# Context Driven Approach for Enterprise Architecture Framework

Priyanka Chakraborty, Anirban Sarkar

# Context Driven Approach for Enterprise Architecture Framework

[1]Priyanka Chakraborty, [2]Anirban Sarkar

[1,2]Department of Computer Applications, National Institute of Technology, Durgapur, India
[1]priyanka1nitdgp@gmail.com, [2]sarkar.anirban@gmail.com

**ABSTRACT.** Decision making process within the enterprise is complex due to unavailability of widely accepted, flexible and dynamic Enterprise Architecture Framework (EAF) that comprises of service-based applications and has strong dependency on Service Oriented Architecture (SOA). Rapidly changing business scenarios have become normal characteristics of SOA based applications and Enterprise Information System. Existing non-SOA based EAFs are lacking of flexibility, scalability, context sensitivity, re-configurability and agility. On the other hand, existing SOA-based EAFs are merely capable to handle context sensitivity, reusability and agility of Enterprise Information System. To address such issues, this paper proposes a novel SOA-based EAF, called SCORE architecture, comprised of five loosely coupled layers namely, Subject layer, Context layer, Object layer, Role layer and Essence layer. Moreover, a set of relationships are proposed for SCORE architecture to exhibits the intra-layer and inter-layer associations among the constructs of different layers. Further, the inter-layer interactions and message flows in SCORE framework are analyzed using UML notations. The proposed enterprise architecture is illustrated using a suitable case study. Finally, a comparative study is performed with the Zachman Framework [1], to exhibit the benefits of the proposed EAF in the context of Enterprise Information system.

**Keywords:** Enterprise Architecture Framework, Service Oriented Architecture, Context Driven, Re-configurability, Agility

## 1 Introduction

Enterprise Architecture (EA) [2] is defined as it provides a “knowledge base and support for decision making within the enterprise and it serves as the blueprint of current situation and strategy for future directions of the enterprise”. Enterprise Architecture Framework (EAF) defines logical structure and systematic approach to create and maintain the Enterprise. An Ideal EA Framework should include-Business Value Measurement Metrics, EA Initiative Model, EA Maturity Model, Enterprise Communication Model.

Zachman Framework [3] is considered as the pioneer in the field of EAF. It introduces the concept of 5W1H (What, How, Where, Who When and Why) for five different Stakeholders (Planner, Owner, Designer, Builder, Integrator and User) in an Enterprise. Based on Zachman Framework approach, several Architecture Frameworks have been proposed such as, *TOGAF* Framework (The Open Group Architecture Framework) [4], *MODAF* (British Ministry of Defense Architecture Framework) [5]**,** *FEAF* (Federal Enterprise Architecture Framework) [6], *DODAF* (Department of Defense Architecture Framework) [7], Treasury Enterprise Architecture Framework (*TEAF*) [8], NATO Architecture Framework (*NAF*) [9], 4+1 View Model of Architecture [10], and *GERAM* (Generalized Enterprise Reference Architecture and Methodology) [11]. These non-SOA based EAF suffers from absence of (i) scalability, (ii) inflexibility to address the continuously changing business requirements of the Enterprise, (iii) re-configurability, (iv) context sensitivity, (v) well-defined model and (vi) traceability. Integration of SOA with existing Enterprise Frameworks resolves many of issues such as scalability, flexibility. Further, SOA based EAFs [12, 13, 14, 15] provides loosely coupled and reusable frameworks. But, still many serious challenges exist which require complex engineering tasks. Existing SOA based enterprise architectures are lacking of emphasizing proper subject orientation, context sensitivity and re-configurability property of Enterprise Architecture. Thus, there exist several research questions like, (i) How to make the Enterprise Architecture Framework components re-configurable and adaptive to changes? (ii) How to achieve context sensitivity in Enterprise Architecture Framework? (iii) How to increase the usability of EAF for all major stakeholders?

To address the above issues, a new enterprise architectural framework, called SCORE framework has been proposed in this paper. This framework focuses on the context sensitivity, subject orientation concept, re-usable and re-configurable capability and flexibility of EAF. SCORE framework contains five layers such as Subject layer, Context Layer, Object layer, Role layer and Essence layer. *Subject layer* defines the interested business topics and related set of goals in enterprise information system. *Context* defines all type of valid and required information that is needed to characterize the situation and surroundings of an entity [16]. Entities may be person, business elements, data set or any kind of resources related to the Enterprise. Context also describes how the entities are related to each other. Thus, context of an entity gives more clear, accurate and useful information about the current situation (like its location, situatedness, interaction with the applications, dependencies on other entities) of an entity. Context of an entity frequently changes with the changing situation and surroundings of that entity, because new attributes are required to characterize the new situation of the entity. Adoption the concept of context makes EAF more expressive, more effective and more flexible. As result, EAF can be easily adaptable of changes in enterprise like, changing business requirements, changing market situation. Integration of service concepts in *object layer*, make SCORE architecture loosely coupled, platform independent, scalable and reconfigurable framework. These mechanisms help SCORE framework to deal with the changing environment of Enterprise. Subject and context orientation in SCORE framework will give privilege to stakeholders (defined by *Role*) to realize and describe business topics, related goals, activities and business entities more clearly and conveniently.

## 2 Related work

Several non-SOA based EAFs are described in recent literatures like, Zachman Framework [2,3], *TOGAF* Framework [4], *MODAF*[5], *FEAF*[6], *DODAF*[7], *TEAF*[8], *NAF*[9], 4+1 View Model of Architecture [10] and *GERAM* [11]Enterprise Architecture Framework. Zachman Framework, MODAF, DoDAF, TOGAF, NAF and FEAF support SOA implementation to some extent while, GERAM, TEAF and 4+1 view model do not support SOA implementation.

There are many approaches regarding the integration of SOA with the Zachman framework [12]. *Approach one*: Adding Service Column as the seventh Column. *Approach two*: SOA on Nine Square: In this approach, the logical position of SOA is at the intersection of “System Model” (Designer perspective) and “Function” column. However, SOA does not only consider the applications and functions of the system, rather it affects information sharing and the network interaction with applications. Therefore, SOA affects all the neighboring eight cells of Zachman Framework. Thus SOA is integrated in the first three columns (What, How, Where) and three perspectives (Owner, Designer, Builder). *Approach three*: This approach integrates SOA in the third (Network) column because SOA concentrates on the connection among all its elements. In this column, different stakeholders view SOA from different perspectives. SOA has been integrated in MODAF [13, 14], DoDAF [13, 14], TOGAF [13, 14], NAF [15] and FEAF [13, 14] frameworks. All the SOA based frameworks are suffering from lack of subject orientation, re-configurability and context sensitivity property. Therefore, a new architecture framework is required that will support re-configurability, reusability and context sensitivity property of EAF.

## 3 SCORE: The Proposed Architecture

Majority of existing frameworks suffer from several drawbacks such as, handling scalability, context sensitivity, re-configurability, reusability and agility. These deficiencies can be overcome in the proposed architecture named as SCORE architecture, which is a context sensitive, re-configurable, reusable and agile Enterprise Architectural Framework. It comprises of five layers namely, Subject Layer, Context Layer, Object Layer, Role Layer and Essence Layer. These layers are loosely coupled, so any lower level layer can be changed according to business requirements, without making any change in upper level layers. It also increases reusability- any other applications can use the functionalities exposed by the layers. In object layer, different relationships exist among three kinds of objects (structural element objects, activity objects and event objects). Similar goal can be achieved by different interaction paths existing among the objects. Reconfigurable services have been incorporated as a part of the activity objects. Thus, object layer is capable to handle any type of changes in internal and external business environment.

### 3.1 SCORE Architecture Layers and Components

Proposed SCORE architecture uses the top down approach. Here, business topic and corresponding related set of goals have been decided first. Depending on the goal, context, object, role and essence are to be determined. If any goal is changed according to the changing needs of organization, then context, object, role and essence are also to be changed. Five layers of SCORE architecture are shown in figure 1 and all notations, used in figure 1, are listed in table 1.

***(a) Subject Layer:*** Subject Layer is the outermost layer that concerns about the business topics, related set of goals and sub goals hierarchies of the enterprise.

Management authority decides business topics depending on the area of interest. Business topics may have a set of Goals. Each goal again can be divided into several sub goals. This layer answers the questions like, what are the business topics and goals related to the specific enterprise. Why those particular business objectives are selected? What is the motivation behind it?

Formally, in Enterprise architecture, a Business topic (*BT*) can be expressed using a set of goals (*G*). Further each goal is comprised of a set of sub-goals (*SG*).

$$BT \ = \ \{BT_1, BT_2, BT_3, \dots, BT_n\}$$

$$BT_i \ \rightarrow \ (G_{i1} \ \cup \ G_{i2} \ \cup \dots \cup \ G_{ij})$$

Where, $S_i$ is an interested Business Topic, $G_{ij}$ is related goal of $S_i$

$$G_{ij} \ \rightarrow \ (SG_{ij1} \cup SG_{ij2} \ \cup \ SG_{ij3} \ \cup \dots . SG_{ijk}),$$

Where, $SG_{ijk}$ is a sub goal of $G_{ij}$

***(b) Context Layer:*** This layer focuses on context that is any kind of related information about the entities to characterize the present situation and surroundings of entities related to the Enterprise. Context can be of two types, (i) primary context and (ii) secondary or auxiliary context. All contexts those are compulsory to describe the situation of a particular entity uniquely are referred as primary context. Secondary context is required to describe the particular entity in more detail. It adds extra information about the situation of entity.

Depending on what questions of the Subject layer, what contexts are to be taken are decided in this layer. So, this layer answers what context is needed to describe the situation and surroundings of a specific entity type? What kind of dependencies and relationships exist among all the entities?

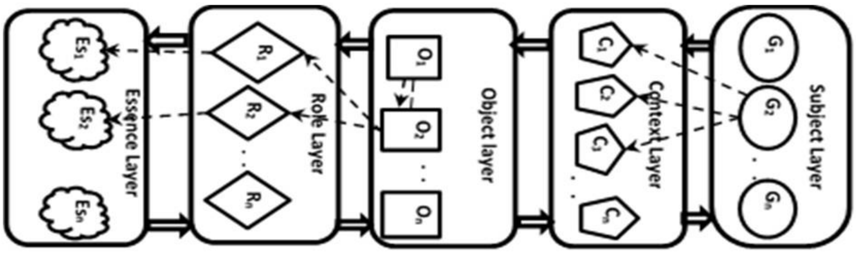


**Fig. 1.** SCORE architecture for Enterprise Architecture

Each goal (*G*) or sub-goal (*SG*) of Subject Layer is realized by certain set of contexts (*C*). So, Goal is any unordered combination of certain set of contexts. These can be expressed as,

$$G = C_1 \times C_2 \times \ldots . \times C_n, \text{ where } G \neq \Phi$$

Goal can also be expressed as the function of a set of contexts.

$$f\ (C_1, C_2, C_3, \ldots \ldots ., C_n) = G$$

Primary context (*PC*) cannot be empty set. Existence of secondary context depends on the presence of primary context. Set of primary context (*PC*) and set of secondary context (*SC*) are disjoint sets. Formal description of the above discussion is as follows

$$PC = \{PC_1, PC_2, \ldots \ldots, PC_m\}\ ,\ SC = \{SC_1, SC_2, \ldots \ldots, SC_r\}\ ,$$
$$PC \rightarrow SC\ , PC \neq \Phi\ , PC \cap SC = \Phi$$

**Table 1.** Summary of notations used in SCORE architecture

| **SCORE architecture Constructs** | **Notations** | **Interpretations** |
|---|---|---|
| Business Topic | BT | Interested business subject of the enterprise. |
| Goal | G | Set of goals to be achieved in specific business topic |
| Sub goal | SG | A goal is comprised of a set of sub goals. |
| Context | C | Related information required to describe the background information of enterprise entities. |
| Object | O | Any type of structural entity, activity or event, related to the enterprise. |
| Activity Object | A | All kinds of activities, initiated by any entity or any event. |
| Structural Object | S | Any kind of data objects, actor object and interface objects. |
| Event Object | E | All types of events those results in an activity. |
| Role | R | Role separates whole object set into different regions. |
| Essence | Es | Ensure that set of quality metrics are being achieved during accessing of specific set of services |

***(c) Object Layer:*** Object can be of three different categories like: (i) structural element objects (*SEO*) to represent the actors, any documents containing dataset and any interface, (ii) activity objects (*AO*) or functional object to represent functional unit and (iii) Event objects (*EO*) to denote events those initiates an activity. These three object types are dependent on each other. A structural element object (*SEO*) initiates an activity object (*AO*) as the result of occurrence of an event object (*EO*). This can be formally described as, $EO \rightarrow AO\ , EO \rightarrow AO$. Activity objects or Functional objects are further divided in two groups like (i) Business Process Objects and (ii) Service Objects. If process (P) exists, then only services (S) exist. So, $P \rightarrow S$. In this layer, any functional unit or any business activity can be represented as services. Service is platform independent that is it does not bother about the underlying technology, enterprise environment. Implementation of service in this layer makes object reconfigurable, so that they can be easily changeable to support the internal and external changing environment of organization.

***(d) Role Layer:*** This layer is all about the Role in the Enterprise. Role includes all the actors related to the enterprise like, owner, designer, planer, developer, customer, and database. Role makes separation of the entire object set depending on different activities. Same structural element object may play different roles depending on what kind of activities they performed. Different roles may collaborate among them. This layer contains who questions. Formal representation is as follows

$$R = \{R_1, R_2, \dots, R_n\}$$

One role can be assigned for any combination of objects. So maximum value of R can be $R_{max} = [P\ (SEO \cup AO \cup EO) - \Phi]$

***(e) Essence Layer:*** This layer takes care of quality of service (QoS) and it describes how effectively enterprise services are integrated and composed based on stakeholders' requirements. An EAF is said to be of good quality if it satisfies the quality factors like understandability, completeness, conciseness, portability, consistency, maintainability, testability, usability, reliability and security. Quality factors help to check the efficiency of the proposed EAF on, (i) whether all defined GOALs of the enterprise are achieved; (ii) whether the proposed EAF is scalable and platform independent, (iii) whether the EAF works properly for a certain time period when there is no changes occurred in system, (iv) which structural element object (*SEO*) plays specific role by performing different activities, for a certain time period, (v) whether the services are available to the stakeholders in given time.

A structural element object (*SEO*) may perform different roles (*R*) by executing distinct set of services $\overline{S'}$, in different time periods. Essence is defined as whether quality metrics are being achieved during accessing of the set of services. Essence varies with time (t) and role(*R*), and it can be denoted by *Es*(*R*, *t*).

Formal representation of above description of essence (*Es*) is as follows,

$$Es_{t2}^{t1} \rightarrow (R_i, \overline{S'})$$

Where, $Es_{t2}^{t1}$ denotes Essence from time period from $t_1$ to $t_2$, $R_i$ is a certain role and $\overline{S'} \subseteq S$, *S* is set of services.

$$\exists i\{(R_i \rightarrow \overline{S'}) \wedge (R_i \rightarrow Es_{t2}^{t1})\}$$

The detail example of the Essence concept has been described in the case study and its diagram (section 5).

### 3.2 Relationships in SCORE Architecture

In the above diagram, various intra-layer and inter-layer relationships exist among different constructs of five layers. Inter-layer relationships exist among different types of constructs from different layers and Intra-layer relationships exist among same type of constructs within the same layer. Realized By relationship can be both Inter-layer and Intra-layer relationship, while, containment, association and collaboration relation -ships are Intra-layer relationships. All notations, used to represent the relationships, are given in table 2.

**Table 2.** Summary of Notations of relationships used in SCORE architecture

| **SCORE architecture Relationships** | **Notations** | **Description** |
|---|---|---|
| Containment /Inclusion | <P,Or> ⟶ | Intra-layer relationship where, *P* denotes participation and $O_r$ denotes order of occurrence. |
| Association | <0..*> <0..*> -·-·-·-·-·-·-· | Intra-layer relationship with cardinality |
| Realized By | - - - - - - - - -> | Both Inter-layer and Intra-layer Relationship. |
| Collaboration | — — — — — — — | Intra-layer relationship. |
| Data/Message flow | ⟹ | Inter-layer Relationship. |

(a) *Containment / Inclusion Relationship:* It exists when one construct encapsulates other similar type of constructs. *P* is the participation pattern that is denoted by integer *1* for total participation and integer *0* for optional participation. Order *(Or)*, denoted by an integer, represents the sequence of occurrences of the relationships. Zero (*0*) denotes the sequence of that relationship is not important. Same order of two relationships represents simultaneous occurrences of those two relationships.

(b) *Association Relationship:* It depicts logical or physical connection between two similar types of constructs, by which those constructs can be aggregated to perform any task. Cardinality of this relationship shows number of occurrences in one constructs are connected with the number of occurrences in other type of constructs.

(c) *Realized By Relationship:* It describes how functionality of one type of construct can be realized by other type of constructs.

(d) *Collaboration Relationship:* It represents interactions between two roles.

(e) *Data/ Message Flow:* It depicts direction of data or message flow between five layers of SCORE architecture.

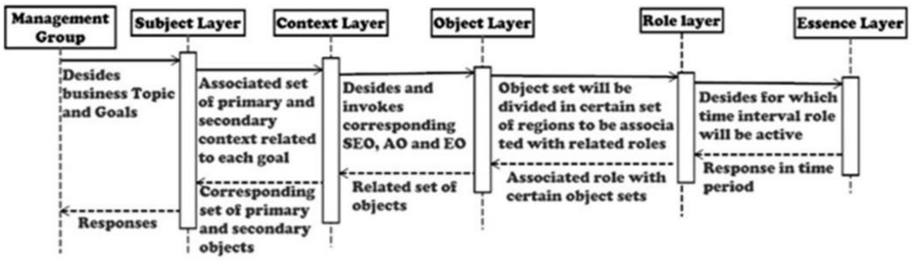

Management Group
Subject Layer
Context Layer
Object Layer
Role layer
Essence Layer
Desides business Topic and Goals
Associated set of primary and secondary context related to each goal
Desides and invokes corresponding SEO, AO and EO
Object set will be divided in certain set of regions to be associated with related roles
Desides for which time interval role will be active
Response in time period
Associated role with certain object sets
Related set of objects
Corresponding set of primary and secondary objects
Responses

**Fig. 2.** Sequence Diagram for inter layer interaction in SCORE

## 4 Analysis of SCORE Architecture using UML Notation

In this section, various inter-layer interactions among all five layers of SCORE architecture are represented and analyzed using UML notations. It demonstrates the behavioral aspects of SCORE framework. Management group interacts with subject layer. In subject layer, set of sub goals are introduced related to each goal. Primary and secondary contexts related to each goal are invoked in context layer. Then, corresponding structural, activity and event objects are to be decided and invoked in object layer. All the roles who handle related objects are invoked in role layer. Essence layer takes care of quality of services. It also describes for what time period a specific role is to be active.

## 5 Illustration of SCORE Architecture using a Case Study

To illustrate the proposed architecture, a case study has been performed on Care management system provided by Electronic Health Record (EHR) System. In this system patients are benefited by the medical guideline and health care plan, after the consultation with specialist doctors. Figure 3 has been illustrated using this case study.

According to SCORE architecture, the Subject layer is all about the business topic and related set of Goal of the organization. Here, business topic (*BT*) is giving Clinical Care of patient. It comprises of certain set of goals like (i) Patient Registration and collection of patient case history (*$G_1$*) and (ii) Medication and Giving care plan and guidelines (*$G_2$*). Goal *$G_1$* has set of sub goals like, (i) Collection of patient information and collection of previous documents and reports (*$SG_1$*) and (ii) Listing out all present problems (*$SG_2$*). Goal *$G_2$* has sub goals like (i) proper diagnosis from all symptoms and medication (*$SG_3$*), (iii) giving care plan and guidelines (*$SG_4$*).

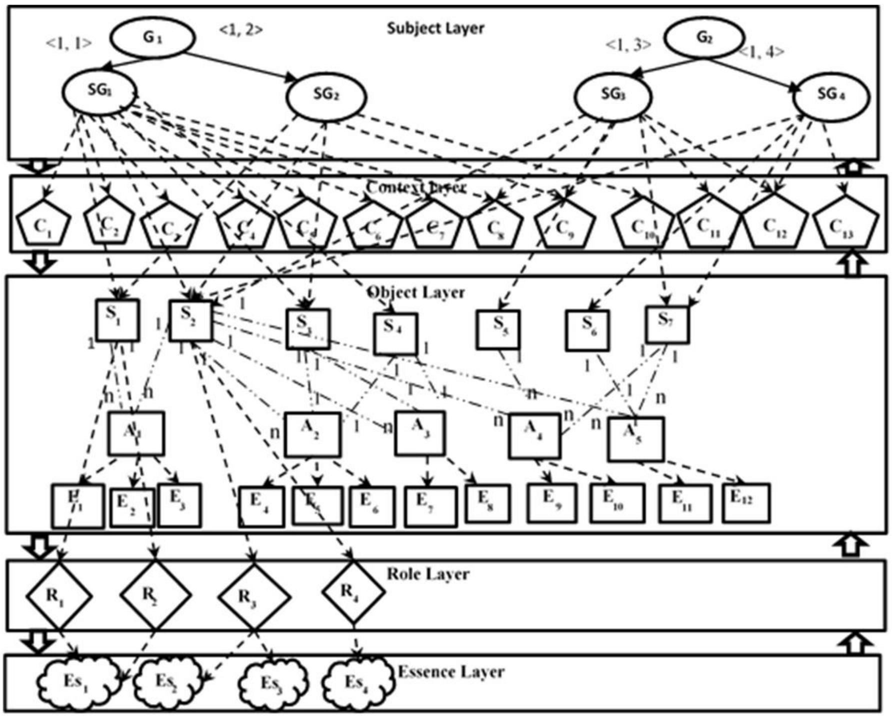


**Fig. 3.** Five Layers of SCORE in Care Management System

Second layer contains context. Patient entity possesses the following attributes as context like patient id (*$C_1$*), age (*$C_2$*), address (*$C_3$*), phone number (*$C_4$*), gender (*$C_5$*), weight

($C_6$), Blood Pressure ($C_7$) and Present Symptoms ($C_8$). Patient id ($C_1$) is primary context and all other contexts are secondary contexts. Any previous report has report id ($C_9$), type of report ( blood report, X ray report, E.C.G Report, Other Report) ($C_{10}$). Prescribed Medicine has primary context, medicine name ($C_{11}$). Generated Health Report has context Report ID ($C_{12}$), Care plan and guideline also have an ID ($C_{13}$) as its primary context.

Third layer contains objects. Here, one type of structural objects is all actors like patients ($S_1$) and the database ($S_2$) where all patient records are stored. Data objects are all information about a patient ($S_3$) that will be stored in database, previous reports ($S_4$) and generated health report ($S_5$) and guidelines and care plan ($S_6$). Interface object is the object where output of any activity object will be reflected. Here electronic gazettes, used for producing output ($S_7$) are used as interface objects. Here, activity objects are activities and functional units like, patient registration ($A_1$), taking patient information and case history and storing all information in database ($A_2$), diagnosis of diseases ($A_3$), prescribing proper medicine ($A_4$) and providing treatment plan and guidelines ($A_5$). All events those are required to perform a specific action are also shown in the following diagram. Activity Object $A_1$ will be performed with help of a set of events like, logging in to the system ($E_1$), entering patient details ($E_2$) and pressing save/submit button ($E_3$). $A_2$ activity can be realized of certain set of events like; entering patient case history ($E_4$), Uploading previous reports ($E_5$) and clicking of save button ($E_6$). $A_3$ activity can be realized by following set of events like, considering all symptoms and performing analysis ($E_7$) and producing result on screen ($E_8$). $A_4$ comprises of events like, searching for effective medicine for that particular diagnosis ($E_9$) and generating health report containing those medicines ($E_{10}$). $A_5$ will be accomplished by realization of events like generating care plan ($E_{11}$) and displaying the plan on output interface ($E_{12}$).

Fourth layer contains role. When a patient enters all information about his disease into the system, and seeks proper diagnosis and medicines for him, then he plays role of Drug Seeker (*R1*), when he finds for care plan and corresponding guidelines, then he plays role of Service Seeker (*R2*). Here, the Database (*S2*) plays different roles by executing different activities. When, a database provides proper diagnosis information and related medicines then it plays role of Drug provider (*R3*). When, database supplies care plan and guidelines then it plays role of Service Provider (*R4*).

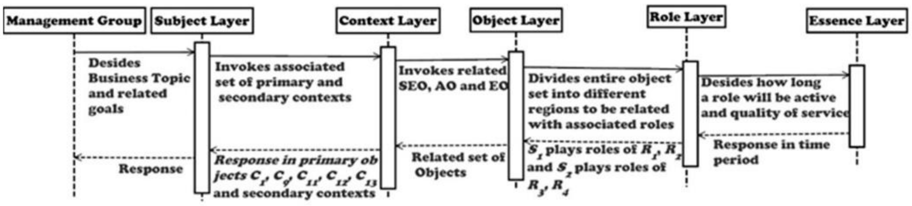
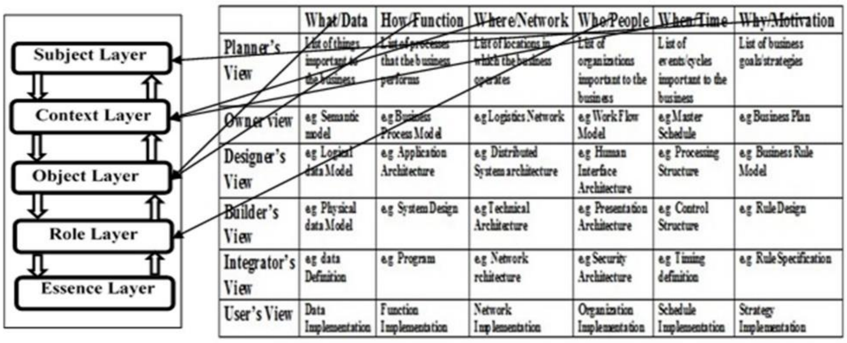

Management Group
Subject Layer
Context Layer
Object Layer
Role Layer
Essence Layer
Desides Business Topic and related goals
Invokes associated set of primary and secondary contexts
Invokes related SEO, AO and EO
Divides entire object set into different regions to be related with associated roles
Desides how long a role will be active and quality of service
Response in time period
$S_1$ plays roles of $R_1$, $R_2$ and $S_2$ plays roles of $R_3$, $R_4$
Related set of Objects
Response in primary objects $C_1$, $C_9$, $C_{11}$, $C_{12}$, $C_{13}$ and secondary contexts
Response

**Fig. 4.** Sequence Diagram for Care Management System

Fifth layer contains essence. Here essence is related to the following matter: (i) Time period in which a structural element object plays a certain role. (*Es1*) (ii) Whether Patient's information and case history has been stored efficiently and in a secured way in database, so that all information will be always available easily in any emergency situation (*Es2*). (iii) Whether this system provides, proper diagnosis and medication service (*Es3*) and (iv) Whether it provides fruitful treatment plan and guidelines to the patients (*Es4*). Figure 4, represents the sequence diagram to demonstrate the sequential interaction among five layers of SCORE framework for care management System.



| | What/Data | How/Function | Where/Network | Who/People | When/Time | Why/Motivation |
|---|---|---|---|---|---|---|
| Planner's View | List of things important to the business | List of processes that the business performs | List of locations in which the business operates | List of organizations important to the business | List of events/cycles important to the business | List of business goals/strategies |
| Owner View | e.g Semantic model | e.g Business Process Model | e.g Logistics Network | e.g Work Flow Model | e.g Master Schedule | e.g Business Plan |
| Designer's View | e.g Logical data Model | e.g Application Architecture | e.g Distributed System architecture | e.g Human Interface Architecture | e.g Processing Structure | e.g Business Rule Model |
| Builder's View | e.g Physical data Model | e.g System Design | e.g Technical Architecture | e.g Presentation Architecture | e.g Control Structure | e.g Rule Design |
| Integrator's View | e.g data Definition | e.g Program | e.g Network rchitecture | e.g Security Architecture | e.g Timing definition | e.g Rule Specification |
| User's View | Data Implementation | Function Implementation | Network Implementation | Organization Implementation | Schedule Implementation | Strategy Implementation |

**Fig. 5.** Mapping of different columns of Zachman Framework to SCORE architecture

## 6 SCORE Framework in The Context of Zachman Framework

The proposed SCORE framework for enterprise architecture is comprised of five loosely coupled layers in contrary to the Zachman framework, which is represented in matrix form. Five layers of SCORE are Subject layer, Context layer, Object layer, Role layer and Essence layer. Subject Layer of SCORE deals with the Business Objective and corresponding set of goals and sub goals hierarchies and it realize the concepts of "Why/Motivation" facets (sixth column) in Zachman framework. However, subject layer of proposed framework is comparatively better capable to provide more detail representation of business objectives. Context Layer of SCORE comprises of all constructs of "Where/Network" column (first Column) and "When/Time" col umn (fifth column) of Zachman framework. In addition, context layer of SCORE includes the facets and related concepts corresponding to the background information related to situatedness and location of different enterprise entities. Third layer in SCORE enterprise architecture, namely object layer, deals with various structural, activity and event objects of an enterprise. Structural objects including various data set, devices can realize the concepts of "What/Data" column in Zachman framework. On the other hand, activity objects along with functional units realize the facets of "How/Function" column in Zachman framework. However, Zachman framework does not give the concept related to events and reconfigurable services in compare to the SCORE architecture. Fourth layer of SCORE containing the concept of roles that can be mapped to "Who/People" (fourth column) concept of Zachman framework. Further, Zachman framework has not considered can artifacts related to the quality requirements including essential non-functional characteristics of enterprise system architecture. However, SCORE has considered such concepts in fifth layer, called Essence Layer. Thus, SCORE provides more broader views and benefits than Zachman framework. Besides, the proposed SCORE framework comprised of different types of relationships (both intra-layer and inter-layer) to exhibit the associativity among the concepts and constructs of enterprise architecture distributed over different layers. Moreover, layered architecture with loosely coupled layers makes SCORE framework more flexible and scalable, as reconfiguration of any components of lower-level layers will hardly affect the upper level layers.

## 7 Conclusion and Future Work

This paper proposes a new enterprise architecture framework, called, SCORE architecture, which is comprised of five loosely coupled layers namely, subject layer, context layer, object layer, role layer and essence layer. Subject layer considers about Business topic, related goal and sub goals hierarchies and motivation of an enterprise. Context layer provides related knowledge about surrounding environment, location, situatedness and time of business entities. Object layer handles various types of objects like, structural element objects, activity objects and event objects of enterprise. The benefits of SOA features are considered inherent of object layer. Role layer contains all type of roles those will be played by different types of structural elements depending upon what kind of activities they performed. Essence layer concentrates on the quality metrics and security metrics of systems. The proposed framework supports several crucial properties like, subject orientation, re-usability, context sensitivity, re-configurability and agility. The SCORE architecture includes the notion of quality of services and nonfunctional properties of EAFs. Proposed EAF is comprised of different relationships those show the interaction and association among various constructs of different layer. The flow of enterprise information through the layers of SCORE architecture is analyzed in using UML notations. A detailed illustration also has been discussed using the case study based on Clinical Care management.

A comparative study also has been performed between the proposed SCORE framework and Zachman framework. It shows that SCORE framework includes several advantages over the Zachman framework in terms of representation of business topics and related goals, enhancement of the facets of “Where” and “When” concepts of Zachman. Further, the proposed EAF provides relevant knowledge about the situatedness and location of business entities. Object layer of SCORE framework integrates SOA properties and notion of event objects makes the framework loosely coupled and adaptive towards business changes. In contrary to the existing SOA-based EAFs, SCORE framework facilitates Subject orientation, Context Sensitivity and Essence features. Moreover, layered architecture of SCORE framework makes it more flexible and scalable in comparison with existing EAFs proposals.

Future work will concentrates on the extension of the proposed SCORE architecture with the detailed formal representation of business subjects, their goals and information flow and interaction mechanism within the different layers. Enhancement of the reconfigurable capability in different layers of SCORE architecture is also a prime objective as future research.

## References

1. Zachman, J. A.: A framework for information systems architecture. IBM systems journal. 26(3), 276-292 (1987)
2. Armour, F. J., Kaisler, S. H., Liu, S. Y.: A big-picture look at enterprise architectures. IT Professional, 1(1), 35–42 (1999)
3. Radwan, A., Aarabi, M.: Study of implementing Zachman framework for modeling information systems for manufacturing enterprises aggregate planning. In: International Conference on Industrial Engineering and Operations Management, pp. 9-14. Kuala Lumpur, Malaysia, (2011)
4. Edward, I. Y. M., Shalannanda, W., Lestariningati, S. I., Agusdian, A.: E-Government Master plan design with TOGAF framework. In: 8th International Conference on Telecommunication Systems Services and Applications (TSSA), pp. 1-6. IEEE, Kuta, (2014)
5. Bailey, I.: Brief Introduction to MODAF with v1.2 Updates. In: IET Seminar on Enterprise Architecture Frameworks. pp. 1-18. IET, London (2008)
6. Wang, W., Gao, C. H., Lin, S. f., Yin, G. f.: Modeling Technologies of FEA Grid Services and Grid Information Management Structure for FEA Grid Portal. In: WRI World Congress on Software Engineering (WCSE '09), pp. 108-113. IEEE, Xiamen (2009)
7. The DoDAF Architecture Framework Version 2.02, `http://dodcio.defense.gov/Portals/0/Documents/DODAF/DoDAF_v2-02_web.pdf`
8. Treasury Enterprise Architecture Framework, `https://www.scribd.com/document/63363136/Treasury-Enterprise-Architecture-Framework-TEAF`
9. Ota, D., Gerz, M.: Benefits and Challenges of Architecture Frameworks. In: Sixteenth International Command and Control Research and Technology Symposium (16TH ICCRTS), Qu´ebec City, Qu´ebec Canada (2011)
10. Kruchten, P. B.: The 4+1 View Model of architecture. IEEE Software, 12(6), 42-50, (1995)
11. Bernus, P., Noran, O., Molina, A.: Enterprise architecture: Twenty years of the GERAM framework, Annual Reviews in Control, 39 (2015), 83-93, (2015)
12. Alwadain, A., Korthaus, A., Fielt, E., Rosemann, M.: Integrating SOA into an Enterprise Architecture- A Comparative Analysis of Alternative Approaches. In: 5th IFIP International Conference on Research and Practical Issues of Enterprise Information Systems (CONFENIS). Natal, Brazil, 2010.
13. Alwadain, A., Fielt, E., Korthaus, A., Rosenmann, M.: Where Do We Find Services in Enterprise Architectures? A Comparative Approach. In: 22nd Australasian Conference on Information Systems (ACIS 2011). Sydney, Australia (2011).
14. Jamjoom, M.M., Alghamdi, A.S., Ahmed, I.: Service Oriented Architecture Support in Various Architecture Frameworks: A Brief Review. In: Ao, S. I., Douglas, C., Grundfest, W. S., Burgstone, J. (eds.) Proceedings of the World Congress on Engineering and Computer Science. vol. 2, pp.1338-1343. Newswood Limited, San Francisco, USA(2012)
15. Wang, L., Luo, A. M.: Service View Description of a Service Oriented C4ISR Architecture Framework. In: IITA International Conference on Services Science, Management and Engineering *(SSME '09)*. pp. 237-241. IEEE, Zhangjiajie (2009)
16. Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., Steggles, P. Towards a better understanding of context and context-awareness. In: Gellersen, H.W.(ed.) Handheld and Ubiquitous Computing. LNCS, vol. 1707. pp. 304-307. Springer-Verlag Berlin Heidelberg, 1999.