

Software Architecture- Evolution and Evaluation

S.Roselin Mary

Department of Information Technology,
Hindustan University,
Chennai, India.

Dr.Paul Rodrigues

CTO,
WisdomTree Software Solutions,
Chennai, India.

Abstract— The growth of various software architectural frameworks and models provides a standard governing structure for different types of organizations. Selection of a suitable framework for a particular environment needs much more detailed information in various aspects and a reference guide of features should be provided. This paper brings out the history of software architecture with a new evolution tree. It also technically analyses well known frameworks used in industries and other governmental organizations and lists out the supportive tools for them. This paper presents the comparative chart that can be used as a reference guide to understand top level frameworks and to further research to enable and promote the utilization of these frameworks in various environments.

Keywords- Framework; Software Architecture; Views.

I. INTRODUCTION

Architecture is playing a vital role to reveal the complexity of a given system. Number of steps will be increased when the system becomes complex. Planning should be done in a detailed manner when the system becomes complex. Architecture comprises the combination of process and product of planning, designing and constructing space to reflect functional, social and aesthetic considerations [21]. Planning for buildings and complexity behind this will be interrelated in civil engineering. As customers and constructors have their own views in a particular subject, the architecture should solve it in a unique manner by covering all of them [13].

Likewise the same concept in software is called software architecture. The term and concept of Software architecture was brought out by the research work of Dijkstra in 1968 and David Parnas in 1970's. The interconnected basic building components and the views of end user, designer, developer and tester are needed to build a complicated, critical system. The design and implementation of the high-level structure of the software are the backbone of software architecture. The architectural elements will be interconnected in well-known manner to get the major functionality and performance requirements of the system and to obtain non-functional requirements such as reliability, scalability, portability, and availability [12]. Software frameworks point out the suitable places in the architecture where specific functionality can be adapted by application programmers [17]. A software framework provides an abstraction where generic functionality can be selectively overridden or specialized by user code. The overall development time will be cut into minimum as it concentrates on the low level details of a working system. So,

the designers and programmers can concentrate only on the software requirements. [7].

The rest of the paper is organized as follows. Section II briefly describes the history of Software architecture and the figure Fig.1 given below clearly portrays the evolution. Section III classifies the frameworks. Section IV and V summarize and compare the different frameworks.

II. HISTORY OF SOFTWARE ARCHITECTURE

The basic principles of 'software architecture' have been applied since the mid 1980's and it crossed various stages from algorithm's era by borrowing the concepts from others to get a shaped form. In 1928, An Algorithm was formulated to solve the problem by the finite sequence of instructions. Von Neumann developed 'Flow Chart' that has a visual representation of the instruction flow, to plan computer programs in 1947 by inheriting the idea from the flow process chart(1921) and multi flow chart(1944) which were used mostly in the area of electrical engineering. But, there is a gap to point out the flow of control. So, 'Control Flow Diagram' (CFD) was developed in the late 1950's to describe the control flow of a business process and program. This was not enough to view the complex systems. The high level view of the work and immediate access of particular points can't be represented using this diagram. So, to reveal the entire system by dividing into blocks, 'Block Diagram' was developed in late 1950's. A specific function for each block and the connection between blocks will be shown in a diagram.

The introduction of abstraction concept became a booster in the field of software architecture. It made a revolution and tremendous growth to that area. By that way, data structures that have similar behaviour, data structures that have similar behaviour, certain data types and modules of one or more programming languages that have similar semantics are grouped in the late 1960's. This was happened by the introduction of Abstract data types. It again leads to 'Modular Programming' that introduces the concept of separate parts called modules in software in 1968. Separation of concerns with the logical boundaries between components is called as modules.

In 1977, 'Three Schema Approach' that adopts layered architecture based on the modular programming was developed. It is used to build information systems using three different views in systems development. Here an application will be broken into tiers and developers have to modify a

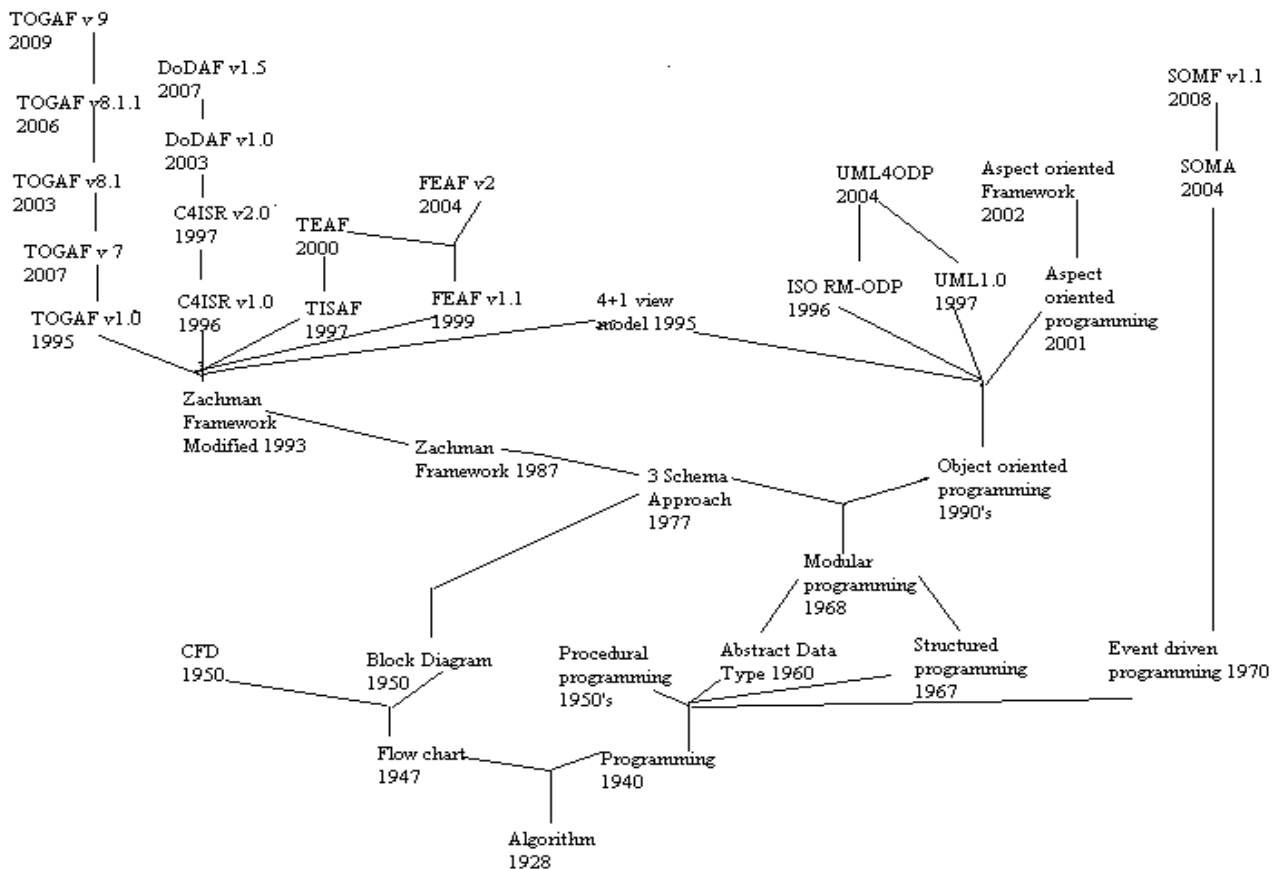


Figure 1. Evolution of software Architecture

specific layer not to rewrite the entire application over. Flexible and reusable applications can be developed using this scheme.

Later, based on this three tier approach, a layer of six perspectives was introduced in 1987 by John Zachman .That is called as ‘The Zachman Framework’ which still plays an important role in the era of ‘Enterprise Architecture’ and influenced frameworks DODAF, TOGAF, TEAF and FEAF. In 1993Zachman released the modified version of Zachman Framework with more number of views. In 1995, 4+1 view model was developed by Kruchten.

Views are used to analyze the complex systems, and to list out the problem elements and the solution. A view of a system suppresses details. It focuses on specific concerns of the system. It provides a simplified model [13] [12].

U.S Government encouraged the researchers to develop the frameworks for defense side applications and it leads to the C4ISR Architecture Framework in 1996. ‘The Department of Defense Architecture Framework (DODAF)’ was released n 2003, which restructured the C4ISR framework ver.2.0 [19] [6].

The restructured C4ISR framework ver.2.0 was released as, ‘The Department of Defense Architecture Framework (DODAF)’ in 2003[19] [6]. ‘The Open Group Architecture Framework (TOGAF)’ was developed by the members of

open architecture forums in 1995. Recently in 2009, TOGAF Version 9 was released [15].

To integrate its myriad agencies and functions under single common and enterprise architecture, the ‘Federal enterprise Architecture Framework (FEAF)’ was developed in 1999 by the Federal Government [18].

‘Treasury Enterprise Architecture Framework (TEAF)’ was developed to support the Treasury’s business processes in terms of products of the US Department of Treasury and published in July 2000 [20].

A reference model RM-ODP was developed by Andrew Herbert in 1984. It combines the concepts of abstraction, composition and emergence on the distributed processing developments. By including the set of UML profiles in the ODP and UML4ODP was released in 2004[10].

In 2001, Aspect oriented programming boom out by inheriting the principles of OOPS. And, it leads to the Aspect oriented software development in later 2002.

IBM announced ‘Service Oriented Modeling Architecture (SOMA)’ in 2004 opposing the distributed processing and Modular programming. It is the first publicly announced SOA related methodology. In addition to this, to provide tactical and strategic solutions to enterprise problems, the SOMF ver 1.1 was released by Michael Bell [4][5].

This section clearly portrays that Zachman framework paves a way to build so many frameworks on it. The application of UML on RM-ODP derives a new framework. This analysis invokes why not to develop new frameworks by combining some existing technology to yield a better framework. The frameworks dealt in the next sections are most widely used for the commercial and Government departments. So, it is necessary to classify and compare them.

III. CLASSIFICATION OF FRAMEWORKS

Classification is the problem of identifying which of a set of categories a new observation belongs to. As the frameworks were developed under the interests of different field masters, they were influenced by various perspectives. So, it is necessary to classify them as whether they are developed by standard bodies or individual interests or by private agencies.

The frameworks developed by standard bodies fall under the standard category and others fall under nonstandard category. And also they are subcategorized based on their usage in commercial or Government purpose.

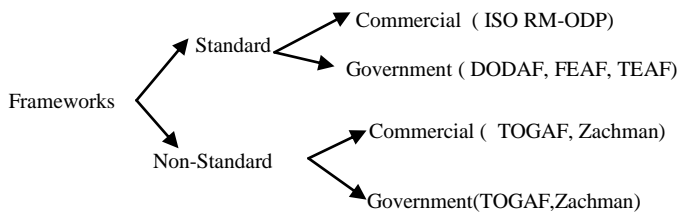


Figure 2. Classification

Frameworks developed and used for the Government departments and for Defense side applications are classified under the Government frameworks. Frameworks used for commercial purpose are classified under the commercial frameworks.

The Open Distributed model ISO RM-ODP falls under the standard and commercial frameworks. DODAF, FEAF and TEAF which were developed for the U.S Government agencies are coming under the standard and government frameworks. The well accepted and most widely used frameworks, TOGAF and Zachman frameworks are used by both the commercial and government agencies.

Even though TOGAF and Zachman frameworks are falling under non-standard category, mapping of these frameworks to DODAF, FEAF and other standard frameworks yielded good products in the industry. The classification described in this section will be very much useful for the customer to choose the suitable framework quickly for his organization based on the job nature also. The next subsection deals the comparison parameters that can be used by the customer to choose the appropriate tool. The following section analyses the well-known frameworks and lists out their criteria.

IV. EVALUATION OF VARIOUS FRAMEWORKS

In this paper, we have taken the survey of few frameworks which are most widely used. The parameters used for comparison in existing surveys are not suitable for a customer

to choose the tool. So, the methodologies, techniques and tools used in these frameworks are considered for the comparison.

A. Zachman Framework

The Zachman Framework describes the complex thing in different ways using different types of descriptions. It provides thirty-six categories to describe anything completely.

1) *Views / Viewpoints*: It has six different views to facilitate each player to view the system in their own particular way.

- Planner's View (Scope)
- Owner's View (Enterprise or Business Model)
- Designer's View (Information Systems Model)
- Builder's View (Technology Model)
- Subcontractor View (Detailed Specifications)
- Actual System View

2) *Domain*: It mainly focuses on Categorizing Deliverables [8].

3) *Origin*: This framework is well suited for Manufacturing Industries [8].

4) *Focus*: It focuses mainly on Business process.

5) *Phase of SDLC*: In the Design stage or planning stage, it can be used [8].

6) *System development methodology*: Organization's own methodology can be followed.

7) *System modeling Technique*: OMG-Model driven Architecture, Organization's own technique

8) *Business Modeling Technique*: BPML is used for this framework.

9) *Advantages* :

- Provides improved professional communications within community [22].
- Understanding the reasons for and risks of not developing any one architectural representation [22].
- Provides variety of tools and/or methodologies [22].
- Developing improved approaches [22].

10) *Weakness*:

- It may lead to more documentation depending on the cases [2]
- It may guide to a process-heavy approach to development [2].
- It isn't well accepted by all the developers [2].
- It seems in its first appearance as a top-down approach to developers. [2].
- It is to be biased towards traditional and data-centric techniques. [2].

B. NATO Architecture Framework / C4ISR / DoDAF

The Department of Defense Architecture Framework (DoDAF) provides the organization of an enterprise architecture (EA) into consistent views. It is well suited for large complicated systems and interoperability challenges. "Operational views" used here are to deal with the external customer's operating domain.

1) *Views / Viewpoints*: DoDAF provides multiple views, each of which describes various aspects of the architecture. DoDAF defines the following views:

- Overarching All View (AV).
- Operational View (OV).
- Systems View (SV).
- Technical Standards View (TV).

2) *Domain*: It mainly focuses on operating domain [8].

3) *Origin*: This framework is developed for Defence [8].

4) *Focus*: It focuses mainly on Architecture data and Business process.

5) *Phase of SDLC*: It is used in a Process or Planning stage [8].

6) *System development methodology*: The Framework does not advice the use of any one methodology. It depends on the organization's decision.

7) *System modeling Technique*: If the system to be developed is larger, then UML tools are likely the best choice.

8) *Business Modeling Technique*: IDEF Family

9) *Advantages*:

- Defines a common approach for describing, presenting, and comparing DoD enterprise architectures [19].
- Common principles, assumptions and terminology are used [19].
- Across the organizational boundaries architecture descriptions can be compared [19].
- Deployment costs and reinvention of same system can be reduced. [9].

10) *Weakness*:

- No common ontology of architecture elements [1].
- Baseline (current) and objective (target) architectures are not addressed [1].
- How the architectures can be used to measure effectiveness is not dealt [1].
- Business-financial plans are not addressed. [1].

C. TOGAF

The Open Group Architecture Framework (TOGAF) provides a comprehensive approach to the design, planning, implementation, and governance of enterprise information architecture.

1) *Views / viewpoints* : TOGAF identifies many views to be modeled in an architecture development process. The architecture views, and corresponding viewpoints come under the following categories:

- Business Architecture Views
- Information Systems Architecture views
- Technology Architecture views
- Composite views

2) *Domain*: It mainly focuses on Business, data and applications [8].

3) *Origin*: This framework is developed due to the motivation in Defence side framework.

4) *Focus*: It focuses mainly on Business process, Data, applications and Technology.

5) *Phase of SDLC*: It is used in a Process or Planning stage [8].

6) *System development methodology*: Rational Unified process (RUP) is used as a system development Methodology.

7) *System modeling Technique*: UML, BPMN are widely used in TOGAF system modeling.

8) *Business Modeling Technique*: IDEF is used for business modeling in TOGAF

9) *Advantages* :

- Increased transparency of accountability [24].
- Controlled risk [24].
- Protection of assets [24].
- Proactive control [24].
- Value creation [24].

2) *Weakness*:

- Lots of Detail [16].
- Planning methods and governance framework [15].
- Weak on Information Architecture [15].
- Can lead startup efforts into too much too soon [16].

D. TEAF

Treasury Enterprise Architecture Framework (TEAF) was developed by the US Department of the Treasury and published in July 2000. It is based on the Zachman Framework.

The Treasury Enterprise Architecture Framework (TEAF) supports Treasury's business processes in terms of products. This framework guides the development and redesign of the business processes for various bureaus.

1) *Views / Viewpoints*: It provides four different views.

- Functional Views
- Information View
- Organizational View
- Infrastructure View

2) *Domain*: It has a domain on Business processes [20][8].

3) *Origin*: This framework is developed for Treasury department [20].

4) *Focus*: It focuses mainly on Business process.

5) *Phase of SDLC*: It is used in a communication or Planning stage [8].

6) *System development methodology*: It does not refer any specific methodology. It depends on the organization's decision.[23].

7) *System modeling Technique*: Flow chart, UML can be used in TEAF.

8) *Business Modeling Technique*: IDEF, ERD can be used as business modeling techniques.

9) *Advantages* :

- Provides the guidance to the treasury bureaus and offices in satisfying OMB and other federal requirements [20].

- Support Treasury bureaus and offices based on their individual priorities and strategic plans [20].
- Leads to Treasury-wide interoperability and reusability [20].

10) *Weakness:*

The TEAF does not contain a detailed description of how to generate the specification documents (work products) that are suggested for each cell of the TEAF Matrix [14].

E. FEAF

Federal Enterprise Architecture (FEA) was developed for the Federal Government to provide a common methodology for information technology (IT) acquisition, use, and disposal in that Federal government. It was built to develop a common taxonomy and ontology to describe IT resources. The FEAF provides documenting architecture descriptions of high-priority areas. It guides to describe architectures for functional segments in multi-organization manner of the Federal Government.

1) *Views / Viewpoints:* Like zachman framework, FEAF is also having five different views in its framework.

- Planner's View (Scope)
- Owner's View (Enterprise or Business Model)
- Designer's View (Information Systems Model)
- Builder's View (Technology Model)
- Subcontractor's View (Detailed Specifications)

2) *Domain:* It has a domain on provision of services [8].

3) *Origin:* This framework is well suited for Enterprise Architecture planning.

4) *Focus:* It focuses mainly on Business process, Data, Application and Technology.

5) *Phase of SDLC:* It is used in a Communication or Planning stage [8].

6) *System development methodology:* RUP (Rational Unified process) is followed in FEAF.

7) *System modeling Technique:* UML is used as a system modeling tool in FEAF.

8) *Business Modeling Technique:* BPML is the technique used in FEAF.

9) *Advantages :*

- Serve customer needs better, faster, and cost effectively [18].
- Promote Federal interoperability [18].
- Promote Agency resource sharing [18].
- Reduced costs for Federal and Agency [18].
- Improve ability to share information [18].
- Supports capital IT investment planning in Federal and Agency [18].

10) *Weakness:*

- The Federal Government could risk allocating too much time and resources to an enterprise architecture description effort yielding potentially little return at significant cost [18].

- The Federal Enterprise Architecture program requires technical and acquisition expertise [18].
- The Federal IT community must keep its eyes on the basic principles rather than near-term objectives and achievements [18].
- The Federal Government has to pay up-front for the right to exercise options in the future [18].
- Concern over territoriality and loss of autonomy may impede the Federal Enterprise Architecture effort due to long-term, realignment of Agency functions and responsibilities [18].
- It is hard to have common, cross-Agency models and standards to ensure interoperability [18].

F. ISO RM-ODP

The ISO Reference Model for Open Distributed Processing provides a framework standard to support the distributed processing in heterogeneous platforms. Object modeling approach is used to describe the systems in distributed environment.

1) *Views / Viewpoint:* The five viewpoints described by RM-ODP are:

- The enterprise viewpoint
- The information viewpoint.
- The computational viewpoint
- The engineering viewpoint
- The technology viewpoint

2) *Domain:* It has a domain on information sharing in distributed environment.

3) *Origin:* This framework is well suited for major computing and telecommunication companies.

4) *Focus:* It focuses mainly on Business process, Technical Functionality and Solution.

5) *Phase of SDLC:* It is used in a Processing and communication stage.

6) *System development methodology:* Object oriented method and IAD can be followed here [3].

7) *System modeling Technique:* UML, OMG (Model driven Architecture) are used as system modeling techniques [3].

8) *Business Modeling Technique:* BPMN is used as business modeling technique in RM-ODP.

9) *Advantages :*

- It provides lot of details for the analysis phases of the development of applications [3].
- It provides the platform to integrate the requirements from different languages consistently. [3].

It provides a set of established reasoning patterns to identify the fundamental entities of the system and the relations among them. It provides the appropriate degrees of abstraction and precision for building useful system specifications [3].

TABLE 1 COMPARATIVE CHART FOR FRAMEWORKS

S/N	Frame work TERMS	ZACHMAN FRAMEWORK http://zachmaninternational.com	DoDAF Cio-nii.defense.gov/docs/DoDAF_Volume_II.pdf	TOGAF http://www.opengroup.org/architecture/	TEAF www.treas.gov/cio	FEAF www.cio.gov/documents/fedarch1.pdf	ISO RM-ODP http://www.rm-odp.net/
1	No of Views	Six	Four	Four	Four	Five	Five
2	Domain	Categorizing deliverables	Operating domain	Business, Data and Applications	Business processes	Provision of services	information sharing in distributed environment
3	Origin	In- Manufacturing	Defence	Defence	Treasury Department	Enterprise Architecture planning	major computing and telecommunication companies
4	Focus	Business process	Architecture Data, Business process	Business process, Data, Applications, Technology	Business processes	Business process, Data, Applications & Technology	Business process, Technical functionality & Solution
5	Phase of SDLC	Planning (Design)	Process/Planning	Process/Planning	planning / communication	Planning & communication	Processing & communication
6	System development methodology	Organization' own methodology	Organization' own methodology	RUP	Organization' own methodology	RUP	Object oriented method, IAD
7	System modeling technique	OMG-Model driven Architecture, Organization's own technique	UML	UML, BPMN	Flow chart, UML	UML	UML, OMG(Model driven Architecture)
8	Business model technique	IDEF	IDEF Family	IDEF Family	IDEF, ERD	BPML	BPMN
9	Advantages	<ul style="list-style-type: none"> Improving professional Communications wide variety of tools improved approaches For Architectural representations 	<ul style="list-style-type: none"> common Approach common principles, assumptions and terminology comparable architecture descriptions across organizational boundaries reduction of deployment costs 	<ul style="list-style-type: none"> increased transparency of accountability controlled risk protection of Assets proactive Control value creation 	<ul style="list-style-type: none"> satisfying OMB support individual Priorities and strategic Plans interoperability and reusability 	<ul style="list-style-type: none"> serve customer needs better, faster and cost effectively promote federal Interoperability provide agency resource sharing reduced costs improve ability to share information support Federal and agency capital IT investment planning 	<ul style="list-style-type: none"> improved requirement collection and analysis phases consistently integrated requirements expressed in separate languages set of already established reasoning patterns used for building robust, efficient and competitive applications backed by industrial products with enough acceptance
10	Weakness	<ul style="list-style-type: none"> documentation heavy approach process heavy approach to development seems like Top down Approach biased towards traditional, data centric techniques 	<ul style="list-style-type: none"> No common ontology Of architecture elements not addressing baseline and objective architectures not addressing capabilities and measures of effectiveness lack of business financial artifacts 	<ul style="list-style-type: none"> lots of detail planning methods and governance framework weak on information Architecture can lead startup efforts into too much too soon 	<ul style="list-style-type: none"> No detailed description of Specification document for each cell Missing the techniques for creating specification document 	<ul style="list-style-type: none"> little return at significant cost need technical and acquisition expertise need a watch on future less future Maneuverability loss of autonomy may impede effort difficult to ensure interoperability 	<ul style="list-style-type: none"> problem of inter-view Consistency Not a truly guaranteed cross-view checks No precise notion of Consistency
11	Tools	<ul style="list-style-type: none"> Adaptive EA Manager Mega V6.1 SystemArchitect V10 Simon Tool 	<ul style="list-style-type: none"> EA Webmodeler Corporate Modeler Enterprise Edition 10 SystemArchitect V10 Metis product family 	<ul style="list-style-type: none"> System Architect 10 Metastorm ProVision EA V6.0 IDS Scheer EA Webmodeler 	<ul style="list-style-type: none"> EA Webmodeler Corporate Modeler Enterprise Edition v10 FEAMS V0.2 Metis product family 	<ul style="list-style-type: none"> Adaptive EA Manager Flashline4 FEAMS V0.2 SystemArchitect V10 	<ul style="list-style-type: none"> ConsVISor TINA Simon Tool MagicDraw

- It provides a set of mechanisms and common services to build robust, efficient and competitive applications, interoperable with other systems [3].

10) Weakness:

RM-ODP has the problem of inter-view and inter-view consistency. A number of cross-view checks to be done to maintain the consistency. These checks don't guarantee the consistency [11].

V. COMPARATIVE CHART OF FRAMEWORKS

Table 1 given above describes the comparison between all the discussed frameworks. It has precise data for the user with the additional information of available supportive tools.

VI. CONCLUSION

This paper presents an overview of software architecture and reviewed the evolution of software architecture. By seeing the evolution tree, one can easily understand the evolution. Well known frameworks have been studied and discussed in detail in this paper. It summarizes the frameworks based on the industry side criteria and it discusses the benefits and drawbacks of each framework. The comparative chart included here clearly figures out the frameworks and it can be used as the reference guide also. It will invoke the user to choose the right framework for their industry, organization and business based on their requirement. Users can easily identify the supporting tools available for their frameworks. All the frameworks analyzed here are mainly focusing on business and IT solutions. In future ancient Indian architecture styles can be mapped to the familiar Frameworks to yield new frameworks to focus on quality.

REFERENCES

[1] Alessio Mosto.: DoD Architecture Framework Overview [Online]. Available : www.enterprise-architecture.info/Images/.../DODAF.ppt (2004, May.)

[2] Ambler, S. Extending the RUP with the Zachman Framework.,<http://www.enterpriseunifiedprocess.com/essays/ZachmanFramework.html> (2007).

[3] Antonio Vallecillo.: RM-ODP: The ISO Reference Model for Open Distributed Processing. ETSI Informática, Universidad de Málaga , www.enterprise-architecture.info/Images/Documents/RM-ODP.pdf

[4] Bell, Michael.: "Introduction to Service-Oriented Modeling", in Service-Oriented Modeling: Service Analysis, Design, and Architecture. Wiley & Sons, (2008)

[5] Buckalew P.M.: Service Oriented Architecture.,<http://www.pmbuckalew.com/soa.htm> (2009).

[6] Cris Kobryn and Chris Sibbald.: Modeling DODAF Complaint Architectures., www.uml-forum.com/.../White_Paper_Modeling_DoDAF_UML2.pdf (2004, Oct. 25).

[7] HighBeam Research.:Software Framework., http://www.reference.com/browse/SoftwaRRre_framework (2008).

[8] Jaap schekkerman.: A comparative survey of Enterprise Architecture Frameworks. Institute for Enterprise Architecture Developments, Capgemini., www.enterprise-architecture.info (2003).

[9] Jim.: Applicability of DODAF in Documenting Business Enterprise Architectures, <http://www.thario.net/2008/08/applicability-of-dodaf-in-documenting.html> (2008, Aug. 9).

[10] Juan Ignacio.: UML4ODP PLUGIN – User guide Version 0.9, Atenea Research Group, Spain., http://issuu.com/i72jamaj/docs/uml4odp_plugin (2009).

[11] Mark Maier and Eberhardt Reichtin, "Architecture Frameworks" in The art of systems architecting,2nd ed.CRC Press, Florida, pp. 229-250.(2000).

[12] Philippe Kruchten: "Architectural Blueprints – The "4+1" View model of software Architecture" IEEE Softw. Vol. 12, pp. 42-50, (1995)

[13] Roger Session: A Comparison of Top Four Enterprise –Architecture Methodologies, ObjectWatch, Inc., www.objectwatch.com/white_papers.htm(2007, May)

[14] Susanne Leist, Gregor Zellner.: Evaluation of Current Architecture Frameworks . University of Regensburg, Germany, www.dcc.uchile.cl/~vramiro/d/p1546-leist.pdf (2006).

[15] The Open Group.: Module 2 TOGAF9 Components. www.opengroup.org/togaf/ (2009).

[16] Tim Westbrook.: Do Frameworks Really Matter? , EADirections, www.eadirections.com/.../EADirections%20Frameworks%20Breakout%20updated.pdf (2007, Oct. 24).

[17] Tony C Shan .: "Taxonomy of Java Web Application Frameworks" in Conf. Rec. 2006 IEEE Int. Conf. e-Business Engg., pp. 378-385.

[18] U.S. Chief Information officers (CIO) Council. :Federal Enterprise Architecture Framework Version 1.1 .www.cio.gov/documents/fedarch1.pdf (1999, Sep).

[19] U.S. Dept. of Defense.: DoD Architecture Framework Version 1.5. cni.defense.gov/docs/DoDAF_Volume_II.pdf (2007, Apr. 23).

[20] U.S. Treasury Chief Information officer Council.: Treasury Enterprise Architecture Framework Version 1, www.treas.gov/cio (2000, Jul.).

[21] Will Conely: About Architecture . http://www.ehow.com/about_4565949_architecture.html (2009).

[22] Zachman, J. A.: "A Framework for Information Systems Architecture". IBM Syst. J. Vol. 26, No. 3,pp. 276 – 292, Apr. (1987.)

[23] "Treasury Enterprise Architecture Framework" , en.wikipedia.org/.../Treasury_Enterprise_Architecture_Framework

[24] "What is TOGAF?" ,<http://www.articlebase.com/information-technology-articles/what-is-togaf-626259.html>

AUTHORS PROFILE

Mrs. S. Roselin Mary is presently working as an Assistant professor in the department of Computer science and engineering, Anand Institute of Higher technology at Chennai. She obtained her B.E(CSE) degree from Madurai Kamaraj University and M.Tech(CSE) from Sastra University, Thanjavur. She is now pursuing her research in Hindustan University, chennai; India. She has 8 years of teaching experience.

Dr.Paul Rodrigues is a Chief Technology Officer, WSS at chennai. He has many years of experience in industries and teaching. He guided many Ph.d's and published more than 25 papers in various journals. He was formerly professor in Kalasalingam college of engineering, Tamilnadu and worked as Dean (IT & MCA)in Hindustan University.