
CYBERSECURITY AND CRITICAL INFRASTRUCTURE PROTECTION

UDC 004 (72+056.53)

YAROSLAV DOROHYI,
VASYL TSURKAN,
SERHII TELENYK,
OLENA DOROHA-IVANIUK

A COMPARISON ENTERPRISE ARCHITECTURE FRAMEWORKS FOR CRITICAL IT INFRASTRUCTURE DESIGN

This paper investigates the concept of architecture by examining 10 Enterprise Architecture Frameworks (EAF) for critical IT infrastructure (CITI) design such as Zachman's, TOGAF, FEAF, DoDAF, BMDAF, NATOAF, TEAF, GEAF, RM-DOP, SOA. Architecture plays a major role in the development of information systems. The act of architecture design in the development cycle is generally understood to be systematic analysis and design of related information to provide a model for guiding the actual development of information systems. To date, there are more than 100 platforms for the development of architecture, which are divided for use in defense, government, open-source, proprietary. The platform helps to improve the understanding of the topic by providing systematic approaches to architectural design and development, but many aspects of architecture remain ambiguous. The uncertainty concerns the following: architecture (whether the architecture should cover only the software components or include other aspects of the development of critical IT infrastructure), the role of the architect (the role of the architect in the lifecycle of critical IT infrastructure development is often unclear), the results (which should be the result of architectural work – business function documents or a detailed project of critical IT infrastructure), architectural activities (includes design and modeling, but what level of detail is required use and when detailed design starts), architecture testing (how much we need to evaluate, check architecture design results), system requirements (size and complexity, whether systems of different sizes and complexity have the same system requirements for architectural design results), architecture level (which the relationship between the architecture of critical infrastructure enterprise and the stand-alone architecture of the critical IT infrastructure). For the architecture of a complex system such as critical IT infrastructure, there are provided considerations, which consist of several dimensions such as business requirements, technical requirements, criteria, current architecture and future architecture. We propose to analyze AF from different points of view. At first, we analyze AF in terms of their goals, inputs and outcomes. At second, each EAF was analyzed in the terms of Concepts, Modeling, and Process. As a third and fourth point of view, we use some qualitative and quantitative metrics for AF analysis.

Keywords: architecture, architecture framework, enterprise architecture framework, EAF, comparison, critical IT infrastructure.

Introduction. Researchers have offered various definitions and explanations of architecture. In article [1] authors suggested that software architecture is concerned with issues beyond algorithms and data structures of computation. Authors in [2] distinguished architecture from design by suggesting that architecture is concerned with the selection of architectural elements, their interaction and their constraints, but design is concerned with the modularization and detailed interfaces of the design element. Monroe et al. [3] suggested that architecture is not about details of implementation. IEEE's definition of architecture states that it is "the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution" [4]. There was also an attempt to formally distinguish architecture activities from design activities [5].

The Open Group states that Enterprise Architecture is about understanding all of the different elements that do the business and how those factors are related to each other [6]. The US, however, defines EA as a strategic base that integrates strategic goals, business requirements and technology solutions [7]. In [8] author sees EA as a management program as well as a documentation method that together can give an integrated view of an enterprise's strategies, business services, information flows, and resources. Nonetheless, most researchers agree that EA started from Zachman and his originally designed The Zachman Framework for Enterprise Architectures (Zachman framework). This framework shows how the information systems fit into the organisation by asking six questions – what, how, where, who, when and why [9]. EA detects all the components in the organisation and is a facilitator for aligning IT and business goals that need to include business related issues such as organisational goals, business processes, performance. If the organisation lacks such an alignment, then it is more challenging for them to adapt to changes in business strategy [10]. For better understanding of what is EA, the visualisation shown in fig. 1, which consists of 4 main organisational levels, is commonly used [6] - [9], [11]. Business layer presents strategic goals that will drive IT solutions, business roles, business functions, business processes and service flows, business information objects. Data architecture shows data that must be collected, organised and later distributed. Application layer describes people and systems that are in the organisation, applications, software components and enterprise services. Technology layer presents technical infrastructure that is composing the systems, network unit.

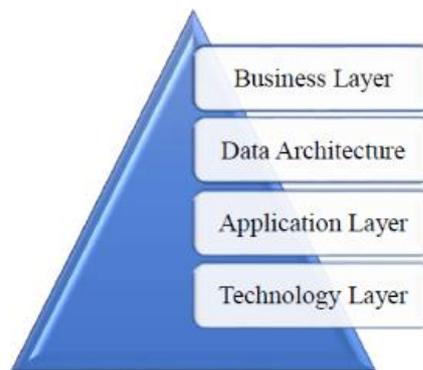


Figure 1 – Layers of Enterprise Architecture [12]

Each of the levels describes either as-is model that already exists or to-be that will exist in the future. As well as this, it is important to remember that all levels are related to each other. For example, the IT system is modelled in the application level as an application that provides services, while in the technology level it is a set of software components that make it possible for services at the application level to work. For creating and future usage of EA, EA frameworks are needed.

Statement of the Problem. Nowadays, there are more than a hundred EA frameworks that are divided into defence industry, government, open-source, proprietary frameworks.

EAFs have helped to improve the understanding of the subject by providing systematic approaches to architecture development, but many aspects of architecture remain ambiguous. The ambiguities are the following:

- the scope of architecture - should the architecture cover only software components or include other aspects of the critical IT infrastructure development?
- the role of the architect - the role of the architect in the life cycle of the CITI development is often unclear.
- results - what should be the result of work with architecture? The results can range from business functions documents to detailed critical IT infrastructures designs.
- architectural activity - architectural activity involves design and modeling, but which level of detail belongs to architecture and when detailed design work starts?
- checking architecture - to what extent should we measure, check, or verify the results of an architecture?

- when to participate - a system which size and complexity require architecture? Can systems of different sizes and complexity require the same architecture results?
- level of architecture. What is the relationship between enterprise architecture and stand-alone architecture of the CITI?

The contribution of this survey is to systematically analyze and compare EAF using the general characteristics of the architecture or elements for using as basis for CITI design. As such, a model of architecture understanding is described, which explains some ambiguity. An analysis of the architecture's structure has revealed some disadvantages, and we offer several ways to overcome them.

Basis for Analysis. For the architecture of a complex system such as critical IT infrastructure, there are provided considerations which consist of several dimensions such as business requirements, technical requirements, criteria, current architecture and future architecture, etc [13]. The dimensions or inputs to the architecture process are interrelated and can not be considered separately in the process. For example, the architecture of a system with flexibility and mobility can have an impact on performance, cost, and graphics. The key result of architecture is the creation of a model for architectural designs. The model considers complex dimensions in order to achieve balanced compromises and minimize the risks of achieving CITI goals. The risks that arise during the construction or development of the system may lie in many areas. They represent uncertainty about achieving the goals of the CITI. Architectural activity eliminates large uncertainties due to the modeling and specification of the CITI until a resolved problem becomes well understood, and the simulated solution has a high degree of confidence in achieving its goals.

In this paper, we propose to analyze AF from different points of view. At first, we analyze AF in terms of their goals, inputs and outcomes [14].

Architecture modeling commonly uses high level abstraction called views. EAFs use viewpoints to create views that represent different perspectives of a critical IT infrastructure model. Common viewpoints are business architecture, information architecture, software architecture and technical architecture. Specific frameworks being studied may have underlying goals to focus on distributed systems, enterprise architecture or industry specific systems. After examining different architecture frameworks and the IEEE Recommended Practice for Architectural Description of Software-Intensive System [4], this paper puts forward a set of common goals for AF. These goals are independent of industry domain, architecture style and system size.

- architecture process – employ a well-defined process to guide the construction of architecture;
- architecture definition and Understanding – make use of standard terms, principles and guidelines for consistent application of the framework for the communication of architecture information to stakeholders;
- architecture analysis – provide a set of viewpoints to guide the collection and analysis of information for making architecture choices;
- architecture evolution support – employ processes and mechanisms that support systems evolution;
- architecture models – provide consistent standards to document architecture specifications for the planning, management, communication and execution of activities related to system development;
- design rationale – document reasons behind design decisions for verification, i.e. “architect for a reason” [15];
- design tradeoffs – select a design from more than one design choices by resolving multi-dimensional conflicting requirements;
- standardization – ensure development and architectural standards are maintained;
- architecture verifiability – provide sufficient information or explanation in the architecture design for review and verification;
- architecture knowledge base – provide consistent representation and repository of design and architecture design rationale.

Inputs represent information that architecture modeling considers. Outcomes represent results and deliverables. Typical inputs to architecture activities are the following.

- technology inputs – strategic architecture direction including technology platforms, future architecture, systems interoperability and emerging technology standards;
- business drivers – business goals, direction, principles, strategies and priorities;
- information system environment – budget, schedule, technical constraints, resources and expertise, organization structure, other constraints, enterprise knowledge base;
- business requirements – users' requirements, functional requirements, data requirements and other business system related requirements;
- non functional requirements – some of these requirements are also referred to as Quality Attributes (QA) or Quality of Services (QoS). These requirements include availability, reliability, scalability, security, performance, inter-operability, modifiability, maintainability, usability and manageability;
- current architecture – current standards and infrastructure.

Using another EAF to develop a CITI will lead to similar but different results, since each structure has different architectural views. On the other hand, the use of the same autofocus for the development of CITIs of varying complexity will require different types of input data and give different results. The results of the EAP reflect the goals that are set to achieve. In order to be neutral, avoid using specific terminology to present the results.

- business model – describes business models, business requirements, business process, system roles, policy statements;
- information model – contains data model, data transformation and data interface;
- system model – models major components of the system. To arrive at a system architecture model, major tradeoffs and design decisions are made. Future system enhancements are also taken into consideration;
- software configuration model – describes how software is packaged, stored, configured, managed and shared;
- computation model – contains system functional description, system process flow, system operations, software components and interactions;
- software processing model – describes how software processes, software threads and run-time environment are structured;
- implementation model – describes physical system structure such as operating environment, hardware components and networking components of the system. Models implementation processes such as installation, deployment, configuration and management;
- platforms – describe platform software such as operating systems, hardware and networking components, protocols and standards;
- transitional design – provides designs and plans to support system transition and evolution;
- non-functional requirements design – models the structure of the system to reflect design of non-functional requirements;
- design rationale – documents reasons of design based on analysis and tradeoffs that involve multiple dimensions of inputs.

At second, we analyze each EAF in the terms of Concepts, Modeling, and Process [16]:

- concepts – EA concepts are importance for enterprises generally and for implementation strategy particularly. According to literature research, a number of considerable EA concepts that are generally addressed, including: definition of EA, alignment between business and IT, importance of repository, the association and communication among artifacts and implementation strategy, governance, EA roles and process are identified [17];
- modeling – since EA concepts provide basis for implementation methodology (IM), thus the modeling for portray designs regarding to those concepts is generally the main part of any IM. A typical modeling comprises of the following major components: notation, syntax and semantics. Modeling different perspectives of enterprise are significant part of modeling that need to utilize in

IM. Consequently, by using an appropriate modeling the IM could reduce the complexities of current and desired architecture, and transition plan effectively [18];

– process – as mentioned above, the modeling is considered as a compulsory part of any IM. However, IM emphasizes the set of process and parts performed as part of the EA life cycle. These activities and steps form the process which guide enterprise architect and business analyzer in EA implementation. A useful IM should cover the following stages, enterprise modeling, current architecture analysis, desired architecture analysis, managing and providing detailed design of projects, describing controlled transition plan, and implementation. IM that covers all parts of the EA development by considering EA concepts is a consistent and complete methodology [17].

Third point of view is concerned with the next five requirements:

– organisational interoperability – deals with the harmonisation of business processes and information technologies, which covers both inter- and intra- organisational boundaries;

– common infrastructure and interoperability – provides an accurate value for exchanging information; the ability of organisations to share information and knowledge within and across organisational boundaries. The underlying foundation for effective interoperability comes from standardised common infrastructure;

– technical interoperability – refers to technological aspects of connecting computer systems to share information or use functionality;

– agility – ability of the organisation to manage changes, which is an essential characteristic for the survival of businesses that are forced to work in dynamic conditions, where changes are permanent;

– reusability – refers to skills that are both business reference models and services. Reusable modules reduce the time and cost of implementation, increase the likelihood of modifications when a change in implementation is required.

And at the end, we chose six requirements for the last evaluation:

– effort required to develop and maintain – the complexity of the modelling tools and methods adopted within the context of EAF – the easier to model and later support the architecture, the better;

– service orientation – applying a performance paradigm about what secures the implementation of sections on operational capabilities for individual tasks.

– evaluation and governance – the framework should allow assessing the effectiveness and maturity of various agents when using EA or the management process to ensure that IT organisations' investments are closely related to business objectives;

– reference models – allow to describe everything using one language;

– documentation – in case the policy and stakeholders are always changing, the documentation on the development of EA is important and needs to be taken into account for the dissemination of knowledge and the exchange of experience;

– cost effectiveness – whether the framework is free or requires some additional money investments.

All this approaches have some intersection points but at whole give us a full view of each EAF shortcomings and benefits.

This paper is not concerned with the format or notation used by EAF. Some EAF is non-specific on representations of its views, but other frameworks prescribe use of formal description languages.

An overview of EA and its frameworks. In this part, the overall analysis of EA and its frameworks is presented. As it was found there are plenty of researchers with description and analysis of EA, a few of them are about implementing EA at various places [19], [20], [21], [22], [23]. No study that focuses on EA in Ukrainian public-sector institutions or Ukrainian companies was found. Process-based management makes organisations more transparent, allowing the development of an effective performance measurement system and improve the cost and resource tracking capability.

As was already mentioned, the frameworks identified for future analysis are the following: The Zachman Framework, The Open Group Architecture Framework, The Federal Enterprise Architecture Framework, The Department Of Defence Architecture Framework, The British Ministry

of Defence Architecture Framework, The NATO Architecture Framework, The Treasury Enterprise Architecture Framework, The Gartner Enterprise Architecture Framework, Open Distributed Processing – Reference Model (RM-ODP) and Service-Oriented Architecture.

The Zachman Framework. The Zachman Framework is usually considered to be the first EAF that was established and proposed by John Zachman [9]. This framework is a structure for helping the management to organise and classify the detailed representation of an enterprise, which represents in a visual way the interaction between the roles in the process. Moreover, it defines owner, designer and builder of the process, as well as setting the component, the way it works, the location where it is situated, the person who is responsible, the team which does the work and why it matters [22]. Zachman Framework, on the one hand, is shown as a planning tool, which can be helpful for enterprises in making better choices, finding the issues in the context of the business and seeing the alternative options and solutions (see fig. 2). On the other hand, it is just a tool for planning and better executing EA development, as it does not focus on the strategy or governance mechanisms. When moving across the table horizontally (for example, from left to right) the different descriptions of the system are shown from the the point of view of the same player. When going through the table vertically (for example, from top to bottom), only one aspect is considered, but the player is changed from the perspective of which this element is considered. Columns give the answers to 6 questions:

- What? – data that needs to be understood and worked with.
- How? – function or how the process of changing the aim of the enterprise into a more detailed description of its operations.
- Where? – network or where the business activities are taking place or will be distributed in the future.
- Who? – people who are involved in the business processes and into implementing the new architecture.
- When? – time and effects of time on the organisation.
- Why? – motivation and formulating the business goals and strategies [9].

Firstly, in Zachman's, each architectural artefact must be only in one cell. The location of a particular artefact must not be undefined, and the architecture is considered complete only if all the cells are filled. Secondly, a cell is considered to be full if it contains artefacts that determine the system for a particular player in a specific aspect. If all cells are filled with objects, this gives enough information to adequately describe the system from each interested stakeholder's point of view and at any possible angle. Thirdly, in Zachman table, the cells in the columns are linked to each other. For example, in the data column of the table, from the point of view of the business owner, the data represents information about the firm, while for the database administrator, the data shows rows and columns in the database. Even though Zachman Framework in most of the cases is considered as EAF, it is rather an ontology rather than a methodology.

The Open Group Architecture Framework (TOGAF). TOGAF is an architectural framework, which in comparison to The Zachman Framework, gives an approach to designing, planning and implementing of EA and is provided by The Open Group free of charge. While TOGAF consists of three main elements: Architecture Development Method (ADM), Enterprise Continuum and Resource Base, ADM is considered as the key component of this framework [23]. Description of TOGAF includes seven parts [24]:

- introduction – contains a high-level description of the key concepts of EA in general and TOGAF in particular;
- ADM – is an essential part of TOGAF, which describes a step-by-step method for developing EA;
- ADM guidelines and techniques – include a detailed description of the rules and techniques that are used in ADM;
- architecture content framework – describes the approach to the description of EA. It contains a metamodel of architectural artefacts, the structure and description of them;
- enterprise continuum & tools – describe the method for categorising and storing the results of core activities in the organisation;

- TOGAF reference models – gives a description of the reference models that can be used in the architectural projects;
- architecture capability framework – an approach to organising the architectural practice in the organisation.



Figure 2 – The Zachman Framework for Enterprise Architecture [9]

ADM (see fig. 3) is a core part of TOGAF and is more than the methodology that uses a step-by-step approach for developing EA. The result of ADM is organisation of concepts, rather than approach for architecting the structure that will help the organisation to solve its problems. The Enterprise Continuum primarily supports the ADM and is a virtual repository where all the information connected to the architecture is stored, as well as the Resource Base – documents, guides and templates. Moreover, ADM process is iterative, cyclic and consists of 8 phases which are shown in Figure 4. Throughout the ADM cycle, the permanent validations of the results against the set expectations have to be done. As to TOGAF, it starts with a preliminary phase, and go all the way from stage A to stage H. Nevertheless, it is possible to return to one of the phases for refinement and more detailed elaboration.

The Federal Enterprise Architecture Framework (FEAF). FEAF was developed by the United States Federal Chief Information Officers Council and is used for promoting shared development for similar US Federal processes, exchange information within governmental and federal agencies [23]. The Federal Government of the United States has more than 300 organisational units of different sizes, scales and means, which include departments, administrations, bureaus, commissions, agencies and councils. These organisations use more than 2.6 million people and spend more than 3.4 trillion dollars per year for the performance of their functions. They often provide services, which are directed to client groups, including civil, industrial, academic, non-profit organisations and the government agencies [7]. FEAF is based on Zachman framework, but refers only to the first three columns there (using slightly different column names) and focuses on the top three rows. FEAF consists of six reference models[7], [8]:

- performance reference model (PRM) – is used for measuring the performance of initial IT investments [26] and estimating how they contribute to identifying opportunities that can be improved;

- business reference model (BRM) – is used for organising, constructing in a hierarchical way and describing day-to-day business operations in the government;
- data reference model (DRM) – describes the interactions and data exchanges between the government and ordinary citizens;
- technical reference model (TRM) – is used for categorising the standards and technologies, supporting and delivering service components;
- infrastructure reference model (IRM) – is used for supporting the hardware that provides functionality;
- security reference model (SRM) – is used as a common language, as well as a methodology, for describing security and privacy regarding business goals in various organisations.

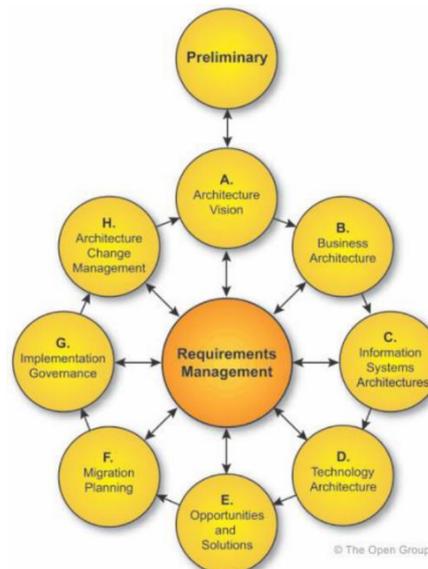


Figure 3 – TOGAF ADM [25]

- A comprehensive description of FEA methodology should include the points, shown in fig. 4:
- the point of view where the architecture of the enterprise will be considered;
 - a set of reference models describing different perspectives on the structure of the organisation (the six models listed above);
 - the process of creating EA;
 - the process of transition from the old paradigm (before the creation of the organisation’s architecture) to the new one (after its inception);
 - taxonomy for classifying assets that fall within the scope of the enterprise’s architecture;
 - the technique, allowing to estimate success of EA use for increasing the business value.

The primary method for modelling FEA is the Collaborative Planning Methodology (CPM) that is a simple, repeatable process that consists of an integrated multidisciplinary analysis, the result of which produces the recommendations developed together with stakeholders, planners and implementers [26]. The CPM is structured in a way that allows to use, reuse and guide planners in determining whether other organisations previously addressed such needs and whether they can use their business models, experiences, and work products. The methodology also helps planners to support management and stakeholders, as they make decisions regarding the directions, which are appropriate for the mission, investment and implementation. Finally, and perhaps most importantly, the methodology provides planners with guidance in their support of measuring the actual performance changes that were the result of the recommendations, and in turn, the use of these results in the future planning activities. The more detailed description of the five steps of the CPM is as follows [26]:

1. Definition and verification – identifying and assessing what needs to be achieved, understanding the primary drivers of change, identifying, approving and prioritising the operational realities of the mission and objectives with management, stakeholders and executive staff.

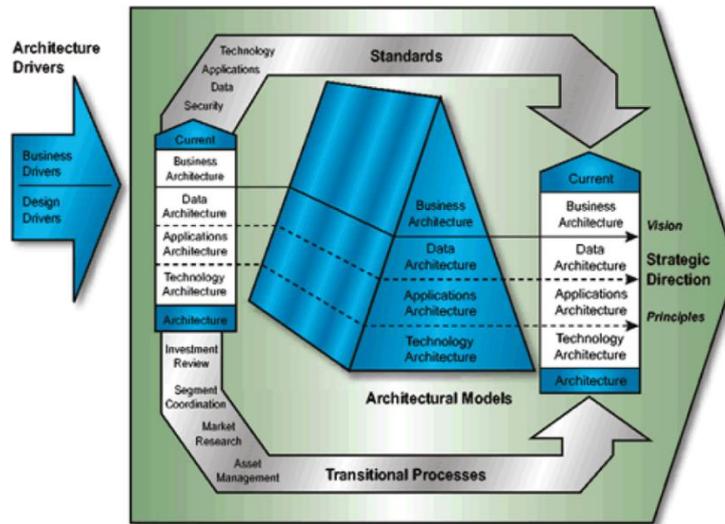


Figure 4 – The whole structure of FEAF [27]

2. Research and use – identifying external organisations and service providers that may have already completed or are currently facing similar needs, analyse their experience and results to determine whether they can be applied.

3. Definition and Planning – developing a plan, which defines what will be done, when it will be done, how much it will cost, how to measure success and what significant risks should be considered to meet the needs identified in Step 1. Also, it includes a timetable that indicates what benefits will be achieved, when they can be expected, and how they will be measured.

4. Invest and Execute – making investment decisions and implementing the changes defined in the integrated plan. At this stage, many groups participate, but it is important to note that these groups will have to work as a coordinated and joint team to achieve the primary goal of this step.

5. Execution and Measurement – managing and measuring the performance of the work by specific indicators.

The Department of Defence Architecture Framework (DoDAF). DoDAF focuses on architectural data, rather than on architectural artefacts, identifying and specifying the information. A model that is displayed as a diagram, narrative text, table, dashboard or other representation is used as a template for organising and displaying data in a format that corresponds to the person, making the decision. DoDAF specification consists of four volumes [28], [29]:

- volume 1 – Introduction, Overview and Concepts – the concepts of the Department of Defence (DoD) architecture are presented, and general recommendations on the development are given. This volume explains the role of the architecture in the first processes of DoD, the key concepts of the structure are defined, which contain overview and vision of DoDAF, structure overview, introduction to the DoDAF meta-model and description of key DoD Viewpoints. Key DoD Viewpoints can be seen in fig. 5;

- volume 2 – Architectural Data and Models – describes DoDAF meta-model, data groups meta-model, perspectives and standard DoDAF models. The DoDAF meta-model defines the types of things that can be modelled and the relationships between these things. Target audience: architects, program managers, system engineers, capability analysts, testers and other users with a technical orientation;

- volume 3 – DoDAF Meta Model Ontology Foundation and Physical Exchange Specification – describes the physical layer format for the exchange of DoDAF compliant architectural data, which helps transferring information between interested parties using different ways. Target audience: developers, analytics;

- volume 4 – DoDAF Journal – is the informative volume of DoDAF. It contains a description of best practices, lessons learned, background documents and other information that complements the three normative of the DoDAF. As well as this, DoDAF describes three main types of architecture that contribute to the DoD architecture [28]:

- *enterprise level reference structure* – provides information about a particular subject area, which directs and limits instances of several architectures and solutions. It consists of 5 elements: strategic purpose – defines the goals and objectives of the architecture; principles – rules, cultures and values that govern technical positions and patterns; technical positions – manuals and standards based on specific principles that should be implemented as a part of the solution; templates – a representation of the generalised architecture, such as viewpoints, graphic and text models, diagrams, etc., which show the relationship between elements and artifacts; vocabulary – terms and definitions that are used in the architecture and are related to the design and solutions;
- *component enterprise architecture* – a description of mission-specific services and capabilities within the component, which displays the relationship between all elements of the DoD;
- *solution architecture* – describes the system or other resources that are used in the organisation to achieve its mission.

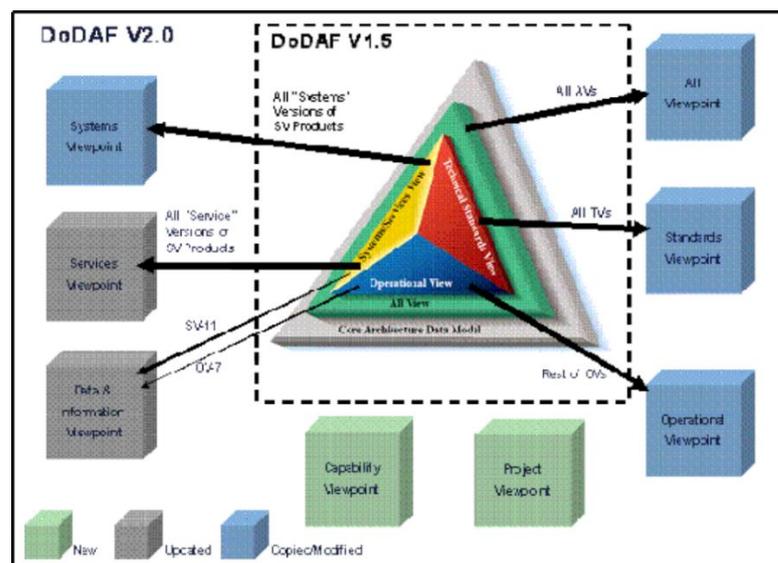


Figure 5 – DoDAF V2.0 Viewpoints [30]

The British Ministry of Defence Architecture Framework (MODAF). MODAF is the architecture framework for describing, analysing and effective managing of defence enterprises, which, to a large extent, is based on DoDAF [31]. MODAF describes enterprises through conceptual models. Complex problems of the business are divided into components, which are described at the highest level in MODAF meta-model, the information presented in the views. Although the meta-model is a generalised model of any enterprise, each of the components must be created with specific standards for a particular organisation. Since the primary architectural data is stored in computer tools or repositories, it is important that data warehouses and modelling tools use common modelling standards so that they can be shared or reused [32]. MODAF maintains compatibility with exact DoDAF views to facilitate the exchange of information with the US, for example, when conducting an international interaction analysis. However, MODAF has supplemented DoDAF with two new points of view that better support defence processes and life cycles. MODAF consists of six templates (called “Views”) that are pictured in fig. 6 [33], [34].

Views are used to query the data model, visualise the architecture components and their dependencies; and to represent real perspectives of the structure of the enterprise:

- *all views (AV)* – provide the summary of the architecture;
- *strategic view (StV)* – defines the goals of the business and the resources that can be used in order to achieve these;
- *operational view (OV)* – presents the activities, functions that are required to conduct business and operational activities;
- *service-oriented view (SOV)* – describes the services, required to support the tasks and activities described in the Operational View;



Figure 6 – MODAF views [35]

- *system view (SV)* – explains what happens when the Operational and Service Oriented Views are implemented and, thereby, define the solution;
- *technical view (TV)* – contains standards, rules, policy and guidance that apply to aspects of the architecture;
- *acquisition view (AcV)* – describes what is needed and how much time it will take for delivering it. Description for some of the MODAF views is provided in tab. 1 [36].

Table 1 – List of some MODAF views [37]

View Category	View Number	View Name	View Description
All Views	AV-2	Integrated Dictionary Defines	Describes the taxonomy elements used by the architecture
Strategic	StV-1	Capability Vision	Outlines the vision for a capability area over a particular time frame
Operational	OV-1a	Operational Concept Graphic	Graphical or textual description of operational concept
System	SV-7	Systems Performance Parameters Matrix	Performance characteristics
Technical	TV-1	Technical Standards Profile	Listing of standards that apply to all the views in a given architecture
Acquisition	AcV-2	System of System Acquisition Programmes	An overview of the complete acquisition programme

MODAF provides a structural model of how essential elements of the organisation are related. However, it does not directly model the resulting behaviour, it only captures and determines some dynamic attributes of the system, for an in-depth evaluation of the required executable models to observe the simulated action. Such a simulation cannot be built without defining an architectural model in the first place.

The NATO Architecture Framework (NAF). Since NATO does not have its forces – the military power depends on the member states, so they need to understand what opportunities can be given to each country to get the optimal military effect. NAF is one of the standards for developing EA, which defines: methodology, viewpoints, stakeholder viewpoints and meta-model [38]. In NAF everything that delivers the result can be seen as a service that plays a fundamental role. The capabilities of each country are modelled as services, and the maximum effect is achieved through the right organisation of these services. For example, the services will be available to the operations planners, to get information about what is happening. Currently, NAF v3.0 is in use, and v4.0 documentation is under development with not much information available. The NAF was delivered from DoDAF, and for now, MODAF and NAF are similar, but not completely aligned. It is expected, that NAF v4.0 will include adapted MODAF Documentation as well NAF v3.x [39]. In NAF v3.0 seven views are available [40]:

- NATO All View (NAV) – captures general aspects related to all seven views, defines the scope and context of the architecture, includes the deadlines, the interrelated conditions, such as techniques, procedures, goals and visions, scenarios, etc., that make up the context for it;
- NATO Capability View (NCV) – fixes the main elements of NATO strategic vision and concepts, explores NATO capabilities, provides detailed information on the dependencies between military capabilities, the possibility of creating more coherent and efficient trade-offs that will be implemented;
- NATO Programme View (NPV) – describes the relationship between the needs for NATO capabilities, various programs and projects, contains program details and conditions for their interaction with NATO operational and financial systems;
- NATO Operational View (NOV) – describes tasks and activities, operational elements and exchange of information that is necessary for the implementation of the NATO mission, determines the types of data exchange, the frequency of it, any activities that support analysis and transfer of the information;
- NATO Technical Systems View (NSV) – a set of graphic and text products describing systems and relationships that provide or accept the functions of NATO. NSV connects system resources with NOV;
- NATO Service-Oriented View (NSOV) – provides a description of the services required to grant an access to the operational area, as described in the NOV and pays particular attention to the identification and description of services;
- NATO Technical View (NTV) – ensures that the system meets a particular set of operational requirements. Also, NTV provides the introduction of technical systems, which is based on technical specifications and include a collection of technical standards, implementation conventions, rules and criteria (see fig. 7).

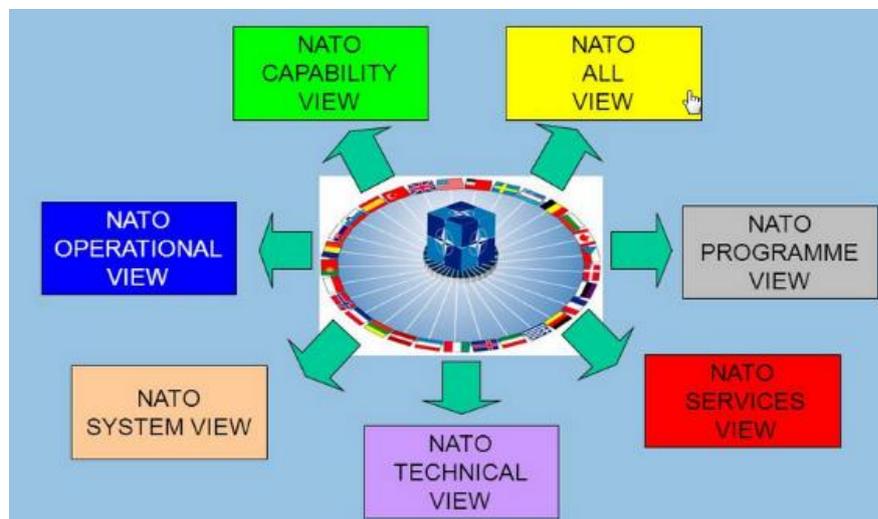


Figure 7 – NATO EAF Views

Each of the views is divided into subviews, which describes the main aim, objects and components to be used, relationships within the particular view to the other subviews [41].

The Treasury Enterprise Architecture Framework (TEAF). The Department of the Treasury published the Treasury Enterprise Architecture Framework (TEAF) in July 2000. The Department of the Treasury is comprised of a number of offices that function as individual enterprises. Therefore, its enterprise architecture needs to map the interrelationships among the organizations in order to manage IT resources. The TEAF aims at facilitating “integration, information sharing, and exploitation of common requirements across the department”(see fig. 8) [42].

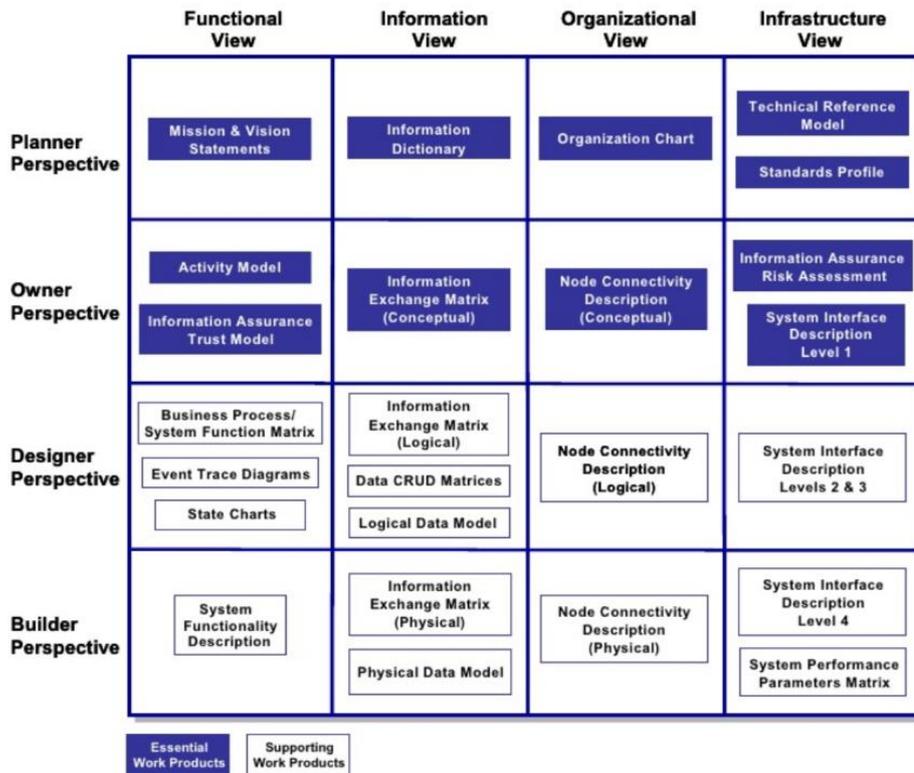


Figure 8 – TEAF Views

Similar to DoDAF, TEAF includes descriptions of work products for documenting and modeling enterprise architectures.

Fig. 8 summarizes the TEAF essential and supporting work products, each mapped to its applicable primary cell of the TEAF Matrix. Many work products integrate information from other views (and sometimes other perspectives) than the view associated with the primary TEAF Matrix cell for the work product. Work products also represent information that spans cells.

TEAF also explicitly states that these work products align with FEAF models and DoDAF products [42].

The Gartner Enterprise Architecture Framework (GEAF). Gartner methodology believes that EA is about bringing together three constituents: business owners, information specialists, and the technology implementers. Bringing given groups together and merge them into the one vision ased on values of business, cause project has succeeded; otherwise project has failed. In Gartner point of view success could be measured by pragmatic term [43] (see fig. 9).

Although the EA Process Model and EA Framework have their own merits and value, they are best used with each other. The Gartner EA Process Model is a valuable complement to any credible, vendor-neutral EA framework. So if an organization has chosen to adopt a different EA framework, the model introduced here will still add significant value to the architecture discipline.

According to Gartner point of view EA project must be started with understanding enterprise direction on business, not with finding its current position. This activity needs to listen to the enterprise strategic plan and understanding how it response to this plan. In order to obtain pure and concise information about enterprise, Gartner tries to achieve them in simple words, without

concerning about recommended standard documents, or technical babbling. The result of this method is providing common understanding about enterprise situation and strategic plan [43].

A framework doesn't answer the question of what to produce when and how it is all related; these are issues addressed by a process model.

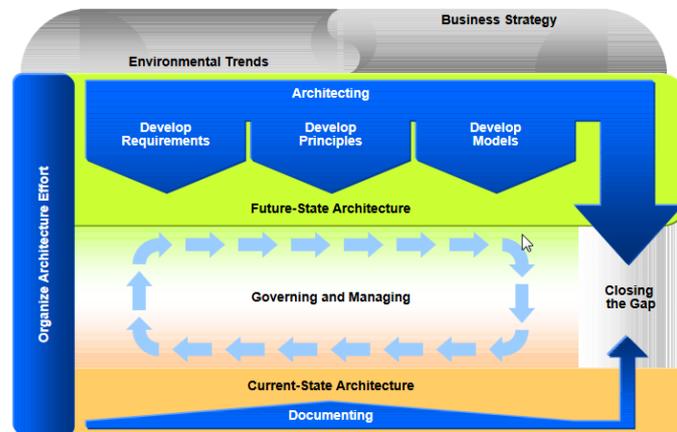


Figure 9. Gartner EA Process Model

Open Distributed Processing – Reference Model (RM-ODP). The ISO RM-ODP Standards [44] is a set of international standards with four parts. Part 1 (ISO 10746-1/ITU-T X.901) provides an overview and a guide to the use of the reference model. Part 2 and Part 3 (ISO 10746-2/ITU-T X.902 and ISO 10746-3/ITU-T X.903) provide a foundation of concepts and prescribe concepts, rules and functions for the modeling of ODP systems. Part 4 (ISO 10746-4/ITU-T X.904) is the architectural semantics which provides a formal description technique for Part 2 and Part 3. The primary objective is to allow the benefits of distribution of information processing services to be realized in an environment of heterogeneous IT resources and multiple organization domains.

RM-ODP uses five viewpoints to represent different aspects of a system. The Enterprise Viewpoint states high level enterprise requirements such as:

- purpose and objectives of systems;
- community or users of system;
- business policies, guidelines, flows and constraints;
- actions performed.

The Information Viewpoint focuses on information semantics and information structures. The Computational Viewpoint focuses on decomposition of the system and on the constraints of the objects and their interactions. The objects specified and modeled can be computational, service support or infrastructure objects. Interactions between objects are connected through interfaces. The Engineering Viewpoint focuses on mechanisms and functions that support interactions between distributed objects. The Technology Viewpoint specifies the choice of technology, including products, standards and technology objects, selected to support the implementation. RM-ODP provides standards to define transparencies for the support of distributed processing. Transparencies are architecture patterns that are defined in the Engineering Viewpoint for hiding transparent functions.

RM-ODP primarily focuses on ODP architecture development. Architecture rationale and tradeoffs are not documented as part of the model. RM-ODP does not provide software configuration model to represent software packaging although Engineering Viewpoint may be used to depict it. It does not concern with business strategies or the evolution of the architecture to meet future needs. RM-ODP is formal and it provides a complete and consistent model for the specification of system architecture design. RM-ODP does not prescribe an architecture process and it is non-specific on what level of details architecture modeling require.

Service-Oriented Architecture (SOA). SOA is the most widely used architecture, with a potential for creating mission-critical, modular, adaptive and enterprise applications [45]. SOA appears as the implementation of a service platform consisting of many services that can be combined

into different solutions and scenarios, as determined by business needs. This capability to integrate and recombine services is what provides the closer relationship between business and IT, as well as flexibility to address new situations. The role of the SOA services platform is to provide a foundation for delivering essential business services in a flexible and easy to compose view. The need for this has led to the creation of SOA, through which composite applications can be set up, modified and removed dynamically using services, abstracted from existing applications and data, presented by the platform or external sources [46].

From a business point of view, SOA can be expressed as a set of flexible services and processes that the organisation wants to expose. In this case, these same services can be recombined and supplemented to support changes in business requirements and models over time. From the technical point of view, SOA defines software regarding discrete services, which are implemented using components that can be called upon for performing a particular business task. Fig. 10 shows a representation of the architecture with the description of each layer below [47].

1. Scope – describes what area of the enterprise is this architecture for.

2. Operational systems layer – consists of existing custom applications, including object-oriented system implementations, as well as applications for business intelligence. A complex, multi-tiered architecture of SOA can use existing systems and integrate them using service-oriented integration methods.

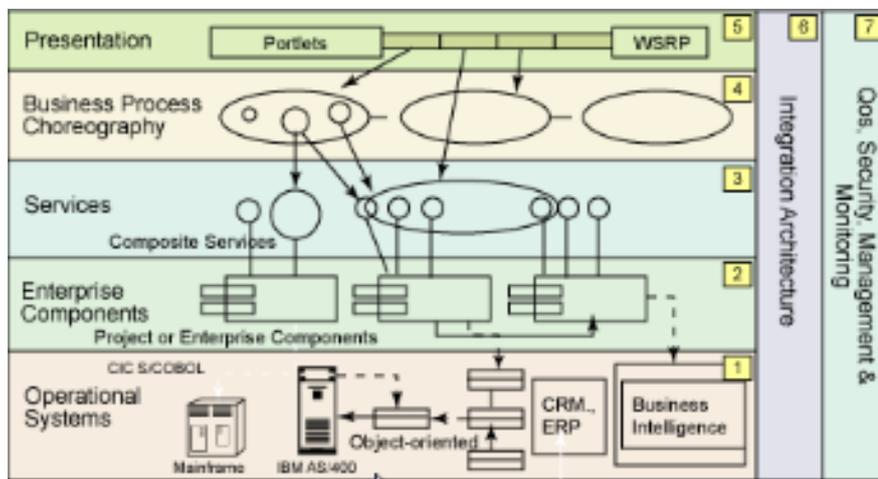


Figure 10. The layers of SOA [48]

3. Enterprise components layer – is responsible for implementing the functionality and maintaining the quality of open services. These unique components are managed, regulated and financed at the organisation or department level. This layer typically uses application servers for component implementation, workload management, high availability and load balancing.

4. Services layer – describes the services that the organisation chooses to finance and exhibit. This level also provides a mechanism for using enterprise-scale components, components specific to business units, and in some cases components for specific projects, and allocates a subset of their interfaces in the form of service descriptions. Thus, the enterprise components provide the implementation of the service at runtime, using the functionality provided by their interfaces.

5. Business process and composition layer – defines the structures of the services exhibited on Layer 3. Services are combined into a stream and thus act together as one application. These applications support specific use cases and business processes.

6. Access or presentation layer – although this level, as a rule, goes beyond the discussions around SOA, it gradually becomes more relevant, it can be thought of as a future layer, which is needed to be considered.

7. Integration layer – allows integrating services by implementing a set of capabilities such as intelligent routing, protocol mediation, and other conversion mechanisms, often described as Enterprise Service Busses (ESB). The Web Services Description Language (WSDL), on the other

hand, specifies a binding that refers to the location where the service is provided. On the contrary, ESB provides a location-independent mechanism for integration.

Evaluation of EAFs. As no such analysis was not done for choosing the best EAF as a basis for critical IT-infrastructure design, the first step for finding the criteria was to check if there are any related works that have done this before. For a general understanding of EAF that has been implemented in the governments all around the world, “enterprise architecture in government” was used in Google Search and 75 first pages were checked. The most widely used EAF turned out to be TOGAF (Switzerland, South Korea, South Africa), FEAF (The United States, Australia, Singapore), Zachman Framework is used in Denmark, while in Finland the Governmental Framework is a combination of TOGAF and FEAF. The popularity of usage of EAF is shown in tab. 2.

Table 2 – Overview of the EAF analysed

Zachman Framework	25%	[22, 9, 11]
TOGAF	11%	[23, 24, 25]
FEAF	9%	[26-27]
DoDAF	11%	[28-30]
MODAF	2%	[31-37]
NAF	1%	[38-41]
TEAF	1%	[42]
GEAF	3%	[43]
RM-ODP	N/A	[44]
SOA	15%	[46-48]

While the literature review was conducted, some of EAF were decided to be excluded from the further comparison. Thus, the following frameworks were chosen for further comparison – ZF, TOGAF, FEAF, DoDAF, MODAF, NAF, TEAF, GEAF, RM-ODP, SOA.

In [49] five frameworks are compared – Zachman, DoDAF, FEAF, TEAF, TOGAF – using three different comparison criteria – by Views/Perspectives, by Abstractions, by Software Development Lifecycle (SDLC) Phases. According to the author [49] Zachman framework appears to be the most comprehensive framework of those studied. It uses some viewpoints related to the different aspects. Most frameworks only represent a small number of viewpoints and aspects. In another article [50] there was no comparison of EAF, but implementation assessment criteria were presented instead, which might be helpful in the future after EA is developed. Article [23] focuses on the evaluation of EAF for e-government focusing on improved interoperability and integration, reduced costs, improved change and risk management, assessment of business-IT alignment.

This study uses 4 various groups of fundamental elements to analyze AF. It shows that all architecture frameworks support the purpose of architecture development. In particular, RM-ODP have a singular focus on software architecture development. TOGAF, DoDAF and FEAF address enterprise architecture issues such as architecture planning, evolution and system interoperability. They use different views for enterprise architecture modeling and have different degrees of specificity in their views. ZF is an enterprise framework but the lack of detailed description of the framework makes it difficult to further analyze its capabilities in this respect. The focus of enterprise frameworks is to facilitate the definition, common understanding and standardization of architecture practice in an enterprise. Their long term goals are to support strategic architecture planning, use an architecture knowledge base to support architecture evolution. The business and architecture models produced from these frameworks describe architecture directions, the “to-be” architectures, and the enterprise strategies to transition from the current architecture to the future architecture.

Although architecture involves design, the objective of architecture differs from detailed design. Design activities are concerned with conceiving and designing in a focused area where architecture is concerned with structure, modeling and planning of the CITI at a higher level. There are no guidelines in any frameworks to distinguish between architecture activities and the extent to which they become detailed design activities. We propose that the guiding principle of the level of details of design in architecture is based on the level of risk. If the risk, or uncertainty, of the

architecture to accurately model the CITI is relatively small, then design could be carried out in the detailed design phase. On the other hand, if the uncertainty or the risk is high, then more architecture activity is required to develop the design to reduce its risk.

Evaluation regarding goal definitions. This section provides a high level comparison and analysis of ten architecture frameworks. Since AFs have different viewpoints or perspectives on how architecture model should be represented, they can be compared only when frameworks are characterized by fundamental elements such as their goals, inputs and outcomes.

Tab. 3 provides an overview and comparisons of frameworks. If a framework explicitly supports an element in the table, it is scored at 2 points . If a framework does not support an element or there is no mention of that element in the documentation, then it is reported as 0 points. Where a framework partially supports or eludes to support an element, it is reported as partial, 1 point. The extent to which each framework supports and interpret an element may differ even when they have the same values in the same row.

Table 3 – Evaluation of the frameworks according to goal definitions

	ZF	TOGAF	FEAF	DoDAF	MODAF	NAF	TEAF	GEAF	RM-ODP	SOA
Goals										
Architecture Definition and Understanding	1	2	2	2	2	2	2	2	2	2
Architecture Process	0	2	2	2	2	2	2	2	0	2
Architecture Evolution Support	0	2	2	2	2	2	1	2	1	2
Architecture Analysis	2	2	2	2	2	2	2	2	2	2
Architecture Models	2	2	2	2	2	2	2	2	2	2
Design Tradeoffs	1	1	1	2	2	2	1	2	1	2
Design Rationale	1	2	1	1	1	1	1	2	2	2
Standardization	0	2	1	2	2	2	1	2	2	2
Architecture Knowledge Base	0	2	2	2	2	2	1	2	2	2
Architecture Verifiability	0	2	0	0	0	1	0	2	1	1
Inputs										
Business Drivers	1	2	2	2	2	2	2	2	1	2
Technology Inputs	0	2	2	2	2	2	1	2	1	2
Business Requirements	2	2	2	2	2	2	2	2	2	2
Information System Environment	1	2	2	2	2	2	2	2	2	2
Current Architecture	1	2	2	2	2	2	2	2	2	2
Non Functional Requirements	1	2	1	1	1	1	0	1	2	2
Outcomes										
Business Model	2	2	2	2	2	2	2	2	2	2
System Model	2	2	2	2	2	2	2	2	2	2
Information Model	2	2	2	2	2	2	2	2	2	2

Table 3. Continuation

Computation Model	2	2	2	2	2	2	2	2	2	2
Software Configuration Model	0	2	0	0	0	0	0	1	1	2
Software Processing Model	2	2	2	2	2	2	2	2	2	2
Implementation Model	1	2	1	2	2	2	2	2	2	2
Platforms	2	2	2	2	2	2	2	2	2	2
Non-functional Requirements Design	1	2	1	1	1	1	1	1	2	2
Transitional Design	0	2	2	2	2	2	1	2	0	2
Design Rationale	0	1	0	1	1	1	0	1	1	1
Total	27	52	42	46	46	47	38	50	42	52

Evaluation regarding conceptual definitions. This section describes a framework for evaluation selected EAFs with a second set of criteria. It comprises a set of criteria that addresses both generic EA attributes and features that are uniquely found in EAF. It covers three major aspects of each EAF: Concepts, Modeling, and Process.

Concepts: TOGAF provides appropriate governance and repository rather than the other by utilizing a specific model for them. Although, TOGAF describes required business and IT architecture in ADM, it more focus on IT development and could not provide appropriate alignment between business and IT. Since FEAF is derived by EAP, almost theirs attributes are same. Although, DODAF is designed for specific domain, it almost considers all EA concepts in acceptable manner. In contrast of other EAFs, Gartner (GEAF) more focus on development process and support adequate EA concepts.

Modeling: utilizing appropriate modeling for both business and IT domains is essential for EAF. GEAF and DODAF/MODAF do not present a method for consistency and traceability. Although, FEAF, NAF, and TOGAF provide appropriate method for modeling, they are different in learning and using. TOGAF provides broad documents about its method and process but access and employing of them need more time rather the others. TOGAF mentioned that EA architects must select needed process for project from TOGAF phases and this is the place that causes difficult using due to its provide complexity on project. Dynamic EA aspect and complexity are the new issue which do not support by all selected EAFs.

Process: TOGAF views EA implementation as continual process, thus it more focus on continuum and repository.

Moreover, TOGAF use requirements process in order to support ADM phases which other EAFs do not use this feature. NAF and FEAF like previous criteria have same condition. DODAF uses required activities in each process attribute in order to support EA implementation in DOD organization, but it does not use requirements process properly. Although Gartner does not consider all concepts attributes efficiently, it considers EA implementation by efficient plan that it comes from their vast experiences.

Notation used:

- 2: high consideration or detailed and clear description;
- 1: medium consideration or little description;
- 0: low consideration or high level description.

To conclude, the following results are achieved based on this research (see tab. 4):

- in concepts: almost most of mentioned EAFs cover all concepts. Strategy and Artifacts are supported by most EAFs; in contrast Alignment and Repository are not utilized in most EAFs;

- in modeling: SOA has the highest grade and TOGAF has fluctuates situation (in some attributes has high grade and in the others has low grade). Moreover, DODAF and Gartner are located in the last respectively. Selected EAF do not have specific plan for depiction complexity and dynamic aspects of EA;
- in process: although, step by step structure, detailed design, and implementation are most usable attributes in EAFs, requirement, maintenance, and continual need to consider more due to lack of consideration in most EAFs.

Table 4 – Evaluation of the frameworks according to conceptual definitions

Criteria	ZF	TOGAF	FEAF	DODAF	MODAF	NAF	TEAF	GEAF	RM-ODP	SOA
Concepts										
Alignment	1	1	0	1	1	1	1	1	1	2
Artifacts	2	2	1	1	1	2	1	1	1	2
Governance	1	2	0	1	1	2	0	1	1	2
Repository	1	1	1	1	1	1	0	1	1	2
Strategy	2	2	2	2	2	2	1	1	1	2
Modeling										
Easy to use	1	0	1	1	1	1	1	1	1	2
Easy to learn	1	0	1	1	1	1	1	1	1	2
Traceability	1	2	1	0	0	1	0	0	0	1
Consistency	1	2	1	0	0	1	0	0	0	1
Different Views	2	1	1	1	1	1	1	0	1	2
Complexity	0	0	0	0	0	0	0	0	0	1
Dynamic	0	0	0	0	0	0	0	0	0	1
Process										
Requirement	2	2	0	0	1	1	1	0	1	2
Step by Step	1	1	1	1	1	1	1	1	1	2
Detailed Design	1	1	1	1	1	1	0	1	1	2
Implementation	1	1	1	1	1	1	0	1	1	2
Guidelines	2	2	2	1	1	2	1	0	0	2
Maintenance	1	1	1	0	1	1	0	0	0	2
Continual	1	2	0	0	0	0	0	0	0	1
Total										
	22	23	14	13	15	20	8	10	12	33

Evaluation regarding qualitative requirements. Tab. 5 presents a comparison of the characteristics of the thitd set given above. These estimates are subjective and were based on the conducted literature review, a personal understanding of EA and needs for applying EA for critical IT-infrastructures design. The following assessment scaling was used: 0 – does not support (the criteria cannot be implemented in EAF), 1 – partially supports (the criteria can be applied to some extent), 2 – fully supports (the criteria can be entirely carried out in EAF).

Table 5 – Evaluation of the frameworks according to qualitative requirements

Criteria	ZF	TOGAF	FEAF	DoDAF	MODAF	NAF	TEAF	GEAF	RM-ODP	SOA
Organisational Interoperability	2	2	2	2	2	2	1	1	1	1
Common infrastructure	1	1	1	1	1	1	1	1	1	1

Table 3. Continuation

and interoperability										
Technical Interoperability	1	1	2	1	1	1	0	1	1	1
Agility	2	2	1	1	1	1	1	1	1	2
Reusability	1	1	2	1	1	0	0	1	0	2
Total	7	7	8	6	6	5	3	5	4	7

Evaluation regarding development requirements. Tab. 6 shows a comparison of the ten frameworks by selected development criteria of the fourth set mentioned above. The same scaling was used for evaluation: 0 – does not support, 1 – partially supports, 2 – fully supports.

Table 6 – Evaluation of the frameworks according to development requirements

Criteria	ZF	TOGAF	FEAF	DoDAF	MODAF	NAF	TEAF	GEAF	RM-ODP	SOA
Effort required to develop and maintain	2	3	2	2	2	2	2	2	1	2
Service Orientation	2	2	2	2	2	2	1	2	1	3
Evaluation and Governance	2	2	3	1	1	1	1	2	1	3
Reference Models	1	2	3	2	2	3	1	2	2	2
Documentation	2	2	3	1	1	1	1	2	2	1
Cost effectiveness	1	3	3	2	2	3	1	2	1	3
Total	10	14	16	10	10	12	7	12	8	14

Evaluation summary. In this part, the results of comparative analysis of the EAF will be presented. Based on the goal, conceptual, qualitative and development requirements comparison, the final overall evaluation of the frameworks is shown in tab. 7, based on which the following two frameworks were chosen for further critical IT-infrastructure analysis and implementation: TOGAF and SOA. Each structure for each criteria was evaluated, and SOA turned out to have the highest score, it might be the best architecture that can be adopted for designing critical IT-infrastructure since it meets most of the criteria. TOGAF has had convergent results as well. TOGAF has been adopted by many governments and has many advantages, especially its mature architectural process, and its integration with the dominant ArchiMate language. Regarding SOA, it is not as widely used as TOGAF but has a plenty of advantages, such as business-focused development, reusability, flexibility, platform independence. However, none of the corporate architectures is complete, they have strengths and weaknesses, and they complement each other.

Table 7 – Overall evaluation of the frameworks

Requirement	ZF	TOGAF	FEAF	DoDAF	MODAF	NAF	TEAF	GEAF	RM-ODP	SOA
Goal	27	52	42	46	46	47	38	50	42	52
Conceptual	22	23	14	13	15	20	8	10	12	33
Qualitative	7	7	8	6	6	5	3	5	4	7
Development	10	14	16	10	10	12	7	12	8	14
Total	66	96	80	75	77	84	56	77	66	106

All architecture frameworks surveyed either omit or have very little description of architecture design rationale even though they are crucial for CITI design. Architectures have to be correctly designed and verified at the early stages of the CITI. We propose to incorporate design rationale in architecture frameworks used for critical IT-infrastructure with the following features:

- cross-reference requirements to architecture design for consistency checking and traceability;
- document tradeoffs rationale based on quantification of criteria set, benefits and risks;
- describe compromises and enhancements made to requirements;
- use scenarios to depict design analysis;
- describe feasibility and infeasibility of the proposed design.

For a given set of inputs there will be more than one possible option for architectural design. Each choice of design in the decision point is associated with a set of criteria, benefits and risks, which are measured both quantitatively and qualitatively. The criteria not only reflect the resources needed to implement the design, but also represent a compromise or alternative criterion in terms of functionality and other factors. Similarly, the advantages are the relative advantages of the design. Risks represent a level of uncertainty for the desired business goals through technical, commercial or other resource reasons. Architecture designs require a set of multi-dimensional inputs l to m as design considerations. Each architecture design option, from l to n , is associated with a set of criteria, benefits and risks.

For critical IT-infrastructure set of criteria consists of [51]:

- reliability – reliability index of critical IT infrastructure during operation;
- survivability – an ability to perform its functions in case of a loss of resources, subsystems;
- security – a measure of the maximum number of processes and services that are served;
- recoverability – the duration of restoration readiness for operation;
- profitability – the cost of various resources for the operation of the critical IT infrastructure;
- safety – a measure of inability to perform unauthorized actions aimed at disrupting the critical it-infrastructure or its parts;
- term of life;
- effectiveness – a combination of all parameters mentioned above in each case for a certain task.

$Arch_i$ is the result of the function $ArchDesign_i$ as shown below (1).

$$ArchDesign_{[1..n]}(I) \rightarrow Arch_{[1..m]}[Criteria, Benefits, Risks] \quad (1)$$

The choice of architectural design from all possible design choices that best suit critical IT infrastructure with a balanced view of minimizing risks and benchmarks with maximum value is always hard solving problem. This is a process of architectural compromise. Currently, some architectural framework expresses compromises by documenting system constraints and assumptions. We believe that the foundations of architecture should be expanded to core documents that include criteria, benefits and risks.

Conclusions. This paper has presented a comparative analysis of architecture frameworks for critical IT infrastructure design. In this regards, we have introduced a model of understanding for EAF based on fundamental elements of architecture. With this model of understanding, EAF could be selected or tailored for critical IT-infrastructure. Selected elements from multiple frameworks could be used in conjunction to meet particular development needs.

To analyze frameworks that have varied viewpoints, we use 4 various types of elements to enable analysis. Based on architecture frameworks' support of these elements, we found two frameworks with distinct characteristics but good enough to be used for CITI design.

We have identified some common deficiencies in the EAFs, especially in the field of streamlining architecture. In this way, we put forward the concept of using criteria, benefits and risks as the basis of compromises. Design justification can be used to combine architectural constructions to provide tracking and verification. None of the surveyed frameworks determines their requirements for the level of detail of architectural design. We believe that this distinction is important in terms of

the life cycle of development for CITI. Therefore, we recommend the use of risk analysis architecture to determine which architectural design required.

Further research. “In future the authors plan to implement several CITI designs based on TOGAF and SOA frameworks and evaluate them using the concept of criteria, benefits and risks as the basis for right CITI design choice.

REFERENCES

- [1] D. Garlan, and M. Shaw, “An Introduction to Software Architecture”, *Advances in Software Engineering and Knowledge Engineering*, vol. 2, pp. 1-39, 1993.
doi: 10.1142/9789812798039_0001.
- [2] D.E. Perry, and A.L. Wolf, “Foundation for the Study of Software Architecture”, *ACM SIGSOFT Software Engineering Notes*, vol. 17, no.4, pp. 40-52, 1992.
- [3] R.T. Monroe, A. Kompanek, R. Melton, and D. Garla, “Architectural Styles, Design Patterns, and Objects”, *IEEE Software*, vol. 14, pp. 43-52, 1997.
- [4] Institute of Electrical and Electronics Engineers. (Sept. 21, 2000). *IEEE Std 1471. IEEE Recommended Practice for Architecture Description of Software-Intensive System*. IEEE Computer Society, 2000.
- [5] A. Eden, and R. Kazman, “Architecture, Design, Implementation”, in *Proc. of the 25th International Conference on Software Engineering*, Washington, 2003, pp. 149-159.
- [6] P. Hinssen, and J. Derynck, *Business/IT Fusion: How to Move Beyond Alignment and Transform IT in Your Organization*, Gent, Belgium: Mach Media, 2009.
- [7] US Office of Management and Budget. (May 2, 2012). *The Common Approach to Federal Enterprise Architecture*, US Office of Management and Budget, 2012.
- [8] J. Schekkerman, *How to Survive in the Jungle of Enterprise Architecture Frameworks Creating Or Choosing an Enterprise Architecture Framework*, Bloomington, USA: Trafford Publishing, 2004.
- [9] J.A. Zachman, “John Zachman’S Concise Definition of the Zachman Framework™”. [Online]. Available: <https://www.zachman.com/resources>. Accessed on: Aug 21, 2017.
- [10] J.W. Ross, “Creating a Strategic IT Architecture Competency: Learning in Stages”, *MIS Quarterly Executives*, vol. 2, no. 1, pp.31-43, 2003.
- [11] J.A. Zachman, “The Zachman Framework for Enterprise Architecture”. [Online]. Available: https://www.zachman.com/images/ZI_Plcs/ZF3.0.jpg. Accessed on: Aug 21, 2017.
- [12] “Layered Enterprise Architecture | LEADing Practice”. [Online]. Available: <http://www.leadingpractice.com/enterprise-standards/enterprise-architecture/layered-enterprise-architecture/>. Accessed on: Aug 21, 2017.
- [13] J. Han, and P. Chen, “Architecture Support for System-of-Systems Evolution”, in *Proc. of the First International Conference on Engineering and Deployment of Cooperative Information Systems*, Berlin, 2002, pp. 332-346.
- [14] A. Tang, J. Han, and P. Chen, “A Comparative Analysis of Architecture Frameworks”, in *Proc. of 11th Asia-Pacific Software Engineering Conference*, Busan, 2004, pp. 640-647.
doi: 10.1109/APSEC.2004.2.
- [15] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, Boston, USA: Addison Wesley, 2003.
- [16] R. Lagerström, F. Ulrik, J. Pontus, and J. Ullberg, “A method for creating enterprise architecture metamodels—applied to systems modifiability analysis”, *International Journal of Computer Science and Applications*, vol. 6, no. 5, pp. 89-120, 2009.
- [17] R. Winter, and R. Fischer, “Essential layers, artifacts, and dependencies of enterprise architecture”, *Journal of Enterprise Architecture*, vol. 3, no. 2, pp. 7-18, 2007.

- [18] J. Sembiring, E. Nuryatno, and Y. Gondokaryo, "Analyzing the Indicators and Requirements in Main Components of Enterprise Architecture Methodology Development using Grounded Theory in Qualitative Methods", in *Proc. of Interdisciplinary Business Research Conference*, Bangkok, 2011, pp. 9.
doi: 10.2139/ssrn.1867875.
- [19] A. Ramadhan, and A. Arman, "Enterprise Architecture in University analyzing of implementation using business process management", in *Proc. of International Conference on ICT for Smart Society*, Bandung, 2014, pp. 250-255.
doi: 10.1109/ICTSS.2014.7013182.
- [20] F. Hald, *Enterprise Architecture in Denmark*, Copenhagen, Denmark: Copenhagen Business School, 2006.
- [21] R. Ravenhorst, *Developing a project architecture and evaluating the Enterprise Architecture initiative at Tata Steel*, Delft, Netherlands: Electrical Engineering, Mathematics and Computer Science, 2012.
- [22] C.L. Parmo, *The use of Enterprise Architecture. IT Strategy and IT Governance at StatoilHydro*, Trondheim, Norwegian: Norwegian University of Science and Technology, 2009.
- [23] M.A. Mohamed, G.H. Galal-edeen, H.A. Hassan, and E.E. Hasanien, "An evaluation of enterprise architecture frameworks for e-government", in *Proc. 7th Seventh International Conference on Computer Engineering & Systems*, Egypt, 2012, pp. 255-260.
doi: 10.1109/ICCES.2012.6408524.
- [24] "TOGAF 9.1". [Online]. Available: <http://pubs.opengroup.org/architecture/togaf9-doc/arch/>. Accessed on: Aug 21, 2017.
- [25] "TOGAF Architecture Development Method (ADM)". [Online]. Available: <http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/adm.png>. Accessed on: Aug 21, 2017.
- [26] "Basics of the Federal Enterprise Architecture Framework". [Online]. Available: <http://innovategov.org/2015/04/08/basics-of-the-federal-enterprise-architecture-framework/>. Accessed on: Aug 21, 2017.
- [27] "The whole structure of FEAF". [Online]. Available: <http://pubs.opengroup.org/architecture/togaf8-doc/arch/Figures/feaf.gif>. Accessed on: Aug 21, 2017.
- [28] Department of Defense. (Apr. 23, 2007). *DoD Architecture Framework. Volume I: Overview and Concepts Manager's Guide*. Department of Defense, 2015.
- [29] Department of Defense. (Sept. 30, 2010). *The DoDAF Architecture Framework Version 2.02*. [Online]. Available: http://dodcio.defense.gov/Portals/0/Documents/DODAF/DoDAF_v2-02_web.pdf. Accessed on: Aug 21, 2017.
- [30] Department of Defense. (May 10, 2010). *DoDAF V2.0 Viewpoints*. [Online]. Available: http://dodcio.defense.gov/portals/0/Images/dodaf20/clip_image002_0011.gif. Accessed on: Aug 21, 2017.
- [31] "MODAF (UK Ministry of Defence Architectural Framework) Training – Tonex Training". [Online]. Available: <https://www.tonex.com/training-courses/modaf-training/>. Accessed on: Aug 21, 2017.
- [32] "The Human View The Human View Handbook for MODAF: Part II – Technical Description". [Online]. Available: <https://www.defencehumancapability.com/DOCS%20HFI%20DTC%20Legacy%20%20HFI%20Process%20Phase%203/MoDAF%20Part%202.pdf>. Accessed on: Aug 21, 2017.
- [33] "IBM Knowledge Center – MODAF overview". [Online]. Available: https://www.ibm.com/support/knowledgecenter/SS6RBX_11.4.3/com.ibm.sa.modaf.doc/topics/c_MoDAF_Overview.html. Accessed on: Aug 21, 2017.

- [34] “MODAF – Ministry of Defence Architecture Framework”. [Online]. Available: <https://www.slideshare.net/AaronVarrone/modaf>. Accessed on: Aug 21, 2017.
- [35] “The British Ministry of Defence Architecture Framework Views”. [Online]. Available: http://incoseonline.org.uk/Images/zGuides/z8_framework.jpg. Accessed on: Aug 21, 2017.
- [36] “MOD Architectural Framework Technical Handbook”. [Online]. Available: <http://www.modaf.com/files/MODAF%20Technical%20Handbook%20V1.0.pdf>. Accessed on: Aug 21, 2017.
- [37] “TOGAF TO MODAF MAPPING”. [Online]. Available: <http://docplayer.net/20752419-Togaf-to-modaf-mapping.html>. Accessed on: Aug 21, 2017.
- [38] “NATO Architecture Framework v4.0 Documentation (draft)”. [Online]. Available: <http://nafdocs.org/>. Accessed on: Aug 21, 2017.
- [39] P. Gorman, K. van Haperen, and I. Bailey, “An Update on the Convergence of MOD and NATO Architecture Frameworks”. [Online]. Available: <http://nafdocs.org/wp-content/uploads/2013/07/Gorman-Bailey-Van-Haperen-IEA14-v2.pdf>. Accessed on: Aug 21, 2017.
- [40] “Nato Architecture Framework (NAF) – 3 – Views”. [Online]. Available: [https://training-course-material.com/training/Nato_Architecture_Framework_\(NAF\)_-_3_-_Views](https://training-course-material.com/training/Nato_Architecture_Framework_(NAF)_-_3_-_Views). Accessed on: Aug 21, 2017.
- [41] D. Ota, and M. Gerz, “Benefits and Challenges of Architecture Frameworks”. [Online]. Available: http://www.dodccrp.org/events/16th_iccrts_2011/presentations/024.pdf. Accessed on: Aug 21, 2017.
- [42] “Treasurey Enterprise Architecture Framework (TEAF)”. [Online]. Available: <https://www.slideshare.net/Aamir97/treasurey-enterprise-architecture-framework-teaf>. Accessed on: Aug 21, 2017.
- [43] J. Greta, R. Handler, A. Lapkin, and N. Gall, “Gartner Enterprise Architecture Framework: Evolution 2005”. [Online]. Available: <https://www.gartner.com/doc/486565/gartner-enterprise-architecture-framework-evolution>. Accessed on: Aug 21, 2017.
- [44] Information Standards Organisation. (Dec. 15, 1998). *ISO/ITU-T 10746 Part 1-4. Reference Model for Open Distributed Processing*, Information Standards Organisation, 1998.
- [45] P.R. Chelliah, “Elucidating the Cloud Enterprise Architecture for Smarter Enterprises”, *IT Professional*, vol. 16, no. 6, p. 33-37, 2014.
doi: 10.1109/MITP.2014.90.
- [46] L.-J. Zhang, “Service-Oriented Architecture”, in *Services Computing*, J. Zhang and H. Cai, Eds. Berlin, Germany: Springer Publisher, 2007, p. 89-113.
- [47] A. Arsanjani, “Service-oriented modeling and architecture. How to identify, specify, and realize services for your SOA”, [Online]. Available: <https://www.ibm.com/developerworks/library/ws-soa-design1>. Accessed on: Aug 21, 2017.
- [48] “The Layers of Service Oriented Architecture”. [Online]. Available: <https://www.ibm.com/developerworks/library/ws-soa-design1/figure3.gif>. Accessed on: Aug 21, 2017.
- [49] L. Urbaczewski, and S. Mrdalj, “A comparison of enterprise architecture frameworks”, *Issues in Information Systems*, vol. 7, no. 2, pp. 18-23, 2006.
- [50] N. Bakar, S. Harihodin, and N. Kama, “Assessment of Enterprise Architecture Implementation Capability and Priority in Public Sector Agency”, *Procedia Computer Science*, vol. 100, p. 198-206, 2016.
doi: 10.1016/j.procs.2016.09.141.
- [51] Y.Y. Dorogyy, “Life cycle of critical IT-infrastructure”, *ElectronComm*, vol. 20, no. 4 (87), pp. 101-105, 2015.

The article was received 03.09.2017

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] D. Garlan, and M. Shaw, “An Introduction to Software Architecture”, *Advances in Software Engineering and Knowledge Engineering*, vol. 2, pp. 1-39, 1993.
doi: 10.1142/9789812798039_0001.
- [2] D.E. Perry, and A.L. Wolf, “Foundation for the Study of Software Architecture”, *ACM SIGSOFT Software Engineering Notes*, vol. 17, no.4, pp. 40-52, 1992.
- [3] R.T. Monroe, A. Kompanek, R. Melton, and D. Garla, “Architectural Styles, Design Patterns, and Objects”, *IEEE Software*, vol. 14, pp. 43-52, 1997.
- [4] Institute of Electrical and Electronics Engineers. (Sept. 21, 2000). *IEEE Std 1471. IEEE Recommended Practice for Architecture Description of Software-Intensive System*. IEEE Computer Society, 2000.
- [5] A. Eden, and R. Kazman, “Architecture, Design, Implementation”, in *Proc. of the 25th International Conference on Software Engineering*, Washington, 2003, pp. 149-159.
- [6] P. Hinssen, and J. Derynck, *Business/IT Fusion: How to Move Beyond Alignment and Transform IT in Your Organization*, Gent, Belgium: Mach Media, 2009.
- [7] US Office of Management and Budget. (May 2, 2012). *The Common Approach to Federal Enterprise Architecture*, US Office of Management and Budget, 2012.
- [8] J. Schekkerman, *How to Survive in the Jungle of Enterprise Architecture Frameworks Creating Or Choosing an Enterprise Architecture Framework*, Bloomington, USA: Trafford Publishing, 2004.
- [9] J.A. Zachman, “John Zachman’s Concise Definition of the Zachman Framework™”. [Online]. Available: <https://www.zachman.com/resources>. Accessed on: Aug 21, 2017.
- [10] J.W. Ross, “Creating a Strategic IT Architecture Competency: Learning in Stages”, *MIS Quarterly Executives*, vol. 2, no. 1, pp.31-43, 2003.
- [11] J.A. Zachman, “The Zachman Framework for Enterprise Architecture”. [Online]. Available: https://www.zachman.com/images/ZI_PICs/ZF3.0.jpg. Accessed on: Aug 21, 2017.
- [12] “Layered Enterprise Architecture | LEADing Practice”. [Online]. Available: <http://www.leadingpractice.com/enterprise-standards/enterprise-architecture/layered-enterprise-architecture/>. Accessed on: Aug 21, 2017.
- [13] J. Han, and P. Chen, “Architecture Support for System-of-Systems Evolution”, in *Proc. of the First International Conference on Engineering and Deployment of Cooperative Information Systems*, Berlin, 2002, pp. 332-346.
- [14] A. Tang, J. Han, and P. Chen, “A Comparative Analysis of Architecture Frameworks” in *Proc. of 11th Asia-Pacific Software Engineering Conference*, Busan, 2004, pp. 640-647.
doi: 10.1109/APSEC.2004.2.
- [15] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, Boston, USA: Addison Wesley, 2003.
- [16] R. Lagerström, F. Ulrik, J. Pontus, and J. Ullberg, “A method for creating enterprise architecture metamodels—applied to systems modifiability analysis”, *International Journal of Computer Science and Applications*, vol. 6, no. 5, pp. 89-120, 2009.
- [17] R. Winter, and R. Fischer, “Essential layers, artifacts, and dependencies of enterprise architecture”, *Journal of Enterprise Architecture*, vol. 3, no. 2, pp. 7-18, 2007.
- [18] J. Sembiring, E. Nuryatno, and Y. Gondokaryo, “Analyzing the Indicators and Requirements in Main Components of Enterprise Architecture Methodology Development using Grounded Theory in Qualitative Methods”, in *Proc. of Interdisciplinary Business Research Conference*, Bangkok, 2011, pp. 9.
doi: 10.2139/ssrn.1867875.

- [19] A. Ramadhan, and A. Arman, "Enterprise Architecture in University analyzing of implementation using business process management", in *Proc. of International Conference on ICT for Smart Society*, Bandung, 2014, pp. 250-255.
doi: 10.1109/ICTSS.2014.7013182.
- [20] F. Hald, *Enterprise Architecture in Denmark*, Copenhagen, Denmark: Copenhagen Business School, 2006.
- [21] R. Ravenhorst, *Developing a project architecture and evaluating the Enterprise Architecture initiative at Tata Steel*, Delft, Netherlands: Electrical Engineering, Mathematics and Computer Science, 2012.
- [22] C.L. Parmo, *The use of Enterprise Architecture. IT Strategy and IT Governance at StatoilHydro*, Trondheim, Norwegian: Norwegian University of Science and Technology, 2009.
- [23] M.A. Mohamed, G.H. Galal-edeen, H.A. Hassan, and E.E. Hasanien, "An evaluation of enterprise architecture frameworks for e-government", in *Proc. 7th Seventh International Conference on Computer Engineering & Systems*, Egypt, 2012, pp. 255-260.
doi: 10.1109/ICCES.2012.6408524.
- [24] "TOGAF 9.1". [Online]. Available: <http://pubs.opengroup.org/architecture/togaf9-doc/arch/>. Accessed on: Aug 21, 2017.
- [25] "TOGAF Architecture Development Method (ADM)". [Online]. Available: <http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/adm.png>. Accessed on: Aug 21, 2017.
- [26] "Basics of the Federal Enterprise Architecture Framework". [Online]. Available: <http://innovategov.org/2015/04/08>. Accessed on: Aug 21, 2017.
- [27] "The whole structure of FEAF". [Online]. Available: <http://pubs.opengroup.org/architecture/togaf8-doc/arch/Figures/feaf.gif>. Accessed on: Aug 21, 2017.
- [28] Department of Defense. (Apr. 23, 2007). *DoD Architecture Framework. Volume I: Overview and Concepts Manager's Guide*. Department of Defense, 2015.
- [29] Department of Defense. (Sept. 30, 2010). *The DoDAF Architecture Framework Version 2.02*. [Online]. Available: http://dodcio.defense.gov/Portals/0/Documents/DODAF/DoDAF_v2-02_web.pdf. Accessed on: Aug 21, 2017.
- [30] Department of Defense. (May 10, 2010). *DoDAF V2.0 Viewpoints*. [Online]. Available: <http://dodcio.defense.gov/portals/0/Images/dodaf20>. Accessed on: Aug 21, 2017.
- [31] "MODAF (UK Ministry of Defence Architectural Framework) Training – Tonex Training". [Online]. Available: <https://www.tonex.com/training-courses/modaf-training/>. Accessed on: Aug 21, 2017.
- [32] "The Human View The Human View Handbook for MODAF: Part II – Technical Description". [Online]. Available: <https://www.defencehumancapability.com/DOCS%20HFI%20DTC%20Legacy%20%20HFI%20Process%20Phase%203/ModAF%20Part%202.pdf>. Accessed on: Aug 21, 2017.
- [33] "IBM Knowledge Center – MODAF overview". [Online]. Available: https://www.ibm.com/support/knowledgecenter/SS6RBX_11.4.3/com.ibm.sa.modaf.doc/topics/c_MoDAF_Overview.html. Accessed on: Aug 21, 2017.
- [34] "MODAF – Ministry of Defence Architecture Framework". [Online]. Available: <https://www.slideshare.net/AaronVarrone/modaf>. Accessed on: Aug 21, 2017.
- [35] "The British Ministry of Defence Architecture Framework Views". [Online]. Available: http://incoseonline.org.uk/Images/zGuides/z8_framework.jpg. Accessed on: Aug 21, 2017.
- [36] "MOD Architectural Framework Technical Handbook". [Online]. Available: <http://www.modaf.com/files/MODAF%20Technical%20Handbook%20V1.0.pdf>. Accessed on: Aug 21, 2017.

- [37] “TOGAF TO MODAF MAPPING”. [Online]. Available: <http://docplayer.net/20752419-Togaf-to-modaf-mapping.html>. Accessed on: Aug 21, 2017.
- [38] “NATO Architecture Framework v4.0 Documentation (draft)”. [Online]. Available: <http://nafdocs.org/>. Accessed on: Aug 21, 2017.
- [39] P. Gorman, K. van Haperen, and I. Bailey, “An Update on the Convergence of MOD and NATO Architecture Frameworks”. [Online]. Available: <http://nafdocs.org/wp-content/uploads/2013/07/Gorman-Bailey-Van-Haperen-IEA14-v2.pdf>. Accessed on: Aug 21, 2017.
- [40] “Nato Architecture Framework (NAF) – 3 – Views”. [Online]. Available: [https://training-course-material.com/training/Nato_Architecture_Framework_\(NAF\)_-_3_-_Views](https://training-course-material.com/training/Nato_Architecture_Framework_(NAF)_-_3_-_Views). Accessed on: Aug 21, 2017.
- [41] D. Ota, and M. Gerz, “Benefits and Challenges of Architecture Frameworks”. [Online]. Available: http://www.dodccrp.org/events/16th_iccrts_2011/presentations/024.pdf. Accessed on: Aug 21, 2017.
- [42] “Treasury Enterprise Architecture Framework (TEAF)”. [Online]. Available: <https://www.slideshare.net/Aamir97/treasury-enterprise-architecture-framework-teaf>. Accessed on: Aug 21, 2017.
- [43] J. Greta, R. Handler, A. Lapkin, and N. Gall, “Gartner Enterprise Architecture Framework: Evolution 2005”. [Online]. Available: <https://www.gartner.com/doc/486565/gartner-enterprise-architecture-framework-evolution>. Accessed on: Aug 21, 2017.
- [44] Information Standards Organisation. (Dec. 15, 1998). *ISO/ITU-T 10746 Part 1-4. Reference Model for Open Distributed Processing*, Information Standards Organisation, 1998.
- [45] P.R. Chelliah, “Elucidating the Cloud Enterprise Architecture for Smarter Enterprises”, *IT Professional*, vol. 16, no. 6, p. 33-37, 2014.
doi: 10.1109/MITP.2014.90.
- [46] L.-J. Zhang, “Service-Oriented Architecture”, in *Services Computing*, J. Zhang and H. Cai, Eds. Berlin, Germany: Springer Publisher, 2007, p. 89-113.
- [47] A. Arsanjani, “Service-oriented modeling and architecture. How to identify, specify, and realize services for your SOA”, [Online]. Available: <https://www.ibm.com/developerworks/library/ws-soa-design1>. Accessed on: Aug 21, 2017.
- [48] “The Layers of Service Oriented Architecture”. [Online]. Available: <https://www.ibm.com/developerworks/library/ws-soa-design1/figure3.gif>. Accessed on: Aug 21, 2017.
- [49] L. Urbaczewski, and S. Mrdalj, “A comparison of enterprise architecture frameworks”, *Issues in Information Systems*, vol. 7, no. 2, pp. 18-23, 2006.
- [50] N. Bakar, S. Harihodin, and N. Kama, “Assessment of Enterprise Architecture Implementation Capability and Priority in Public Sector Agency”, *Procedia Computer Science*, vol. 100, p. 198-206, 2016.
doi: 10.1016/j.procs.2016.09.141.
- [51] Y.Y. Dorogyu, “Life cycle of critical IT-infrastructure”, *ElectronComm*, vol. 20, no. 4 (87), pp. 101-105, 2015.

ЯРОСЛАВ ДОРОГИЙ,
ВАСИЛЬ ЦУРКАН,
СЕРГІЙ ТЕЛЕНИК,
ОЛЕНА ДОРОГА-ІВАНЮК

ПОРІВНЯЛЬНИЙ АНАЛІЗ ПЛАТФОРМ ПРОЕКТУВАННЯ АРХІТЕКТУРИ КРИТИЧНОЇ ІТ-ІНФРАСТРУКТУРИ

Досліджено концепцію архітектури критичної інформаційної інфраструктури шляхом аналізування десяти корпоративних архітектурних платформ: Захмана, TOGAF, FEAF, DoDAF, VMDF, NATOAF, TEAF, GEAF, RM-DOP, SOA. Архітектура відіграє важливу роль у розвитку інформаційних систем. Акт створення архітектури в циклі розробки, як правило,

розуміється як систематичний аналіз та опрацювання відповідної інформації з метою створення моделі для керування фактичним розвитком інформаційних систем. На сьогоднішній день існує понад сто платформ для розробки архітектури, які поділяються за використанням на оборонні, урядові, відкритого типу, пропріетарні. Платформа допомагає покращити розуміння теми шляхом надання систематичних підходів до проектування та розвитку архітектури, але багато аспектів архітектури залишаються неоднозначними. Неясність полягає в наступному: осяжність архітектури (чи повинна архітектура охоплювати лише програмні компоненти або включати інші аспекти розвитку критично важливої ІТ-інфраструктури), роль архітектора (роль архітектора в життєвому циклі розвитку критичної ІТ-інфраструктури часто незрозуміла), результати (що має бути результатом роботи з архітектурою - документи бізнес-функцій чи детальний проект критичної ІТ-інфраструктури), архітектурна діяльність (включає в себе проектування та моделювання, але який рівень деталізації потрібно використати та коли починається детальне проектування), перевірка архітектури (наскільки ми повинні оцінювати, перевіряти результати проектування архітектури), системні вимоги (розмір та складність, чи можуть системи різних розмірів та складності мати однакові системні вимоги до результатів проектування архітектури), рівень архітектури (який взаємозв'язок між архітектурою підприємства критичної інфраструктури та автономною архітектурою критичної ІТ-інфраструктури). Для архітектури складної системи, такої як критична ІТ-інфраструктура, представлені міркування, які складаються з таких аспектів, як бізнес-вимоги, технічні вимоги, критерії, поточна архітектура та майбутня архітектура. Ми пропонуємо аналізувати платформи з різних точок зору. Спочатку платформи аналізуються з точки зору їх цілей, внесків та результатів. По друге, кожна платформа проаналізована в поняттях концепцій, моделей та процесів. У якості третьої та четвертої точок зору використано деякі якісні та кількісні показники для аналізу наведених платформ.

Ключові слова: архітектура, архітектурна платформа, архітектурна платформа підприємства, порівняльний аналіз, критична ІТ-інфраструктура.

ЯРОСЛАВ ДОРОГОЙ,
ВАСИЛИЙ ЦУРКАН,
СЕРГЕЙ ТЕЛЕНИК,
ЕЛЕНА ДОРОГАЯ-ИВАНЮК

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПЛАТФОРМ ПРОЕКТИРОВАНИЯ АРХИТЕКТУРЫ КРИТИЧЕСКОЙ ИТ-ИНФРАСТРУКТУРЫ

Исследовано концепцию архитектуры критической информационной инфраструктуры путем анализа десяти корпоративных архитектурных платформ: Захмана, TOGAF, FEAF, DoDAF, BMDAF, NATOAF, TEAF, GEAF, RM-DOP, SOA. Архитектура играет важную роль для развития информационных систем. Акт создания архитектуры в цикле разработки, как правило, понимается как систематичный анализ и обработка соответствующей информации с целью создания модели для управления фактическим развитием информационных систем. На сегодняшний день существует более ста платформ для разработки архитектуры, которые делятся по использованию на оборонные, правительственные, открытого типа, проприетарные. Платформа помогает улучшить понимание темы путем предоставления систематических подходов к проектированию и развитию архитектуры, но много аспектов архитектуры остаются неоднозначными. Неясность состоит в следующем: осягаемость архитектуры (должна ли архитектура охватывать только программные компоненты или включать другие аспекты развития критически важной ИТ-инфраструктуры, роль архитектора (роль архитектора в жизненном цикле развития критической ИТ-инфраструктуры часто непонятна), результаты (что должно быть результатом работы с архитектурой – документы бизнес-функций или детальные проект критической ИТ-инфраструктуры), архитектурная деятельность (включает в себя проектирование и моделирование, но какой уровень детализации нужно использовать и когда начинается детальное проектирование), проверка архитектуры (насколько должны оценивать, проверять результаты проектирования архитектуры), системные требования (размер и сложность, могут

ли системы различных размеров и сложности иметь одинаковые системные требования к результатам проектирования архитектуры), уровень архитектуры (какая взаимосвязь между архитектурой предприятия критической инфраструктуры и автономной архитектурой критической ИТ-инфраструктуры). Для архитектуры сложной системы, такой как критическая ИТ-инфраструктуры, изложены подходы, которые состоят из таких аспектов как бизнес-требования, технические требования, критерии, текущая и будущая архитектуры. Предлагается анализ платформ с различных точек зрения. Во-первых, они анализируются с точки зрения их целей, вкладов и результатов. Во-вторых, каждая платформа проанализирована в понятиях концепций, моделей и процессов. В качестве третьей и четвертой точки зрения использовано качественные и количественные показатели анализа приведенные платформ.

Ключевые слова: архитектура, архитектурная платформа, архитектурная платформа предприятия, сравнительный анализ, критическая ИТ-инфраструктура.

Yaroslav Dorohyi, candidate of technical sciences, associate professor, doctoral student, National technical university of Ukraine “Igor Sikorsky Kyiv polytechnic institute”, Kyiv, Ukraine.

E-mail: argusyk@gmail.com.

Vasyl Tsurkan, candidate of technical sciences, associate professor at the cybersecurity and application of information systems and technologies academic department, Institute of special communications and information protection National technical university of Ukraine “Igor Sikorsky Kyiv polytechnic institute”, Kyiv, Ukraine.

E-mail: v.v.tsurkan@gmail.com.

Serhii Telenyk, doctor of technical sciences, professor, head of automation and control in technical systems academic department, National technical university of Ukraine “Igor Sikorsky Kyiv polytechnic institute”, Kyiv, Ukraine.

E-mail: S.telenyk@gmail.com.

Olena Doroha-Ivaniuk, postgraduate student, National technical university of Ukraine “Igor Sikorsky Kyiv polytechnic institute”, Kyiv, Ukraine.

E-mail: Dorogaya.helen@gmail.com.

Ярослав Юрійович Дорогий, кандидат технічних наук, доцент, докторант, Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”, Київ, Україна.

Василь Васильович Цуркан, кандидат технічних наук, доцент кафедри кібербезпеки та застосування автоматизованих інформаційних систем і технологій, Інститут спеціального зв'язку та захисту інформації Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського”, Київ, Україна.

Сергій Федорович Теленик, доктор технічних наук, професор, завідувач кафедри автоматики та управління в технічних системах, Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”, Київ, Україна.

Олена Дорога-Іванюк, аспірант, Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”, Київ, Україна.

Ярослав Юрьевич Дорогой, кандидат технических наук, доцент, докторант, Национальный технический университет Украины “Киевский политехнический институт имени Игоря Сикорского”, Киев, Украина.

Василий Васильевич Цуркан, кандидат технических наук, доцент кафедры кибербезопасности и применения автоматизированных информационных систем и технологий, Институт специальной связи и защиты информации Национального технического университета Украины “Киевский политехнический институт имени Игоря Сикорского”, Киев, Украина.

Сергей Федорович Теленик, доктор технических наук, профессор, заведующий кафедрой автоматики и управления в технических системах, Национальный технический университет Украины “Киевский политехнический институт имени Игоря Сикорского”, Киев, Украина.

Елена Дорогая-Иванюк, аспирант, Национальный технический университет Украины “Киевский политехнический институт имени Игоря Сикорского”, Киев, Украина.