculation to determine EA component requirements in each layer and ensures couple design absence in each of them. Thus, enterprises are ready to redeploy and shift to new architecture rapidly. Also, for a variety of reasons, after a certain time period, the system range in each EA layer or domain can change and deviate from its initial distribution. Once deterioration from the original distribution is discovered, assignable causes must be identified. One of the reasons why it is typically hard to find out the root cause is that they attempt to link directly between a symptom (high-level FR) and low-level parameters. If the architecture of the system is well understood and represented by {FR}-[DM]-{DP} hierarchy, it quickly narrows down the scope of the problem of locating the root cause [10]. Thus, a complete

decomposition of the domains along with associated design matrices can be very helpful when identifying the root cause of the problem.

2. EA components Alignment and fitting together is another concern about current EAIMs. Since the components in each layer are heterogeneous and have deference function, creating end to end capabilities through all EA layers are the issues that remained so far. Using AD capabilities include "zigzagging" decomposition method and two axioms, ensure more perfect fitting in the components.
3. Communication between heterogeneous stakeholder groups and ability to learn and use EA products play a great role in implementation and supporting them. As mentioned before, the current models are difficult in learning and using, while AD offers a clear method to demonstrate architecture elements by details.
4. Since various architectures can be created through EA projects, choosing a method for validating them is vital, which absence in current EAIMs is sensed. Evidently, AD offers a strong tool for design validating.
5. "Zigzagging" decomposition method can provides a step-by-step method that goes from identifying enterprise business strategies to process analysis and improvements.

3. Proposed EAIM

In this research, as demonstrated in figure 4, for mapping the EA layers, each EA quadruple layer is considered as one of the quadruple domains of AD, and the "zigzagging" decomposition method is utilized. In other words, at the beginning, business strategic goals are identified in customer domain. Then, processes and related data are set in the functional domain to meet the goals. In the same manner, applications which handle the processes are specified in the physical domain. Finally, infrastructures for supporting the applications determine in process domain. Figure 3 shows the proposed algorithm for EAIM based on the AD. Accordingly, AD is used for four Critical points of EA:

3.1. Define current situation in "As-Is" phase and gap analysis:

According to [3], gap analysis is employed in EA implementation in order to identify the differences in the baseline and target architectures from the architectural views. Gap analysis enumerates the components that an organization needs to change to help resolve each difference. Since identifying the requirement of target architecture and performing all needed activities for defining baseline architecture is done by architectural design; gap analysis can be seen as the partial process of architectural design [3].

"Zigzagging" decomposition method can be utilized to depict existing architecture structure and model components correlation in the EA framework. Afterwards, by constructing the design matrices and calculating Reangularity and Semangularity, conditions of matrices are specified. For information content calculation, measures and Key Process Indicators (KPIs) are considered at each level of each domain. Eventually, coupling design and comment range for each level of hierarchy in each domain are determined.

Accordingly, it is possible to clarify the current capability of the enterprise to meet the goals by detail. High percentage of common range indicates that existing architecture is capable of meeting the goals. Otherwise, conditions of matrices should be considered. It is obvious that inappropriate output of a system does not necessarily stem from its incapability and it could be caused by other factors such as inappropriate usage. Hence, if percentage of common range is not desired, conditions of matrices will help to recognize whether the current architecture is efficient or not. In this case, high percentage of coupling design is a signal for re-architecting at the right level of each domain. It can be a worthy opportunity to avoid inopportune or redundant re-architecting that AD might present at this step.

Table 2 represents a simple example of an automotive company, which consists of two levels of the hierarchy in each domain. In this case, coupling design and common range displayed totally in percentage. For more details, architects can drill down in lower levels. It can be a useful tool for them to diagnose existing troubles in current structure and making decision for improvement project type.

3.2. Need for action

Depending on coupling design condition, two approaches for improvement can be considered after gap analysis:

- Radical redesign: Fundamental rethinking and re-architect core EA components with the aim of achieving dramatic improvements in critical performance measures, such as cost, quality, service, and speed. This approach is considered in "To-Be" phase in figure 3.
- Functional or incremental improvement: This approach focuses on how to improve the current organization elements without any fundamental change. The expected outcomes of this approach are not as significant as the first one, but the process is also not so traumatic to see radical changes.

Actually, determining the proper way for improvement depends on enterprise strategic situations and needs performing Cost-Benefit Analysis. But as general role, incremental changes may be considered in situations lacking the support necessary for more radical changes or tolerable coupling design percentage of existing architecture.

Table 2. EA capability analysis

EA layers	Business	Data	Applications	Technology
AD domains	Customer Needs (CNs)	Functional Requirements (FRs)	Design Parameters (DPs)	Process Variables (PVs)
"zigzagging" decomposition hierarchy -level 1	Producing 10,000 cars per month	Design and balancing assembly lines	Customize Enterprise Resource Planning system	Design and setting up IT infrastructures
	Coupling Design= 100%, average common range = 73%			
		Coupling Design= 100%, average common range =88%		
			Coupling Design=100%, average common range = 95%	
"zigzagging" decomposition hierarchy -level 2	4,000 Model 1 per month; 6,000 Model 2 per month	Trim line A; Trim line B	Backflushing process; kanban process	Barcode; RFID; IT Landscape
	Coupling Design= 66%, average common range = 51%			
	`	Coupling Design=73%, average common range =55%		
			Coupling Design=43%, average common range =58%	

It is worth noting that this method presents a facility to partially re-architecting the domain or even a level of hierarchy in the domain that has unacceptable situation to reduce cost and effort of EA projects. Obviously, in this case, effect of changing should be considered in lower domains.

3.3. Re-architecture

As mentioned earlier, AD provides a proper facility for translating and mapping each EA domain into others. Architects and specialists can design each domain components accurately by "zigzagging" decomposition method in "To- Be" phase. It should be noted that customer domain decomposition process is usually not considered [Marques et al., 2009], but in this case, customer domain should be decomposed to break down strategic goals. (It is a merit of AD for EA while various scenarios can be created and evaluated by defining different strategic goals subset in lower levels. For example, evaluate different production rate scenarios).

In addition, enterprise values can be considered as design constraints, for instance, responsible balance between work and life or preventing further increase of working hours or workload of the staff

3.4. Evaluating and select the best design

AD allows the selection of the best alternative within a set of constraints, and also assures the most appropriate solution [Suh, 1990]. The axioms offer a basis for evaluating and selecting designs. These two axioms jointly maximize the probability of the design to fulfill its purpose, and thereby achieve the optimal design for a set of FRs [Brown, 2007].

Obviously, there is no data for measures and Key Processes Indicators (KPIs) in "To-Be" phase. So for calculating information content in this step, forecasting methods and simulating tools and software should be utilized to estimate performance of each business element. For instance, approximate capacity of assembly and pre-assembly lines in the second domain or performance of hardware in the last domain.

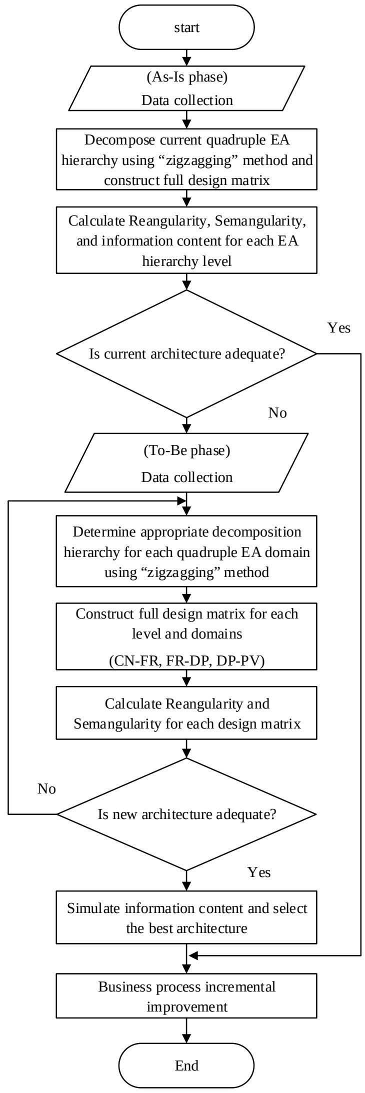

Fig. 3. The proposed algorithm for EAIM based on AD

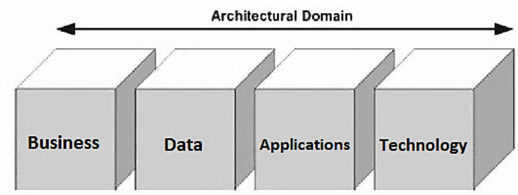

Fig. 4. Enterprise architecture domains.

4. Summary and Directions for Further Research

This study attempts to develop a methodology based on axiomatic design theory to clarify “As-Is” condition mathematically and determine the requirements of each enterprise architecture layer in “To-Be” phase. In this EAIM each EA domain is considered as an AD domain to map processes and break down the requirements to other domains in order to design EA components integrated and aligned with the organization's strategic goals. Further studies need to be conducted to develop EA using AD, including:

- EA project phase and breakdown structure modeling
- Risk analysis in EA layers
- EA Activity-Based Costing
- Moderate the complexly of the Enterprises
- EA maturity determination
- EA Value engineering
- Cost–benefit analysis of EA projects

5. References

[1] Lankhorst, Marc. nterprise architecture at work: Modeling, communication and analysis. Springer, 2013

[2] Babak Darvish-Rouhani, Fatemeh Nikpay, "Aagent-Oriented Enterprise Architecture: new approach for Enterprise Architecture", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 1, Dec 2012.

[3] Armour, Frank J., and Stephen H. Kaisler. ‘‘Enterprise architecture: Agile transition and implementation.’’ IT professional 3.6 (2001): 30–37.

[4] Niemi, E.: Quality Attributes for Enterprise Architecture Processes. Enterprise Architecture, 8 (2013)

[5] Hoogervorst , J.”Enterprise architecture: Enabling integration, agility and change” Intervational Journal Of Cooperative Information Systems 13 (3): 213-233 SEP 2004 November 19, 201

[6] Sembiring, Jaka, Edi Nuryatno, and Yudi Gondokaryono. "Analyzing the Indicators and Requirements in Main Components of Enterprise Architecture Methodology Development using Grounded Theory in Qualitative Methods."Society of Interdisciplinary Business Research (SIBR) 2011 Conference on Interdisciplinary Business Research. 2011.

[7] S. Bente, U. Bombosch, S. Langade, Collaborative Enterprise Architecture: Enriching EA with Lean, Agile, and Enterprise 2.0 Practices, Morgan Kaufmann, Waltham, MA, 2013

[8] B.D. Rouhani, M.N. Mahrin, F. Nikpay, P. Nikfard, A comparison enterprise architecture implementation methodologies, in: 2013 International Conference on Informatics and Creative Multimedia (ICICM), 2013, pp. 1–6.

[9] B. Darvish Rouhani, M. Mahrin, F. Nikpay, B. Darvish Rouhani, Current issues on enterprise architecture implementation methodology, New Perspectives in Information Systems and Technologies, VOLUME 2, vol. 276, Springer International Publishing, 2014, pp. 239–246.

[10] Taesik, L. Complexity Theory in Axiomatic Design. Massachusetts Institute of Technology 2003.

[11] Cuenca, L., Boza, A., Ortiz, A.: Enterprise Architecture Framework with Early Business/ICT Alignment for Extended Enterprises. In: Ortiz, Á., Franco, R.D., Gasquet, P.G. (eds.) BASYS 2010. IFIP AICT, vol. 322, pp. 11–18. Springer, Heidelberg (2010)

[12] Suh, N. P. (2005). Complexity: theory and applications. Oxford University Press on Demand

[13] Rouhani B.D, Mahrin M., Nikpay F., Najafabadi M.K., Nikfard P.,. "A Framework for Evaluation of Enterprise Architecture Implementation Methodologies." International Journal of Social, Behavioral, Educational Economic, Business and Industrial Engineering Vol:9, No:1, 2015

[14] Park,G.-J.”Analitic methods for design practice”, ISNB:978-1-,2007, XVI, 627 p. 286 illu., Spinger, hardcover

[15] Federation of EA Professional Organizations, Common Perspectives on Enterprise Architecture, Architecture and Governance Magazine, Issue 9-4, November 2013 (2013).

[16] A. Nakakawa, P. Bommel, H.A.E. Proper, Quality enhancement in creating enterprise architecture: relevance of academic models in practice, in: E. Proper, F. Harmsen, J.G. Dietz (Eds.), Advances in Enterprise Engineering II, vol. 28, Springer, Berlin Heidelberg, 2009, pp. 109–133.

[17] Hernández, J.E., Poler, R., Mula, J.: An Interoperable Enterprise Architecture to Support Decentralized Collaborative Planning Processes in Supply Chain Networks. In: Popplewell, K., et al. (eds.) Enterprise Interoperability IV, pp. 213–224. Springer, London (2010)

[18] Arnold, Heinrich, Michael Erner, Peter Möckel, and Christopher Schläffer. "Enterprise Architecture in Innovation Implementation." In Applied Technology and Innovation Management, pp. 132-144. Springer Berlin Heidelberg, 2010.

[19] Hjort-Madsen, Kristian. "Enterprise architecture implementation and management: A case study on interoperability." System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on. Vol. 4. IEEE, 2006.

[20] Spewak, S. H., and Michael Tiemann. "Updating the Enterprise Architecture Planning Model." Journal of Enterprise Architecture 2.2 (2006): 11-19.

[21] Hoyland, C.A.: RQ-Tech, A Strategic-Level Approach for Conceptualizing Enterprise Architectures. Procedia Computer Science 12, 37–42 (2012)

[22] Davidrajuh, R., : Developing enterprise architecture with “model-evaluate extend” approach. In: 2010 International Conference on Educational and Information Technology (ICEIT). IEEE (2010)

[23] Niles E Hewlett , The USDA Enterprise Architecture Program. PMP CEA, Enterprise Architecture Team, USDA-OCIO. January 25, 2006.

[24] Nikpay F, Selamat H, Rouhani B,D, Nikfard P, “A Review of Critical Success Factors of Enterprise Architecture Implementation”, International Conference on Informatics and Creative Multimedia, IEEE, 2013.

[25] S. Leist, G. Zellner, Evaluation of current architecture frameworks, in: Proceedings of the 2006 ACM Symposium on Applied Computing, Dijon, France, 2006, pp. 1546–1553.

[26] J. Saat, S. Aier, B. Gleichauf, Assessing the complexity of dynamics in enterprise architecture planning-lessons from chaos theory, in: Proceedings of 15th Americas Conference on Information Systems, San Francisco, 2009.

[27] S. Aier, B. Gleichauf, J. Saat, R. Winter, Complexity levels of representing dynamics in EA planning, in: Advances in Enterprise Engineering III, Springer, 2009, pp. 55–69.

[28] Brunner, T., Zimmermann, A.: Pattern-oriented Enterprise Architecture Management. In: The Fourth International Conferences on Pervasive Patterns and Applications, PATTERNS 2012 (2012)

[29] A. _Sa_sa, M. Krisper, Enterprise architecture patterns for business process support analysis, J. Syst. Softw. 84 (2011) 27.

[30] T. Clark, B.S. Barn, S. Oussena, A method for enterprise architecture alignment, in: Practice-Driven Research on Enterprise Transformation, Springer, 2012, pp. 48–76.