



**HAL**  
open science

# A Proposal of a Situational Approach for Enterprise Architecture Frameworks: Application to TOGAF

Elena Kornyshova, Rébecca Deneckère

► **To cite this version:**

Elena Kornyshova, Rébecca Deneckère. A Proposal of a Situational Approach for Enterprise Architecture Frameworks: Application to TOGAF. International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022), Sep 2022, Verona, Italy. pp.3499-3506, 10.1016/j.procs.2022.09.408 . hal-03686090

**HAL Id: hal-03686090**

**<https://hal.science/hal-03686090>**

Submitted on 21 Dec 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License



26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022)

# A Proposal of a Situational Approach for Enterprise Architecture Frameworks: Application to TOGAF

Elena Kornyshova<sup>a,\*</sup>, Rébecca Deneckère<sup>b</sup>

<sup>a</sup>Centre d'études et de recherche en informatique et communications, Conservatoire National des Arts et Métiers, Paris, France

<sup>b</sup>Centre de Recherche en Informatique, Université Paris 1 Panthéon-Sorbonne, Paris, France

---

## Abstract

Enterprise Architecture (EA) frameworks have justified their efficiency to improve the enterprise functioning by providing a whole vision of an organization aligned to its strategy. However, these frameworks are often heavy and are not used integrally as they are not completely suitable for the situation at hand. To tackle this problem, we suggest using a component-based vision of EA frameworks. Our goal is to identify a set of EA components that could be used independently from each other and to use them depending on the context. This approach could be used to implement the whole EA method in an organization, to provide a progressive integration of different components, or to improve an existing EA by adding the lacking components. We called our proposal the SEA (Situational Enterprise Architecture) approach. In this paper, we propose a model to formalize EA components and illustrate this model with the TOGAF framework.

© 2022 The Authors. Published by ELSEVIER B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>)

Peer-review under responsibility of the scientific committee of KES International

*Keywords:* Enterprise Architecture; Components; Situational Method Engineering

---

## 1. Introduction

Enterprise Architecture (EA) is a coherent set of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure [1]. EA Management (EAM) aims at defining a holistic view of an organization around its Information System (IS) to

---

\* Corresponding author. Tel.: + 331 40 27 25 51; fax: +331 40 27 27 09.

*E-mail address:* elena.kornyshova@cnam.fr

optimize its global functioning, more precisely, to (i) enhance strategic business outcomes, (ii) reduce IT costs, (iii) bring together different and conflicting stakeholders' interests, and (iv) improve reactivity and risk management [2].

Since key initial work in the EAM field in 1987 by Zachman [3], many approaches were developed [1] [2] [4], generally using frameworks to structure the different architectural domains. The latter EA frameworks usually provide a context in which all stakeholders in an organization can communicate and collaborate about their enterprise architecture [5]. They generally contain different artefacts describing the enterprise strategy, processes, organization, data, applications, technologies, and so on, and the relationships between these artefacts considering static and dynamic aspects. The most known frameworks are the Zachman framework [3], The OPEN Group Architecture Framework (TOGAF) [6], and the USA Department of Defense Architecture Framework (DODAF) [7]. Another group of EA approaches is unified within the school of Urbanization (the French school of EA), which is based on the city-planning metaphor [8] [9]. These approaches are more oriented toward the establishment of the rules and principles conducting the EA processes rather than the usage of frameworks.

In the literature, several works deal with the adaptation of EA frameworks to different kinds of companies. In general, the goal of these works is to suggest a simplified version of the EA framework for a small or medium enterprise or for its lightweight usage [10]. For instance, [11] suggests configuring EA framework, and [12] proposes a component-based method for creating EA documentation. However, none of these works propose an adapted EA method or a set of components suitable for a given case.

Situational Method Engineering (SME) promotes a situation-specific method construction by reusing parts of existing methods (autonomous components) stored in method repositories. Many SME approaches have been proposed [13] but their implementation in practice can be difficult as companies are slow to adopt them even though they acknowledge the significance of the role that methods play in their engineering activities. A way to use SME in a smoother way is the method family approach. The initial methods are unified into a family using the variability analysis and the intentional perspective [14] [15]. In this approach, the common components are factorized, and the variable components are collected into decision points and selected in function of the context.

Based on the method family approach of SME, our main goal is to propose an approach of Situational Enterprise Architecture (SEA) management by applying the SME principles to EA frameworks. This approach will include a base of method components of the main EA frameworks unified within an EA method family and a guideline on how to select the EA method components fitting a given case. The SEA approach could be used to implement the whole EA method in an organization, to provide a progressive integration of different components, or to improve an existing EA by adding the lacking components. This answers the challenge to have adaptable and scalable EA frameworks.

In this paper, we present the SEA approach and its conceptual models of the Method Family and of method component and we describe the set of EA method components issued from TOGAF as one of the most known EA frameworks. Section 2 presents the SEA approach and its application to TOGAF. Section 3 is dedicated to Related Works. We discuss the advantages of the SME-oriented EA modeling and conclude in Section 4.

## **2. Situational Enterprise Architecture (SEA) Approach**

In this section, we introduce briefly the SEA approach construction (Sub-section 2.1.) and then we give an overview of the SEA approach together with the underlying conceptual model (Sub-section 2.2.). We aim to use Situational Method Principles on Enterprise Architecture?" This leads to three hypotheses: H1: It is possible to define EA method components, H2: It is possible to define the context of the use of EA components to help their selection, and H3: The use of a method family to organize the EA components helps the guidance. In this paper, we focus on H1 and H2.

### *2.1. SEA Approach Construction*

The SEA approach is based on the principles of Situational Method Engineering (SME). We have selected the method family approach [14] [15], which simplifies the application of the SME principles. The SEA approach is elaborated using the methodology presented in [16]. The first step is the identification of the goals and scope of the proposed approach. The next steps are the EA method components' extraction, description, their organization within a method family, and, finally, specification of decision points.

The SEA approach can answer several EA challenges:

- Customize the EA method family into an adapted method. In this case, a particular EA approach is elaborated accordingly to the needs of a given company.
- Provide progressive integration of the EA method components. This is a more lightweight usage of the EA approach, as only the main EA components are identified, and their implementation is done one by one.
- Select one or a set of adapted components to enhance an existing EA. In this case, an EA framework is already implemented but it does not work well. Based on the context including the actual state of EA, the goal is to complete it with the lacking EA components.

The scope of the SEA approach is limited to the main EA frameworks. At this stage of our work, we began to study TOGAF as it is the most widespread and complete EA framework. TOGAF recognizes that it is contextual and adaptable [6], even if the adaptation guidelines are limited. It includes ADM (Architecture Development Method) guidelines (EA implementation in a company) which includes 10 phases: Preliminary phase, Architecture Vision, Business Architecture, Information System Architecture, Technology Architecture, Opportunities and Solutions, Migration Planning, Implementation Governance, Architecture Change Management, and Requirements Management.

## 2.2. EA Method Component Meta-Model

The EA method component is the core concept of the model. It represents a building block, a subset of a method, which may contain other components. The method components are organized in method families, each one specific for a predefined domain [14] [15]. In a method family, the components may be considered either as common or variable. Common components will be mandatory regardless of the context whereas variable components will have to be selected following the characteristics of the project at hand. These situations are called variation points. The variability dependency represents the relationship between the variation point and the component (mandatory or optional - respectively corresponding to common and variable method components). We can say that a set of variable method components represent the alternatives offered to the engineer at a specific variation point. The method component context characterizes all possible situations in which this component may be applied.

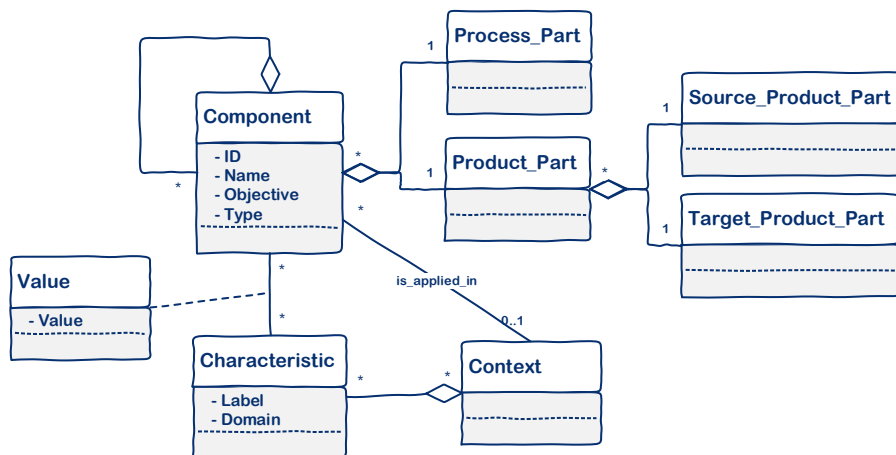


Fig. 1. EA Component Meta-Model (adapted from [14]).

Different types of method components exist fragment, chunk, component, and so on [14]. We have adapted the meta-model from [14] to represent the EA method components (Figure 1). This kind of method component includes an intentional part (signature) corresponding to the intention of the given component and to its usage situation; a product

part representing the situation before and after the method component application; a process part describing the component execution guidelines (of different types depending on the component complexity). A context composed of different characteristics with the corresponding values is associated with each component to provide a better method to select components. A component could be atomic or composite (composed of a set of more detailed components).

[17] gives a typology of characteristics that can be used in the construction of a method family. Generic characteristics can be of four types: organizational, human, application domain, or development strategy. There are also specific characteristics that can only be used in this domain.

We based the EA components on the structure of the method chunks (of the assembly-based SME approach [18]). A method chunk ensures a tight coupling of the process part and its related product part. It is a coherent module, and any method is viewed as a set of loosely coupled method chunks expressed at different levels of granularity. This modular view of methods favors their adaptation and extension and permits to reuse chunks of a given method in the construction of new ones [18]. The structure of the method chunk that we use in this work is an improved one [14] which considers context and decision points with indicators and values.

The second main SME approach used in this work is the method family one [14] [15]. The originality of this approach mainly stays in the organization of the components, which facilitates their selection and reuse. A Method Family is a set of method components, organized for a specific domain. Thus, the engineer selects a specific method line, inside a family, to apply to a specific project. A method line is a particular subset of method components of a method family selected for a given situation. More precisely, according to the project characteristics and requirements, method components are selected in the method family in order to construct a project-specific method line. A method line includes all common method components of the method family and some variable method components selected for the given project based on the project characteristics. This approach allows using interoperable components, organized for a specific domain, based on the situation context. This concept has been successfully used in different domains such as decision-making [14], requirement elicitation [25], data analysis for humanities [26], information services co-creation [27], and agile methods [16] among others. We believe that the organization of EA method components within a method family will facilitate the selection and usage of the appropriate components depending on the context as it is already done in the mentioned fields.

Table 1. List of EA Method Components from the Information System Architecture Phase.

ID	Component Name	Phase
CD1	Select reference models, viewpoints, and tools for the data architecture	C: IS data Architecture
CD2	Describe the Baseline Data Architecture to support the target data architecture	
CD3	Develop a Target Description for the Data Architecture to support the Architecture Vision and Target Business Architecture.	
CD4	Verify the data architecture models for internal consistency and accuracy by analysis the gap between baseline and target data architecture	
CD5	Prioritize data architecture activities over the coming phases by defining candidate data roadmap components	
CD6	Identify the potential impacts on the data architecture Landscape	
CD7	Verify the motivation for the proposed data architecture project by conducting formal stakeholder review	
CD8	Verify the motivation for the proposed data architecture project by conducting formal stakeholder review	
CD9	Finalize the Data Architecture	
CD10	Document the Data Architecture sections in the Architecture Definition Document	
CA1	Select reference models, viewpoints, and tools for the application architecture	C: IS application Architecture
CA2	Describe the Baseline Application Architecture to support the target application architecture	
CA3	Develop a Target Description for the Application Architecture to support the Architecture Vision and Target Business Architecture.	
CA4	Verify the application architecture models for internal consistency and accuracy by analysis the gap between baseline and target application architecture	
CA5	Prioritize application architecture activities over the coming phases by defining candidate application roadmap components	
CA6	Identify the potential impacts on the data architecture Landscape	
CA7	Verify the motivation for the proposed application architecture project by conducting formal stakeholder review	
CA8	Finalize the Application Architecture	
CA9	Document the Application Architecture sections in the Architecture Definition Document	

### 2.3. EA Method Component Illustration with TOGAF

We have analyzed TOGAF ADM and identified 88 method components (accessible at [https://cri-dist.univ-paris1.fr/method-family/TOGAF\\_EA\\_Method\\_Components.pdf](https://cri-dist.univ-paris1.fr/method-family/TOGAF_EA_Method_Components.pdf)). As an illustrative example, we present in this paper the list of the EA method components from the Information System Architecture phase, which is composed of the Data Architecture and Application Architecture (cf. Table 1).

An example of the EA method component from TOGAF is given in Figure 2. This component represents a specific part of the TOGAF C phase (Information Systems Data Architectures) on the review and validating of the set of data principles, resources (reference models, patterns, etc.), viewpoints (stakeholders of the data, various time dimensions, locations, business processes, etc.), tools and techniques (simple documents or spreadsheets, or more sophisticated modeling tools and techniques such as data management models, data models, etc.) to use in the EA project. The component describes the source and target product parts to show the added value of its use. The process part explains the different steps to execute to reach the objective. Finally, the context shows the characteristics values inherent to the component (for instance, the component has to be executed by an enterprise architect who has a high level of expertise; it doesn't cost a lot as it is not very long; etc.).

<b>ID:</b> CD1	<b>Name:</b> Select reference models, viewpoints, and tools for the data architecture		<b>Type:</b> Atomic
<b>Objective:</b> To select the set of the reference models, viewpoints, and tools for the Data Architecture.			
<b>Source Product Part:</b>		<b>Target Product Part:</b>	
<ol style="list-style-type: none"> <li>1. Organizational Model for Enterprise Architecture</li> <li>2. Tailored Architecture Framework</li> <li>3. Architecture Vision</li> <li>4. Architecture Repository</li> <li>5. Draft Architecture Definition Document</li> <li>6. Draft Architecture Requirements Specification</li> </ol>		<ol style="list-style-type: none"> <li>1. Overall Data Modeling Process</li> <li>2. Catalogs of Data Building Blocks</li> <li>3. Matrices (the core relationships between related model entities)</li> <li>4. Diagrams</li> <li>5. Types of Requirements to be Collected</li> </ol>	
<b>Additional product part</b> (not mandatory for the execution of the component):			
<ol style="list-style-type: none"> <li>7. Data principles, if existing</li> <li>8. Statement of Architecture Work</li> <li>9. Business Architecture components of an Architecture Roadmap</li> </ol>			
<b>Process Part:</b>			
<ol style="list-style-type: none"> <li>1. Determine Overall Modeling Process</li> <li>2. Identify Required Catalogs of Data Building Blocks</li> <li>3. Identify Required Matrices (the core relationships between related model entities)</li> <li>4. Identify Required Diagrams</li> <li>5. Identify Types of Requirements to be Collected</li> </ol>			
<b>Context:</b>			
<b>Generic characteristics:</b>		<b>Specific characteristics:</b>	
Expert role := Enterprise Architect	Time pressure := low	EA scope := Intra-organization, Organization level	
Expertise degree := high	Formality degree := medium	Relationships with other EA components := high	
Importance := high	Complexity degree := high	EA level := Data Architecture	
Impact := high	Iterativity := one-shot		
Cost of component implementation := low	Organizational size := any		
Management commitment := medium			

Fig. 2. EA Component “Select reference models, viewpoints, and tools for the data architecture”.

## 3. Related Works

### 3.1. EA Frameworks

The Zachman framework [3] provides a logical structure for organizing the enterprises' artifacts. It consists of a 6x6 matrix. The columns represent six aspects of the enterprise that can be described or modeled: data, function, network, people, time, and motivation; the rows represent six viewpoints from which the aspects can be described: scope, business, system, technology, detailed representation, and functioning enterprise viewpoints. Each cell represents an aspect of the enterprise modeled from a particular viewpoint, which can then be selected by an enterprise architect to serve a specific purpose. This means that this framework allows architects to look at an enterprise system in an organized way, which helps in analyzing the system as a whole.

The MoDAF framework [28] has been defined by the UK Ministry of Defense to be capable of integrating various IT systems inside the ministry. MoDAF consists of six viewpoints: the overall viewpoint, operational viewpoint, system viewpoint, technical viewpoint, standard viewpoint, and acquisition viewpoint.

DoDAF [7] is a framework that was developed for US defense systems. It classifies architectures based on four views: system, technical, operational, and overall.

The Gartner framework [29] has been defined with the objective to be consolidated from top to bottom, which means to first consider the business, then information, applications, and finally the technology. Most of the effort is spent on communicating, strategizing, leading, and governing, while architecting receives little attention.

The TOGAF framework [6] is derived from the US DOD framework and is meant to improve the business efficiency of organizations. TOGAF can provide simple implementation and excellent alignment between business and IT. The framework consists of six components: architecture development method (ADM), architecture content framework, reference models, ADM guidelines and techniques, enterprise continuum, and enterprise capability framework. It is an iterative process that comprises eight phases. There are several artifacts that are produced across the TOGAF ADM cycle and organized into three categories: catalogs, matrices, and diagrams.

The FEAF framework [30] was developed to assist the US government federal entities to use EA by eliminating duplicated resources and increasing shared systems. It defines a number of general principles to ensure that potential investment and architectural decisions are weighed (Future Readiness, Investment Support, Shared Services, Interoperability Standards, Information Access, Security and Privacy, and Technology Adoption).

The OEAF framework [31] was developed by Oracle to engage its customers in defining strategic plans which would increase the alignment between business and IT. OEAF was defined to be a hybrid of other EA frameworks. The OEAF focuses on one principle, which is to create only the necessary structure for an organization that can be delivered on time and accomplishes the organization's business requirements.

All EA frameworks share the same purpose, which is to create an EA that increases the alignment of business and IT resources. However, no real guidance is offered to choose between them. In addition, several elements are common in all frameworks, others are really specific to one framework only.

### *3.2. Related Works on the Adaptation of the EA Frameworks*

Main EA frameworks are briefly introduced in section 1 and could be consulted in detail in the corresponding referenced papers. We focus here on the papers presenting attempts to adapt EA frameworks to different contexts.

A survey concerning the applicability of TOGAF in small and medium enterprises is proposed in [10]. This survey shows that TOGAF experts see great potential in applying EAM in Small and Medium Enterprises, even if the cost due to its complexity is very high. The authors conclude that there is a need to reduce the complexity of the application of the EA methods to help their integration into Small and Medium Enterprises. However, few proposals have been made to reduce this complexity. [32] proposes the SMEAG model that combines the existing growth stage model of Scott and Bruce, the Enterprise Architecture Framework by Hoogervorst, and the EA as Foundation for Business Execution Model by Ross, Weill, and Robertson. However, the proposed approach is still complex. Some works propose approaches addressing specific domains, like the ECLORA project [33] whose aim is to develop a reference EA for Small and Medium Enterprises in the utility industry to help to tailor their actual EA to their business demands depending on its characteristics like holden market roles.

Some works make easier the EA approaches by simplifying the inside metrics, like [34]. However, this approach has been proposed only for Archimate (EA modeling language of the Open Group) and its use on EA frameworks, in general, is still to be demonstrated.

[35] states that often the knowledge of the people that are responsible for business processes, applications, or processes is just ignored. Using Web 2.0 and Enterprise 2.0 techniques, it uses the 'wisdom of the crowd' to develop the information models. All employees contribute and make explicit their expert knowledge for the parts of the EA they are responsible for. Wiki techniques are used for the documentation of EA-related concepts and the proposal offers an open templating mechanism for wiki pages to collaboratively evolve an information model in the context of EA management. Two ways of lightweight EA are possible, bottom-up by evolving the model by means of the templates and top-down by using template definition. Even if this approach uses interesting techniques, it is only highlighting the documentation part of the EA methods.

The CHOOSE meta-model for EA in Small and Medium Enterprises [11] proposes to create configurable EA methods. It is based on the essential dimensions of EA frameworks and is conceived as simple so that it may be applied in a Small and Medium Enterprises context. It kept the number of meta-model concepts and relationships relatively lower than in other EA frameworks. However, as the entire CHOOSE approach must be implemented to be useful, its complete knowledge is necessary to use it.

Situational methods techniques offer a way to simplify any method by decomposing it into a set of components easier to implement one by one in a progressive integration [36]. Therefore, a way to simplify EA methods could be to decompose them into EA components. [12] proposes several composable EA semi-automated documentation techniques in order to create a situational method for EA documentation. Each technique is composed of a purpose, a process, roles, context, and configuration options. The application context is stated for each of these techniques to help with the run selection. The techniques' description corresponds to the concept of process method fragment as they don't include the product part of the techniques (as in method components or method chunks [14]), which means that we don't know formally what specific concepts are used in these documentation components. The proposed method construction process doesn't explain how the assembly is technically done (how the techniques processes are put together in the same method) but only what are the main steps. It represents a static view of the EA state. Our goal is to include the method of EA construction and its evolution to the To-Be state of EA.

#### 4. Conclusion and Future Works

Our goal is to elaborate an approach of Enterprise Architecture that will be flexible, light, and adaptable to the situation at hand. We use the SME principles to construct this approach and call it SEA (Situational Enterprise Architecture). We have distinguished the three most important use cases for our approach: implementing an adapted to a given context EA method, providing a progressive integration of EA method components, and selecting EA method components to complete an already implemented EA framework.

SME allows to define a method following the situation at hand. All methods can be cut into a set of reusable components that can be used to define new methods. Method family, the last approach of the SME field of research, proposes to guide the selection of the variable components by using a characterization of the context of reuse and by defining variation points. The configuration process of method families allows to define method lines, completely adapted to the project context, in a smoother way than the preceding SME approaches.

The advantage of using a specific method family for EA components is that several EA method lines can be defined, from this family, following the specific needs of the EA project, as the components will be selected based on these needs. For instance, the method line created to implement a whole EA method in an organization will be composed of a bigger set of components than the method line chosen to improve an existing EA which will only be composed of the EA components lacking in the method. A progressive integration also will be possible with another method line. All these EA method lines will be a subset of the set of components composing the EA method family.

In this paper, we have presented the SEA approach and started to apply it to the TOGAF EA framework. We have identified 88 method components within TOGAF. We showed that it is possible to define EA components, which can be used in a flexible way (H1) and that it is possible to define their context of use (H2). In our future works, we intend to firstly validate H3 by demonstrating that the organization of EA method components in a method family helps to guide the user in their selection on the fly. We also plan to give a complete description of these 88 components, complete them with components from other EA frameworks, and finalize our approach by formalizing decision points to offer better guidance for the application of our approach in practice.

#### References

- [1] Lankhorst M.M. editor. (2005) "Enterprise Architecture at Work: Modelling", *Communication and Analysis*. Springer, Berlin, Germany.
- [2] Ross J.W., Weill P., and Robertson D.C. (2006) "Enterprise architecture as strategy: creating a foundation for business execution". *Harvard Business School Press*, Boston, Massachusetts.
- [3] Zachman J.A. (1987) "A framework for information systems architecture". *IBM Systems Journal*, 26(3).
- [4] Greefhorst D. and Proper H.A. (2011) "Architecture Principles – The Cornerstones of Enterprise Architecture". *Enterprise Engineering Series*. Springer, Berlin, Germany.



- [5] Jensen C.T., Cline O., and Owen M. (2011) “Combining Business Process Management and Enterprise Architecture for Better Business Outcomes”. *International Business Machines Corporation*, USA.
- [6] The TOGAF® standard, version 9.2 (2022) <https://publications.opengroup.org/standards/enterprise-architecture/c182>, Accessed by February 2022.
- [7] The DoDAF Architecture Framework Version 2.02. (2011) Department of Defense.
- [8] Caseau Y. (2008) “Urbanisation, SOA et BPM”. Dunod, France (In French).
- [9] Longépé C. (2006) « Le projet d’urbanisation du SI: Démarche pratique avec cas concret ». Dunod, Paris (In French).
- [10] Alm R., and Wissotzki M. (2013) “TOGAF Adaption for Small and Medium Enterprises”, *International Conference on Business Information Systems (BIS 2013)*, 112-123.
- [11] Bernaert M., Poels G., Snoeck M., and De Backer M. (2016) “CHOOSE: Towards a meta model for enterprise architecture in small and medium-sized enterprises”, *Information System Frontiers journal*, Vol 18, 781-818.
- [12] Farwick M., Schweda C.M., Breu R., and Hanschke I. (2016) “A situational method for semi-automated enterprise architecture documentation”, *Software System Model journal*, vol 15, 397-426.
- [13] Henderson-Sellers B., Ralyté J., Ågerfalk P., and Rossi M. (2014) “Situational Method Engineering”. Springer, ISBN 978-3-642-41466-4, 1-274.
- [14] Kornysheva E. (2011) “MADISE: Method Engineering-based Approach for Enhancing Decision-Making in Information Systems Engineering”. PhD Thesis, Paris, France.
- [15] Kornysheva E., Deneckère R., and Rolland C. (2011) “Method Families Concept: Application to Decision-Making Methods”. *Enterprise, Business-Process and Information Systems Modeling. BPMDS 2011, EMMSAD 2011*, London, UK.
- [16] Deneckere R., Kornysheva E., and Ralyte R. (2014) « Famille de méthodes : la flexibilité au cœur du processus de construction de méthode », *Revue Ingénierie des Systèmes d’information (RSTI-ISI)*, 19/1, 67-95.
- [17] Kornysheva E., Deneckere R., and Claudepierre B (2010) “Contextualization of method components”, *Proceedings of the Research Challenges in Information Science international conference (RCIS)*, Nice, France. 235-246.
- [18] Ralyté J., and Rolland R. (2001) “An Assembly Process Model for Method Engineering”. *Proceedings of CAISE 2001*, Springer, Berlin, 267-283.
- [19] Deneckere R. (2001) « Approche d’extension de méthodes fondée sur l’utilisation de composants génériques ». PhD thesis. University of Paris 1 Panthéon-Sorbonne.
- [20] Karlsson F., and Ågerfalk P. J. (2008) “Method Configuration: The eXtreme Programming Case”, in XP 2008, Limerick, Ireland, pp 32-41.
- [21] Rossi M., Ramesh B., Lyytinen K., and Tolvanen J.-P. (2004) “Managing evolutionary method engineering by method rationale”. *Journal of the AIS*, vol. 5(9), 356-391.
- [22] Cervera M., Albert M., Torres V., and Pelechano V. (2012) “A Model- Driven Approach for the Design and Implementation of Software Development Methods”. *International Journal of Information System Modeling and Design (IJISMD)*, vol. 3(4), 86-103.
- [23] Guzélian G., and Cauvet C. (2007) “SO2M: Towards a Service-Oriented Approach for Method Engineering”, *Proceedings of IKE’07*, Las Vegas, Nevada, USA.
- [24] Iacovelli A. (2012) « Approche orientée service pour la configuration de méthodes outillées ». PhD thesis. University Paris 1 Panthéon-Sorbonne.
- [25] Deneckere R., Kornysheva E., and Rolland C. (2010) “Method Family Description and Configuration”, *ICEIS*, Beijing, China. 384-387.
- [26] Ammar A., Hug C., and Deneckere R. (2014) “Intentional Process Modeling of Statistical Analysis Methods”. *Computer Applications and Quantitative Methods in Archaeology (CAA)*, France.
- [27] Ralyté J. (2013) “Towards a Method Family Supporting Information Services Co-Creation in the Transdisciplinary Context”, *International Journal of Information Systems Modeling and Design*, IGI-Global, vol. 4(3).
- [28] UK Ministry of Defence. (2012) “Mod architecture framework”. <https://www.gov.uk/guidance/modarchitecture-framework>.
- [29] R. Sessions. (2007) “A Comparison of the Top Four Enterprise Architecture Methodologies”. <https://msdn.microsoft.com/en-us/library/bb466232.aspx>.
- [30] The Executive Office of the President of the United States (EOPOTUS). (2012) “A Common Approach to Federal Enterprise Architecture”. Technical Report.
- [31] R. Covington, H. Jahangir, G. Wright, P. Silverstein, H. Dia, and B. Rasmussen. (2009) “The oracle enterprise architecture framework”. White Paper Oracle, <http://www.oracle.com/technetwork/articles/entarch/oeaframework-133702.pdf>.
- [32] Jacobs D., Kotzé P., van der Merwe A., and Gerber A. (2011) “Enterprise Architecture for Small and Medium Enterprise Growth”, *Advances in Enterprise Engineering V: First Enterprise Engineering Working Conference (EEWC)*, Antwerp, Belgium, May 16-17, Proceedings, 61-75.
- [33] Timm, F., Wißotzki, M., Köpp, C., and Sandkuhl, K. (2015) “Current state of enterprise architecture management in SME utilities”. *Proceedings of INFORMATIK 2015*, Bonn, S. 895-907.
- [34] Singh P., and van Sinderen C. (2015) “Lightweight Metrics for Enterprise Architecture Analysis”, *DBpedia in the Art Market*, 113-125.
- [35] Buckl S., Matthes F., Neubert C. and Schweda C.M. (2010) “A lightweight Approach To Enterprise Architecture Modeling and Documentation”, *International Conference on Advanced Information Systems Engineering (CAiSE 2010)*, pp 136-149.
- [36] Deneckère R., Kornysheva E. and Iacovelli A. (2016) “Progressive integration of method components: A case of agile IS development methods”. In proceedings of RCIS 2016, Grenoble, France.