Towards an integrated architecture framework

Abeer Samy Yousef Mohamed and Dina Ahmed Ahmed El.melegy
Architecture Engg., Dept., Faculty of Engg., Tanta University, Tanta, Egypt
drabeersamy@yahoo.com
dina_ahmed25@hotmail.com

In the architecture, engineering and construction industries, computer usage can cover the whole lifecycle of a product from presentation of initial concepts to the final stages of production and can also extend to maintenance issues. Although architecture enjoys an enormous and ever increasing attention, it is not always correctly understood and applied. One of the most ambiguous words in the initial understanding of Information Architecture is that of the Framework. An Architecture Framework provides tools and methods that can bring the task into focus and allow valuable artifacts to be produced when they are most needed. Architecture Frameworks are commonly used in Information technology and Information system governance. So This paper initially reports on making a clear view about undersatanding the meaning of architectur and its rules from the vision of Integrated Architecture, also an architecture framwork and how to make an Integrated Architecture Framework (IAF).

فى العمارة والهندسة والصناعات الإنشائية يمكن استخدام جهاز الكمبيوتر لتكوين أى منتج بدءاً من كونه فكرة مجرده قيد البحث حتى الوصول إلى المراحل النهائية لإنتاجه واستمرارية انتاجه وصيانته. وبالرغم من أن العمارة ذات أهمية كبيرة إلا أنها ليست دائماً يتم استيعابها وتطبيقها بشكل صحيح . فأحد الكلمات الأكثر غموضا فى التفهم المبدئى للعمارة المعلوماتية تتمثل فى الإطار المعمارى "framework" أي الكيفية التي يتم بها تنظيم العمل المعماري. إن الإطار المعمارى يتضمن أدوات وأساليب تمكن من تسليط الضوء على نتاج معين مع السماح بظهور نتاجات معمارية ذات قيمة إبداعية متفردة ومتمايزة عند الحاجة لذلك . كما أن الأطر المعمارية تستخدم فى أغلب الأحيان فى تكنولوجيا المعلومات والنظم المعلوماتية للحكومة. ولذلك فهذا البحث يؤكد على عمل رؤية واضحة لفهم معنى العمارة وقواعدها من خلال إطار معمارى متكامل وكيفية عمل هذا الإطار المعمارى المتكامل " Integrated Architecture Framework (IAF)".

Keywords: Architecture, Integrated architecture, Integrated Architecture Framework (IAF)

1. Introduction

The challenges of the knowledge society cannot be met without getting at the contents of the vast volume of digital information. Given today's fast pace of technological evolutions and diversity of computing platforms (both hardware and software), building applications which can work in such a wide range of systems is becoming more and more challenging. Moreover, even on a specific platform, the execution context and available resources tend to vary a lot at runtime.

Most complex decisions involve many data, human, and technological sources collaborating to support decision makers. However, when responsibility for task accomplishment moves from the province of one person to a multitude of natural and artificial intelligences, the system changes quantitatively and qualitatively. Quantitatively, the system is more complex and dynamic. This complexity increases further as the constituent intelligences are separated in time. Qualitatively, the system exhibits properties that were not evident when a lone individual is working on a set of tasks.

Experiences made with massively-parallel computers show a high discrepancy between peak and sustained performance for many applications in computional science. This is often a consequence of the fact that parts of the application are not well suited for the architecture they execute on. With increasing size and complexity of the implementations of information systems, it is necessary to use some logical construct (or architecture) for defining and controlling the interfaces and the integration of all of the components of the

system according to the clear understanding of architecture and its rules.

So, this paper defines architecture and engineering, then by analogy Integrated Architecture. Also, some preliminary conclusions about Integrated Architecture Framework (IAF) and its fundamental principles that can make architects benefits from it.

2. Designing and engineering

Before defining architecture, the terms 'designing' and 'engineering' must be made clear. The clarification of the term 'designing' is needed for making a distinction between the design and the architecture of a system. All these terms refer to how systems are constructed. According to [1], two different system notions exist: the teleological and the ontological system notion. The teleological system notion is about the function and the (external) behavior of a system. This notion can be visualized with a black-box model. The teleological system notion is adequate for the purpose of using or controlling a system. The ontological system notion, on the other side, can be used for building or changing a system. It is about the construction and operation of a system and can be modeled with a white-box model, fig. 1.

Both the teleological and the ontological system notion are relevant for designing a system, [1]. In other words: both the functional and the constructional perspective of a system are relevant. In software, the terms functional design and technical design are often used to refer to these perspectives. How to make these designs is described in a so-called software development process. Such a process also shows the connection between the different models. A lot of such software development processes exist in literature. All of them have their own advantages and disadvantages. However, commonalities exist in these processes. As the name already states this process is not only suitable for software development but it holds for each system you want to design.

So the *designing* of a system essentially consist of two things: [2].

- Determining the (functional) requirements
- Devising the (constructional) specifications

When the design of a system is completed, the system can be engineered. If the design processed by the system design process is specified as an *ontology* [2], the process of engineering a system is like the one shown in figure 2. An ontology model of a system is fully independent of the implementation, it only shows the essential features. Good examples of such ontology's are the SMART model for system development and the DEMO (Design and Engineering Methodology for Organizations) approach to enterprise ontology (considering enterprises also as a system), [3].

3. What is architecture?

Architecture is that set of design artifacts, or descriptive representations, that are relevant for describing an object, such that it can be produced to requirements as well as maintained over the period of its useful life. Another definition which looks like the previous one reads like:

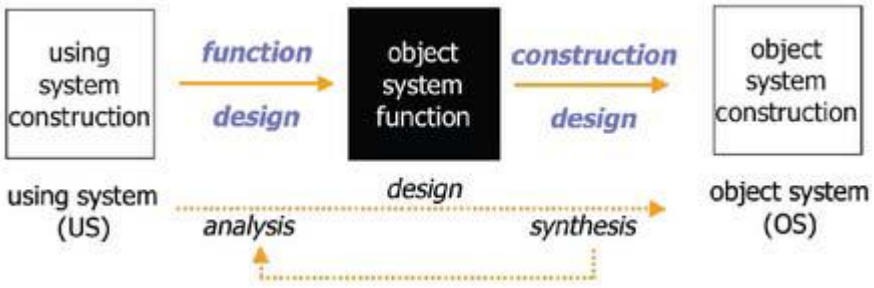

Fig. 1. The generic system design process, [1] .

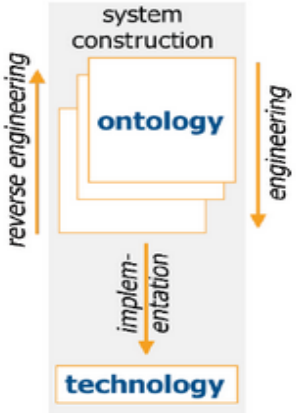

Fig. 2. The process of engineering a system, [1].

An architecture is the set of significant decisions about the organization of a software system, the selection of the structural elements and their interfaces by which the system is composed, together with their behavior as specified in the collaborations among those elements, the composition of these structural and behavioral elements into progressively larger subsystems, and the architectural style that guides this organization, these elements and their interfaces, their collaborations, and their composition, [4].

A representation of a defined domain in terms of its component parts, what these parts do, how the parts relate to each other, and the rules and constraints under which the parts function, [5]. So, the architecture description process can be explained [6]:

- In general, the design freedom of designers is undesirably large. The idea of architecture is to take advantage of this. Therefore, architecture is defined as normative restriction of design freedom, fig. 3.
- This idea of consciously applying normative restriction of design freedom is the really new thing. It makes architecture a prescriptive notion; any descriptive interpretation is cogently rejected.
- This notion of architecture is in contrast with the one that is used in the construction of buildings. There, architecture is hardly distinguished from design.
- Practically, architecture is a consistent and coherent set of design principles that embody general requirements. Applying a design principle satisfies one or more general requirements.

Many other definitions can be found saying the same in other words. As in all three of the given definitions architecture is defined as the description of a system. These definitions of architecture look like what is called ***ontology*** in the previous part of this paper. When adopting this definition the difference between design (expressed in an ontology model [7], and architecture does no longer exist. So, why introducing the term architecture if the term design already is known? According to [8], the given definitions can be referred to as the *descriptive* notion of architecture.

Theoretically, architecture is the normative restriction of design freedom. Practically, architecture is a consistent and coherent set of design principles. Combining this definition of *architecture* with the given definitions of *design* and *engineering*, architecture can be visualized as shown in fig. 4.

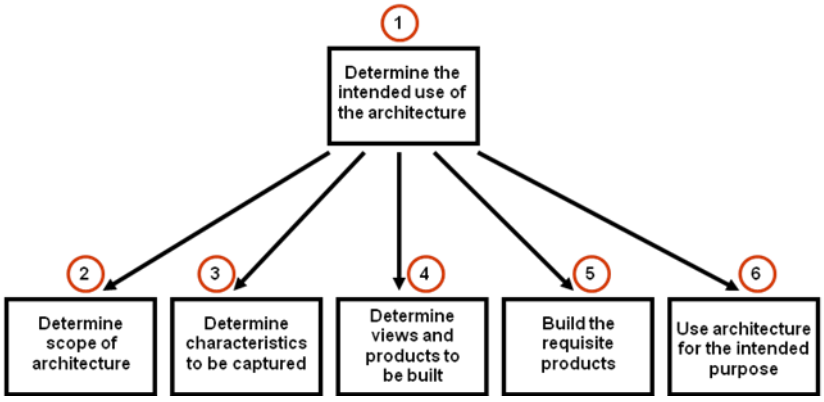

Fig. 3.The architecture description process [6].

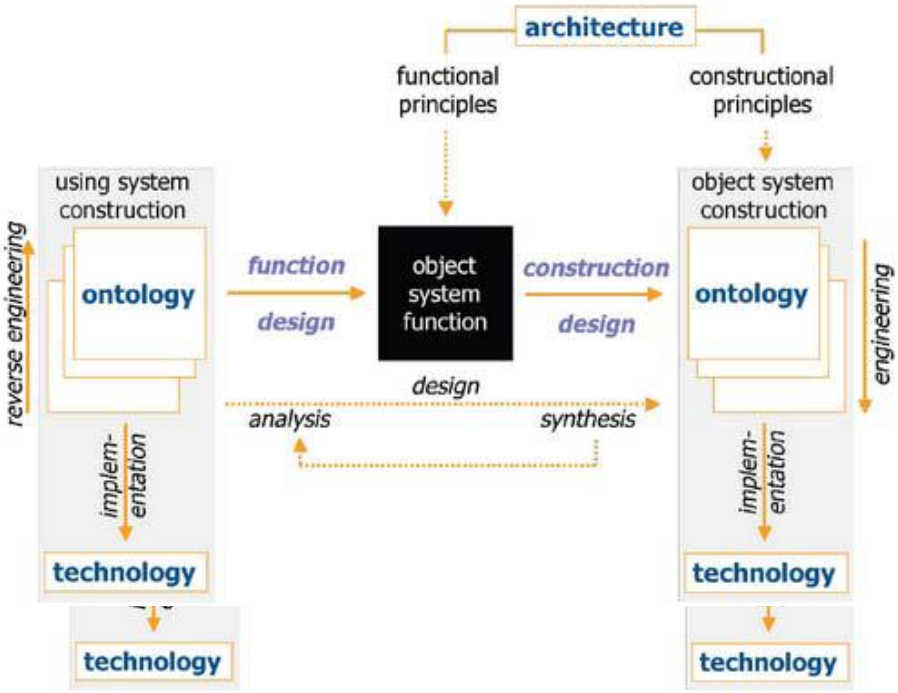

Fig 4. The generic system development process, [1].

4. Architecture and engineering a dynamic Tension [6]

Architecture: The structure of components, their relationships, and the principles and guidelines governing their design and evolution over time.

The architect: he articulates through the design the vision of the operator.

System: A set of interacting components in which the behavior of each component affects the behavior of the whole set.

Systems Engineering: The design, production, and maintenance of trustworthy systems within cost and time constraints.

An interdisciplinary process that ensures that a customer's needs are satisfied throughout a system's entire life-cycle.

The system engineer: it implements a system that conforms to the architecture within cost and time constraints.

5. The rules of architecture

In the development of a house, building or the like we can always identify the following main steps: [9], fig. 5.

- A discovery process to identify the needs and requirements in the context of a certain situation.
- A design process which leads to a design of the object in the form of drawings and/or models.
- A transformation process to plan the realisation of the object in its environment.
- A construction process that regards the realisation of the actual object based on the design and realisation plan.

The principles, guidelines and rules identified in the discovery phase are used in both design, transformation and construction process. As such, the architecture impacts all processes [10].

The architecture constraints the freedom of the designer and constructor of the object and guides them towards a structure that complies with the business vision and concepts of the architecture. The architecture serves as a prescription for the design, transformation and construction of the object. As a result the object will be recognised as being 'designed

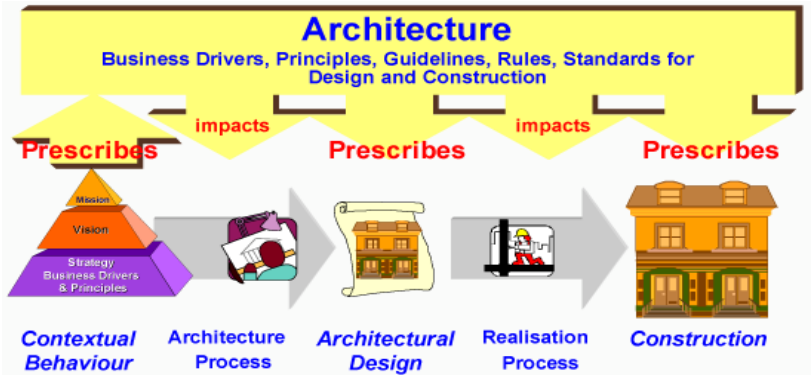

Fig. 5. The architecture impacts all processes, [9].

and constructed under architecture'. The object will inherit the added value of the architecture and will support the (cultural) values, goals and objectives of the organisation. The described role of architecture originates from the building industry. In prescribing the structure, function and style of a building the architecture defines principles, guidelines and rules for [11]:

- The type of components of which the building may be composed;
- How these components must fit together;
- What assemblies of the components are allowed;
- What functions (usage, living, and working) do the components and component assemblies support;
- And how the style represents the values of the owner.

The prescription concerns the overall architecture as well as the design models and the actual construction of the building.

6. Integrated architecture

It has long been recognized that "architecture" has a strong influence over the life cycle of a system. In the past, hardware-related architectural aspects were dominant, whereas softwarerelated architectural integrity, when it existed, was often first to be sacrificed in the course of system development. Today, software-intensive systems are pervasive. The cost of software development and the increasing complexity of software systems have changed the relative balance. Software technology is maturing rapidly. The practice of system development can benefit greatly from adherence to architectural precepts.

However, the concepts of architecture have not been consistently defined and applied within the life cycle of software-intensive systems. Despite significant industrial and research activity in this area, there is no single, accepted framework for codifying architectural thinking, and thereby facilitating the common application and evolution of available and emerging architectural practices.

SO Integrated architectures definition: An architecture consisting of multiple, *Views* : (operational, systems, and technical standards) that facilitate *integration* and promote *interoperability* across family-of-systems (FoS), system-of-systems (SoS) and *compatibility* among related mission area architectures [9].

Integrated architectures provide a logical, structured approach for defining how forces *operate*, the associated *information flow*, the relation between that information flow and *system capabilities,* and the relation between system capabilities and *technical standards*.

7. What is an architecture framework?

How are architectures created? Which things should be taken into account when devising a list of design principles? It is smart to have a structured checklist of issues when doing this. Such a structured checklist is called an Architecture Framework [12]. The term Architecture framework is used very

widely, sometimes it refers to a 'real' framework as just defined, but it also refers a lot of times to some kind of design methodologies. Giving an overview of different definitions of the term 'architecture framework' is not helpful, because most of them are based on another notion of architecture.

One of the most ambiguous words in the initial understanding of Information Architecture is that of the *Framework.* The *Framework* is a generic categorization method for architecture design artifacts. As a support for devising architectures, it is convenient to have a structured checklist of issues that must be paid attention to or that must be taken into account. Such a structured checklist is called an architecture framework.

- *What it consists of*

– Common definitions, products, data, and references.

- *What it does*

– Provides *guidance* on how to *describe* architectures.

– Provides a *generic problem space* and a *common vocabulary* within which individuals can cooperate to solve a specific problem.

– Provides the rules, guidance, and product descriptions for developing and presenting architecture descriptions that ensure a common denominator for *understanding, comparing,* and *integrating* architectures:

– Can be leveraged to provide at least a starter set of issues and concerns that must be addressed in architecture development.

- *What it does not do*

– Provide guidance on how to *design* or *implement* a specific architectur.

– Provide guidance on how to *develop* or *acquire* systems.

Any good categorization method achieves these goals:

1. Simplify information for understanding and communication.
2. Clearly focus on independent components for analytical purposes.
3. Provide and maintain a disciplined method of depicting relationshipsbetween the domains.

A Framework is a set of approaches, components, configurations,models, services, standards and principles that can guide you in your documentation adventure of a particular view of an architecture.

- An object type determines a class of objects for which an AF is meant. Examples: churches, computer networks, organizations.
- Domains are distinctions that are inherent to the object type(s) due to the (possibly unwittingly) applied paradigm. Ideally, domains are independent of each other. Examples: function and construction.
- The general requirements, put in by the stakeholders, are grouped in distinct areas of concern. Examples: security and maintainability.

7.1. Benefit from using a framework to guide design?

A framework does several things:

- it makes it easier to work with complex technologies.
- it ties together a bunch of discrete objects/components into something more useful.
- it forces the team (or just me) to implement code in a way that promotes consistent coding, fewer bugs, and more flexible applications.
- everyone can easily test and debug the code, even code that they did not write.

While there are many benefits in using a framework to guide the development efforts of an architecture, there are two that are moreelementary and prevalent than others: Guidance and Communication. The framework is a guidance tool which is used to ensure that each architecture domain is adequately reviewed and documented, and can most easily be compared in context to the other domains or views. It guides both the method and approach to completing an architecture,and enables all stakeholders to see a consistent and complete picture when reviewing an architecture. The framework can be used as the primary communication vehicle between architects, the project team, stakeholders and IT technical staff. It can be used to show what you've planned, and how you will achieve flexibility in the future. Other benefits that you may expect from using a framework, is the provision of a generic space and structure for a very complex and varying

problem. It can include a starter set of principles, issues and concerns. It can provide guidance on which sets of diagrams and models to include. Think of it as a "placeholders" map to what would make a complete architecture. It is also known as a set of views of an organization and its Business and IT Resources and assets, fig. 6.

7.2. What framework choices are available?

Frameworks are special components that act as base for the structuring and assembly of components in more complex constructions. The use of architecture will not only has consequences for the design and constructions content but will also impact the design and construction process.

The most important consequences are that:

- An architect prepares a design that provides a clear picture of the style, construction and the structure of the resulting business and IT system.
- The organization reviews the design for reflecting the style elements, usability and functionality and the developers assess the feasibility of the design and also the risks and costs of realisation. This may cause changes in the design.
- The organization accepts the final design, which meets their expectations regarding style, usability, functionality and costs of realisation.
- The architect defines the impact of change and realisation in a transformation plan.
- The developers realise the change in conformance to the final design.

The use of components and frameworks both in the design and construction of and Information Technology (IT) system will considerably reduce the effort of designing and building new and IT systems.

There are numerous frameworks available to be used, but they are primarily categorized by domain. They are typically designed for government, or non-government (private organization). Some are strategically oriented and some are focused on organizing the evolution of the architecture. An organization can choose to adopt an existing architecture framework or to build a custom framework. In either case, it is rarely necessary to start from scratch, and the primary focus should be on the stakeholders and the domains you wish to capture. Try to keep in mind that the primary goal here, is 'Communicate that architecture!' If you choose to adopt an architecture, consider visualizing what kind of data and

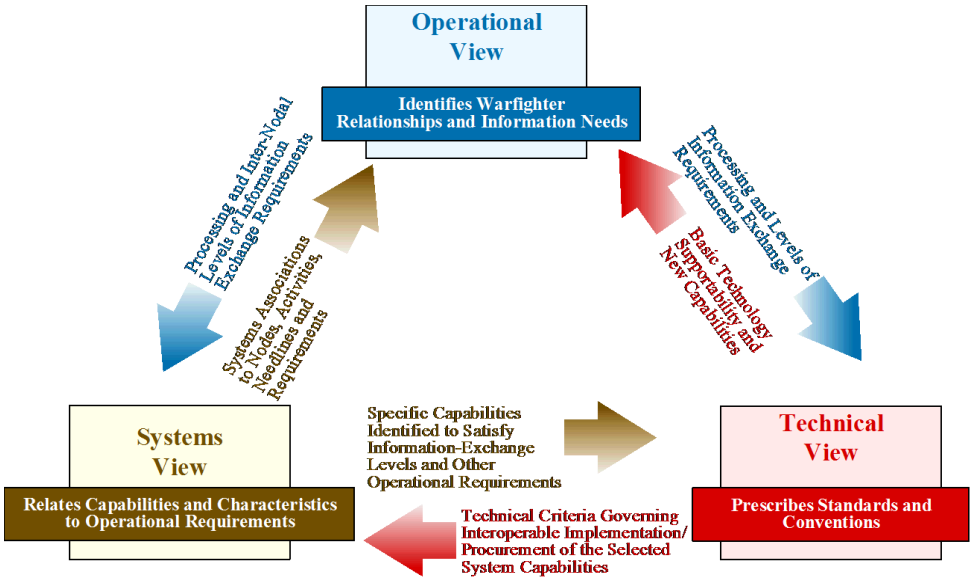

Fig. 6. Linkages among views, the three views of the architectural framework and the interrelationship between architecture views, [6].

models you will use to populate it. Ask yourself 'what is important to my organization?', and 'how have we organized our departments and business in the past?' Many frameworks are used for a particular type of community, so try to pick the one that is most similar to yours. Enquire with peer organizations as to which type of framework they have used. There is value in understanding and using more than one, so pick the one that is the closest, and adapt it by selecting one or more components you would need to close the gaps between the one you select and what you need for your organization. Customize it to suit your culture and technical vocabularies for the ultimate in success [13].

7.3. what follows your framework choice?

After a framework has been selected, evaluate its use within your business environment. Review the goals and objectives in using such a framework, and see if what you picked will work. After treading through a few examples, list the lessons you may have learned and refine what you've selected.

8. The value of architecture

Architecture is capable of delivering value to the business in many ways, through reduced cost and increased value/competitiveness. It is important that architecture clearly delivers value to a business, not be seen as purely an academic exercise [14]. Architecture must be able to demonstrate two key attributes:

- Architectures should be demonstrably linked to the business goals and vision.
- Architectures should provide decision support.
- Architecture should be capable of being applied to different scenarios (Enterprise, Solution, Technical, etc).

Some of the specific benefits delivered from an architectural approach can be readily quantified whilst others have a more subjective value: fig. 7.

- *Reduced project risk and complexity* by reducing project over-runs (both in terms of cost and time) through a better understanding of the solution context and potential problems before the build phase starts [15].
- *Improved project success* demonstrated by the quality of a solution and its fit into a strategic view of the business as well as the delivery on-time and within budget [16].
- *Greater cost control and improved ROI* by ensuring the often departmentally focused projects understand what is available for reuse, and make use of this where appropriate (and understand the long-term costs of not).
- *Reduced costs for business as usual* by better managing operational costs through the consideration of Governance as part of the overall architecture and not, as is often the case, an afterthought.
- *Facilitate delivery of IT strategy and better business/IT alignment:* through the governance model for solution development and portfolio management that an Enterprise Architecture can bring.
- *Increased agility and competitiveness*: with IT becoming an enabler and partner with the business and not just a cost and constraint on it.

9. Integrated Architecture Framework (IAF)

The Integrated Architecture Framework (IAF) is used to structure and define the architecture content. The framework [17]:

- Provides a model for architecture development and usage.
- Describes the format and content of elements of the architecture.
- Specifies the way in which these elements relate to each other.

Fig. 8 shows the basic structure of IAF. The model is broken down into Aspect Areas and Abstraction Levels. Each “cell” in this model has a defined set Artifacts [18]. Views then allow the architects to bring together and visualize the artifacts to help in modeling the architecture and to communicate the architecture with the various stakeholders.

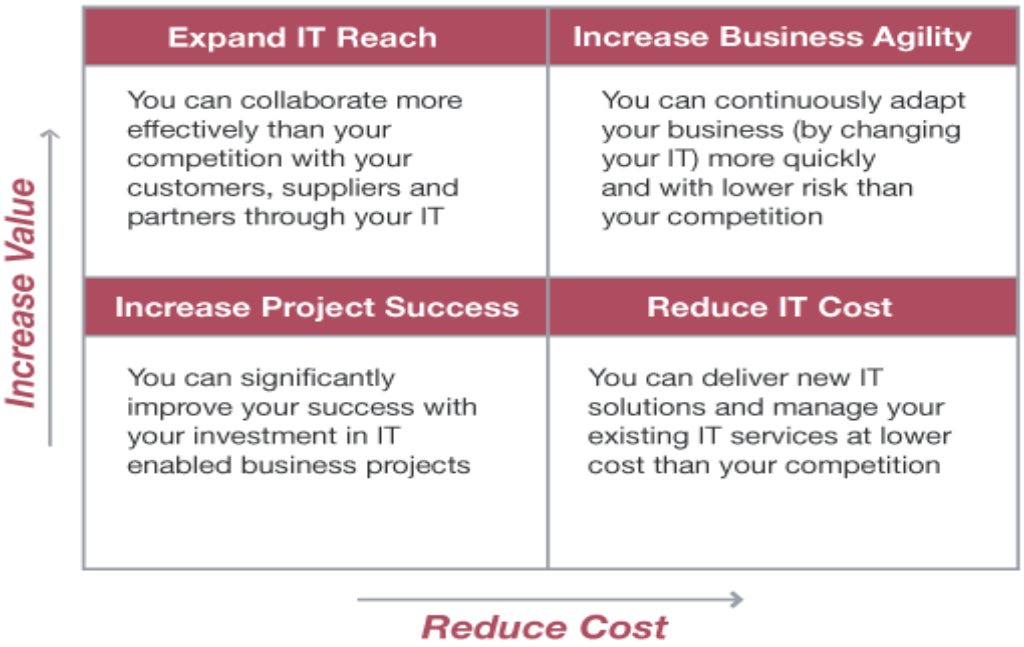

Fig. 7. The value of architecture [17].

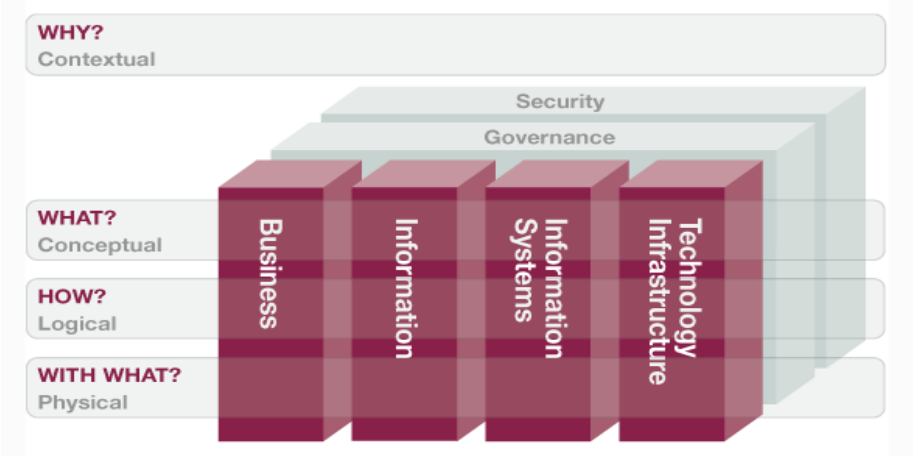

Fig. 8. The integrated architecture framework [17].

Within this framework, IAF describes the architecture using two basic constructs. The first, Artifacts, fundamentally describe the architecture elements. Artifacts belong to, and are derived within, specific areas in the architecture framework. The second, Views, are used for both analyzing and presenting the architecture from different perspectives and for documenting the relationships between Artifacts. Views show why the architecture is what it is, providing both traceability for, and the justification of decisions that have been made in the development of the architecture.
Principles and Characteristics of IAF [17].

In IA framework based on key requirements and fundamental principles:
Value focused: The approach must be focused on delivering value to our clients.
Traceable to business: The architecture and any decisions made as part of the architecture must be clearly justified and traceable back to the needs of the business.

Levels of abstraction: Architecture structures complex problems; the way to address this is by using different levels of abstraction.
Holistic and integrated: The architecture approach must integrate the full scope of business and technology issues.
Different aspects: Architecture must support different business issues and domains of architecture.
Special focus: Developments in business and technology require the capability to provide special attention in specific areas.
*Stable: T*he architecture approach requires a stable content that is easy to hand-over and easy to communicate.
Flexible: The approach must support architecture work in projects and stand-alone architectural services. The approach must not contain a prescriptive methodology, but must provide a stable platform for innovation.
Scalable: The approach must scale from Solution to Enterprise level architecture.
Fast and efficient: The market demands the architecture approach be a fast and efficient as possible.

To achieve these, the Integrated Architecture Framework explicitly separates Process from Content—the content is formally described in the content framework whilst process is incorporated in Engagament Roadmaps, supported by tools and techniques.

10. Conclusions

- The following terms are now distinguished and clearly defined: architecture framework, architecting, architecture, designing, design, ontology, engineering, implementation model, implementing and system. An overview of all these terms is given in fig. 4. This figure is based on fig. 9, [19, 20].
- The presented framework xAF has a well-defined scientific foundation (although not completely presented here). Architectures created within an xAF can be verified and validated.
- Important steps to take are:

- Assessment of the practicality of the xAF by validating current frameworks in it.
- Design of a (formal) language for specifying architectures.

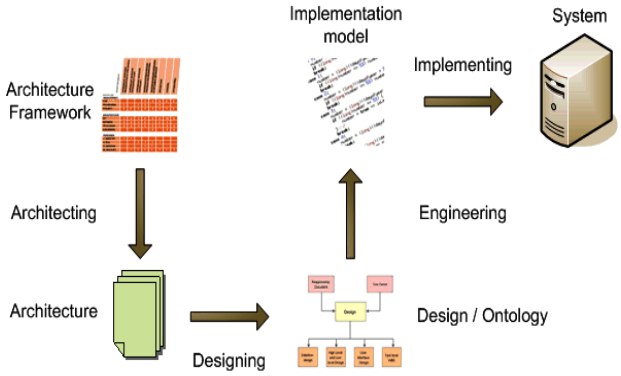

Fig. 9. Main activities in realizing a system, [3] .

- Development of an xAF/MS.

- The architecture framework is required to be language independent and close to industry standards, in particular to the Service Component Architecture (SCA).
- Finally, having formalized the architecture framework we want to spend effort in the verification of components and as a long-term objective in the development of tools for the design and management of component-based systems.
- The IAF, underpinned by the Architects Learning program, the Architects Community, provides the basis on which these architectures are delivered. The Integrated Architecture Framework is successfully deployed in client organizations, as part of their Architecture Capability, and offers these advantages to the client organization.

References

[1] J.D. Haan, "Architecture A Definition", from http://www.theenterprisearchitect.eu/archive/2007/05/15/architecture_a_definition (2007).
[2] J.L.G. Dietz, "Enterprise Ontology - Theory and Methodology", Springer-Verlag Heidelberg, Berlin, New York, pp. 8-13 (2006).
[3] J.D. Haan,"Architecture Definition", http://www.theenterprisearchitect.eu/archive/2007/05/15/architecture-a-definition (2007) .
[4] G. Booch, J. Rumbaugh and I. Jacobson, "The Unified Modeling Language User Guide", Addison-Wesley,

U.S.A. from www.unap.cl/metadot/index.pl?id=25450&isa=Item&field_name=item_attachment_file&op=download_file (1998).

[5] H. William and Others, "An Architecture Framework: From Business Strategies to Implementation", Architecture Services, Canadian Imperial Bank of Commerce, Toronto, Ontario, Canada, From http://jeffsutherland.com/oopsla/herthpub.pdf

[6] J. Watkins, "Relating the Mission and Means Framework to DoD Architecture Framework Products", The University of Texas, USA, from www.dtic.mil/ndia/2004test/mon/tutorial/6perspectives.ppt

[7] J.L.G. Dietz,"System Ontology and its role in Software Development", Proceedings of the Open Interop Workshop on Enterprise Modelling and Ontologies for Interoperability", Vol.160 from http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-160/paper8.pdf (2005).

[8] J.A.P. Hoogervorst, "Enterprise Architecture: Enabling Integration", Agility and Change, Published in: International Journal of Cooperative Information Systems, Vol. 13 (3), pp. 213-233 (2004).

[9] J. Schekkerman, "Architecture Framework Essentials Guide Version 1.5 Extended Enterprise", Institute for Enterprise Architecture Developments, fromhttp://www/enterprise-architecture.info (2006).

[10] J.A. Zachman, "A Framework for Information Systems Architecture", IBM Systems Journal, Vol. 26 (3), pp. 545-470. from http://www.aeablogs.org/eakd/files/Zachman_s_Original_1987_Paper.pdf (1987).

[11] L. Rothrock, "A Theoretical Framework and Quantitative Architecture to Assess Team Task Complexity in Dynamic Environments", Theoretical Issues in Ergonomics Science, Vol. 6 (2), pp.157–171, U.S.A. from http://www.tandf.co.uk/journals (2005).

[12] http://www.codeproject.com/KB/architecture/WhatIsAFramework.aspx

[13] http://www.The Architect Abstract WHAT IS AN ARCHITECTURE FRAMEWORK.htm

[14] N. Jeffry, "A Framework For GlobaL Illumination In Animated Environments", Presented at the Sixth EurographicsWorkshop on Rendering, Dublin, Ireland (from http://cg.cis.upenn.edu/~jnimerof/EGWR6/Framework.html (1995).

[15] G. William, and Others "DoD Architecture Framework and Software Architecture Workshop Report", Carnegie Mellon University, U.S.A.fromhttp://www.ichnet.org/DODAF%20SEI%20report.pdf (2003).

[16] J.L.G. Dietz, "Extensible Architecture Framework (xAF)", version 1.1 (formal edition), from http://www.naf.nl/content/bestanden/xaf-1.1_fe.pdf

[17] http://www.capgemini.com

[18] http://architectbootcamp.blogspot.com/2006/04/what-is-architecture-framework.html

[19] http://www.theenterprisearchitect.eu/archive/2007/06/18/soa-defined-in-a-formal-way

[20] J.S. Dahmann and Others "The DoD High Level Architecture: an Update", the Second International Workshop on Distributed Interactive Simulation and Real-Time Applications from http://www.cc.gatech.edu/computing/pads/PAPERS/High_Level_Architecture_For_Simulation.pdf (1998).

Received March 15, 2009
Accepted May 19, 2009