The CSI Journal on
Computer Science and Engineering
Vol. 15, No. 2, 2018
Pages 5-11
Regular Paper

Towards a Knowledge Management-Based Framework for Aligning Enterprise and Software Architecture

Razieh Dehghani[1] Jafar Habibi[1]

[1]Department of Computer Engineering, Sharif University of Technology,Tehran, Iran

Abstract

Continuous changes in organizational requirements, including the architecturally significant ones, have drawn researchers' attentions to the importance of architecture evolution process. Considering the widespread effect of structural design decisions, the consistency with previously designed architectural structures should thus be continuously preserved throughout the evolution process. Evolving Enterprise Architecture (EA) and Software Architecture (SA), as a kind of architectural structure, is thus important. One of the approaches for managing this process is managing the evolution process knowledge. The knowledge that has been used for designing the architecture is as valuable as the designed architecture itself because the architecture would be destroyed if it is updated without considering the knowledge that was used (or created) during the design process. On the other hand, one of the reasons for which updating the architecture is important is the need for aligning SA and EA. Although the idea of using Knowledge Management (KM) process for evolving architecture has somehow been addressed, researchers have sporadically studied the applicability of using KM techniques for addressing the alignment issues. In this research, a framework is proposed for aligning the SA and EA based on KM process. Architectural designers can use this framework for aligning the architectures, and researchers can use it for proposing new alignment methods.

Keywords: Architecture Alignment, Enterprise Architecture, Software Architecture, Knowledge Management-Based Framework, Knowledge Management Process.

1. Introduction

Business-IT alignment has been researched in various dimensions including strategy (like [1]), architecture (like [2]), and process (like [3]). This research is aimed at providing a framework for aligning SA an EA. Although being expert in the area of SA is not a prerequisite for designing EA, EA designers should have a glance at designed SAs, in order to ensure the consistency between EA and SA. In return, SA designers should consider the consistency between EA and SA. Sometimes, the previously designed EA (SA) should be updated to be aligned with the new SA (EA), so that its practicability is maintained. This may require redesigning the architecture, which results in wasting the organizational assets as well as missing the deadlines.

Although process of designing EA has been viewed from different perspectives, most of the researchers have unanimously agreed that the EA design is aimed at engineering/reengineering the business enterprise in a way that the organizational business goals are accomplished through aligning the business activities with action plans throughout the whole organization. SA is also recognized as an engineered structure which provides an organized framework for the target software, and is inspired by architecturally significant requirements [4].

Various research efforts have been made in the area of architectural alignment. Although KM has been used for evolving designed architectures, KM process has not been directly used for aligning architectures. It should also be mentioned that in most of the conducted research, alignment has been seen as a generic process, rather than an engineering process which requires identifying alignment requirements precisely, and also prescribing situational alignment solutions based on the identified requirements.

Available EA and SA management approaches do not support KM process, through the commonalities have somehow been approved [5], [6]. We have used these commonalities as a basis for aligning the architectures through managing the knowledge that is required for alignment.

The rest of this paper is organized as follows: Section 2 discusses the background of this research, Section 3 explains the proposed framework, Section 4 discusses the results of evaluating the proposed framework, and finally the last section provides the conclusion and proposes the potential future works.

2. Background Study

Generally, the conducted research in the area of this research incorporates the following sub-areas:

1. Identifying the corresponded components to be aligned (Component-Based Approach): This category of research efforts has focused on identifying the correspondence between SA and EA components. As an expressive example, Pereira and Sousa have provided heuristics within four main components, including Business, Information, Application, and Technology [7].
2. Assessing the established alignment (Evaluation-Based Approach): Some researchers have studied the appropriate way for investigating alignment strategies and also the way of preserving the alignment. As an example, some researchers have proposed a model for assessing the maturity and alignment status of architecture [8].
3. Proposing process-based alignment methods (Process-Based Approach): Some of the researchers have tried to propose methods for aligning architectures. As an example, Eck et al. [9] have proposed a framework which can be used for extracting patterns for aligning strategic and software dimensions. Although this framework has skillfully been engineered, it has not considered the differences between architectural and strategic alignment.
4. Proposing multi-dimensional EAs (Integration-Based Approach): In this category, the researchers have tried to integrate existing EAs in a way that the resulted EA would be more consistent with the existing SAs. For example, Zarvic and Wieringa have proposed such a framework [10].
5. Providing model-based alignment methods (Model-Based Approach): This category of research is concerned about aligning, and does not consider the domain features. In other words, this category of research focuses on logical issues. As an example, Clark et al. have published a research paper of this category [10].
6. Presenting domain-specific alignment approaches (Domain-Based Approach): In this type of research, the researchers have tried to investigate the useful techniques for aligning the architecture in a special domain. As an example, Riempp et al. have focused on special enterprise architecture for US army [12].

3. Proposed Framework

Architectural design is sometimes defined as the process of making a sequence of design decisions [11] [12], and thus aligning the architectures can be seen as aligning these decisions. On the other hand, managing the organizational knowledge impacts on decision making process [13]. In this basis, our proposed framework uses KM process for aligning the design decision processes, which results in aligning software and enterprise architectures. For this reason, the main phases of our proposed framework are the main phases of KM process (Figure 1).

It should be noted that different researchers have presented various phases for managing the organizational knowledge [14], but we have used the four common ones in our framework, named as Discovering, Capturing, Sharing, and Improving. These phases guide the procedure of making appropriate decisions for alignment, and also aligning the decisions that have previously been made.

Falessi et al. have surveyed decision-making techniques for designing software architecture, and they have proposed a process for selecting the best decision-making technique among various alternatives [15]. We have used this process for analyzing the decisions, and also discovering the knowledge that should be shared to alleviate the inconsistencies.

Steps of our proposed framework are sequentially explained below:

1. Discover: This phase is aimed at discovering the existing misalignments or the potential points of misalignment. Furthermore, the knowledge sources that contain the alignment knowledge, which is needed for alleviating misalignments or preventing them to happen, are discovered. For this purpose, the following steps should be taken:

- Activity Identification: Identify the activities that have been taken (or are going to be taken as new decisions) in designing the SA and also the EA.
- Decision Analysis: Table 1 shows some of the most important aspects in making architectural design decisions. In this step, the identified activities in the previous step are analyzed. For each decision, result of performing the under-investigation activity is analyzed; it may improve the design decision or it may make it worse.
- Knowledge Source Identification: Identify all the people who have been (or are going to be) involved in making decisions in the process of designing the software and enterprise architecture. Then, ask them to specify all the knowledge resources that have been (or are going to be) involved in making the intended decisions. Note that all the explicit and tacit knowledge resources should be identified [16], [18].

2. Capture: This phase is targeted at capturing the knowledge that is required for aligning the decisions that have been made (or the decisions that are going to be made). The following steps are prescribed to achieve this goal:

- Knowledge Elicitation: As shown in Figure 1, this step is performed in parallel with all the steps that should be performed in this phase. Use the knowledge elicitation techniques to capture the knowledge that is needed for performing the steps within this phase [17]. The knowledge should be extracted from the knowledge resources that have been recognized in the previous phase.
- Relation Analysis: Use Table 1 in order to investigate the impact of SA design decisions on EA design decisions (that have been made or are going to be made) and vice versa. The tick symbols show that the corresponded rows and columns affect each other.

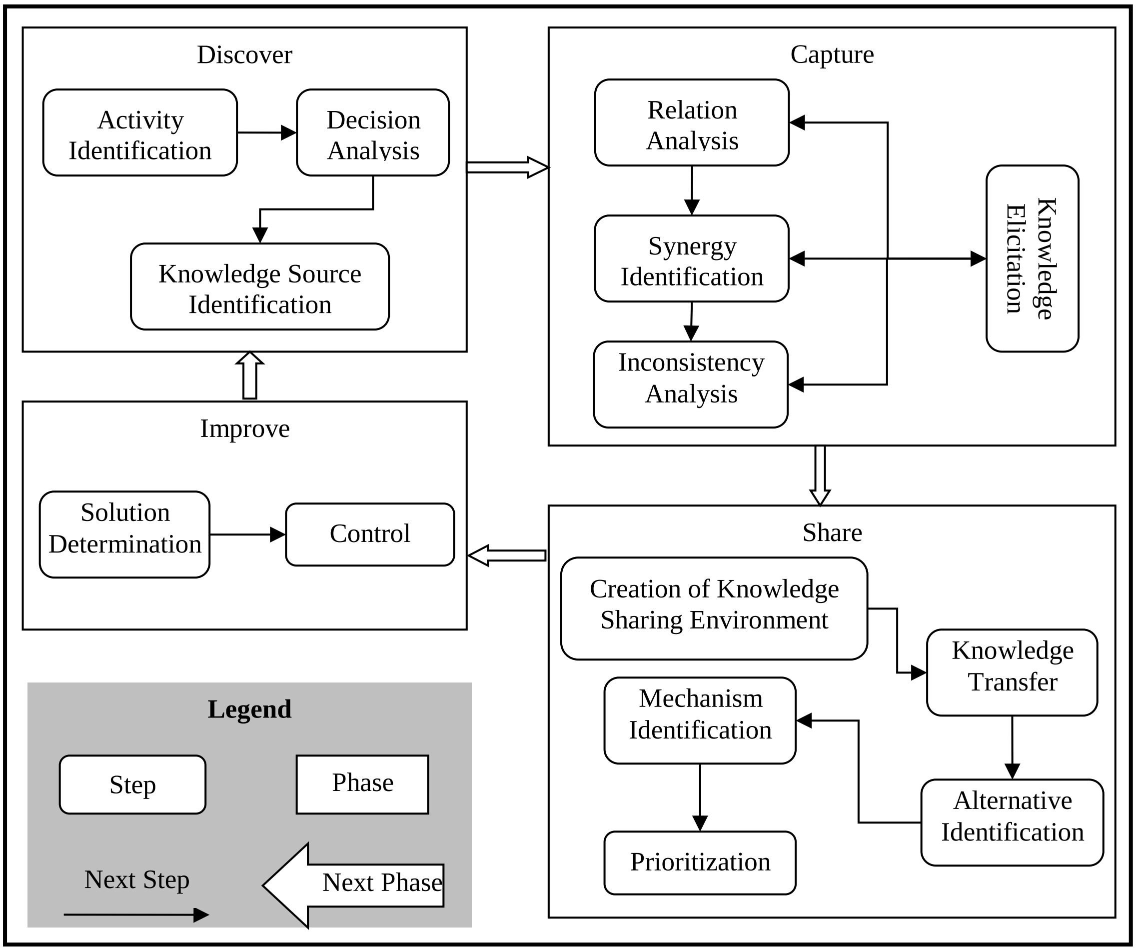

Figure 1. Proposed Aligning Framework

Table 1. Design Decision Aspects

SA Design Decision Aspect / **EA Design Decision Aspect**	Complexity of Architecture Description [15]	Satisfaction of Stakeholders' Requirements [15]	Understandability of Presentation Method [15]	Uncertainty of Designed Architecture [15]
Dependency between Decisions [24]			✓	
Rationality (Level of Certainty) [25]			✓	✓
Number of Alternative Decisions [25]		✓		
Amount of Attention to Business, Application, and Technology Layers [25]		✓	✓	✓
Predictability of Consequences of Decisions [25]	✓	✓		✓
Number of Positive and Negative Consequences of Already-Made Decisions [25]		✓		✓
Decision Traceability [25]	✓		✓	✓

- Synergy Identification: Identify the synergetic points between the SA and EA design decisions. Synergetic points are those activities that result in aligning decisions that have been analyzed in the previous step.
- Inconsistency Analysis: Identify the inconsistent points between SA and EA design decisions. Similar to synergy points, inconsistent points are the activities that cause misalignment between decisions.

3. Share: In this phase, knowledge is shared between SA and EA designers. In other words, the SA and EA designers share their knowledge in order to remove (or prevent) inconsistencies.

- Creation of Knowledge Sharing Environment: Develop appropriate environment in which the SA and EA designers can share their knowledge. Consider the cultural and the organizational factors that are mentioned in [19], [20], [21], and [22].
- Knowledge Transfer: Establish the right knowledge flow between SA and EA design decision makers. For this purpose, transfer the knowledge that has been gained in the previous phase. The knowledge sharing technique should be chosen based on the EA and SA designers' opinions. Try to motivate designers to share their knowledge [22].
- Alternative Identification: Identify all the activities that may be performed instead of the activities that were previously been identified.
- Mechanism Identification: Find all the mechanisms that may be used for performing the activities that were analyzed in the previous step.
- Prioritization: Investigate the advantages and disadvantages for selecting each of the identified activities and mechanisms, and prioritize them in a way that the activities with the highest priority cause less inconsistency and have the most advantages and the least disadvantages. Study the available heuristics for alignment (like [7] and [23]), and use them if appropriate.

4. Improve: The final goal (alignment) should be realized in this phase. The following steps are prescribed for this purpose:

- Solution Determination: Select the best solution among the solutions, prioritized in previous phase.
- Control: If a new decision is made, find the new issues that cause inconsistency. Follow the alignment steps, if needed. It should be noted that the available assessment models (such as the model proposed in [8]) can be reused for performing this step.

4. Evaluation of Proposed Framework

This section is targeted at evaluating the proposed framework. To this aim, an example of using the proposed framework is first provided, and then the available alignment approaches, reviewed in the second section, are compared with our proposed framework.

4.1 Example-Based Evaluation

In this subsection, the proposed framework is used for aligning a new SA design decision with a previously designed EA.

Dynamic software architecture is a kind of software architecture that may change during the maintenance phase, and thus particular description languages and techniques are required for applying the changes [26]. Suppose that an in-use EA has been designed based on the Zachman framework [27], and the SA changes continuously (because of the change in customers' requirements). As mentioned in [28], the Zachman framework does not support the maintenance phase. Thus, our proposed framework can help aligning the architectures as follows:

1. Discover: The following steps have been taken:

- These activities have been performed for changing the software architecture: Extract the services from component functionalities, and define the relations between services.
- It has been discovered that the mentioned activities increase the uncertainty because all the service would not be extracted from component functionalities; in the other words, services are not equal to component functionalities.
- The following knowledge resources have been discovered: SA designer, SA design documents.

2. Capture: Results of performing this phase are as follows:

- The following tacit knowledge contents were elicited: The reason behind the relation between components, results of analyzing different types of relations between services, the reason behind the way of identifying the services, and the knowledge about the relation between component functionalities and services. Furthermore, the following explicit knowledge contents have been identified: Designed component-based architecture, designed service-oriented architecture, and the post-mortem documents which incorporate the lessons learnt about the architecture that has previously been designed.
- The following issues have been discovered as the items that are affected by the new design decision: The rationality level, the level of predictability for the consequences of decisions, the number of positive and negative consequences for already-made decisions, the traceability of decisions, and the relation between business, application, and technology aspects of previously designed architecture.
- The reusability aspect of consistency (between business process model and the previously designed architecture) is expected to be maintained because both component-based and service-oriented architectures support reusability.
- Some inconsistencies may occur as follows: The concept of "service" is different from "component functionality". For example, the services may be coarse-grained while the component functionality is fine-grained and vice versa. Furthermore, all the component functionalities may not be directly mapped to a service. As a result, the business plan that has been designed (throughout the EA design process) may be inconsistent with the new functionalities that are now mapped to services.

3. Share: Results of performing this phase are as follows:

- Both SA and EA designers should be motivated for sharing their knowledge. For this purpose, an unofficial session is held and the advantages and goals of sharing the designers' knowledge are discussed. The designers are also asked to talk about their worries and opinions. In this step, the managers should try to alleviate the worries and provide an unofficial environment (like a coffee shop) for sharing knowledge.

- The technologies, which facilitate sharing designers' knowledge, should be recognized and prepared.
- Using the knowledge that has been shared in previous steps, the following activities are prescribed to be performed instead of the activities that have first been proposed: Extract the services from the system requirements, Refine the services, and continuously check the consistency with both the existing business plan, and the in-use business process workflows.
- As an alternative mechanism, using the available tools that support automatic definition of services has been discovered.
- The advantage of directly extracting the services is accuracy, and the advantage of extracting services from components is speediness. Also, automating part of the extraction process helps speeding the service identification process. Thus, the activities can be prioritized based on two criteria: speed, and accuracy.

4. Improve: In order to improve the alignment status, the designers should integrate the recognized activities. Meaning that, at first, some services are extracted directly while some are extracted indirectly. Then, the designers compare and integrate these results.

4.2. Comparison-Based Evaluation

In this subsection, the proposed framework is compared with a number of available alignment approaches, which were briefly reviewed in the background study section. To do this, a number of criteria have first been extracted based on the previously conducted research in this area (Table 2), and then the proposed framework has been compared with other research efforts in this area (Table 3). The evaluation criteria may be satisfied in three levels, as follows:

"+" Symbol: Means that the criterion is satisfied.
"*" Symbol: Refers to partially satisfied criteria.
"-" Symbol: Indicates that the criterion is not satisfied.

As shown in Table 3, the proposed framework satisfies most of the criteria. The reasons behind assigning the satisfaction values are sequentially provided below:

1. The proposed framework provides a high-level method for aligning architectures; thus it has a methodological approach.
2. The framework guides both SA and EA designers in regard to consider the special characteristics of the domain for which the architectures are (have been) designed. In other words, sharing the designers' knowledge helps satisfying this criterion.
3. Similar to the second criterion, this criterion is also satisfied due to active involvement of designers to whom the alignment issues have been conceded.
4. This framework has not been tested through case study; however, it has been proposed based on practically evaluated methods.
5. Analyzing the decisions (and the relation between them) supports this criterion.
6. Active involvement of designers helps satisfying this criterion. The designers are actively involved to share their knowledge.
7. The stages that analyze the synergy and inconsistency support satisfying this criterion.
8. Providing appropriate environment for knowledge sharing and also identifying the alternative activities and mechanisms support prevention of misunderstanding.

5. Conclusion and Future Works

The process aspect of aligning EA and SA has poorly been considered in the area of architecture alignment. In this research, we have proposed a high-level framework for aligning an existing (or new designed) EA of an organization with existing (or new designed) SAs. KM process has been used as the main frame of our proposed framework because the alignment process can be seen as a KM process that guides making architectural design decisions in a way that alignment is guaranteed. The SA and EA designers can take the proposed steps in the framework in order to find the issues that should be aligned. Then, they can make appropriate decisions for eliminating these issues.

It should be noted that the proposed framework is not detailed enough, and it should be configured by designers. In fact, the detail are dependent on the domain for which the framework is aimed. As an example, there exists various knowledge sharing techniques but selecting the best technique depends on the organizational characteristics (like culture); meaning that the techniques cannot be prescribed in advance. This research can be continued through proposing context-specific methods which are instantiated from this framework. It should be noted that various case studies should be performed to evaluate alignment methods in detail.

Table 2. Evaluation Criteria

ID	Criterion	Description
1	Considering the methodological aspect of SA [4]	The aligning approach should not ignore the methodological aspect of SA
2	Attention to domain issues [29]	The domain area in which the alignment is applied, affects the alignment method.
3	Distinguishing alignment situations [30]	The alignment should be guaranteed according to the situation at hand. The architectures should be optimally adapted.
4	Practicality [31]	The proposed method should be evaluated in practice.
5	Considering the relation with other aspects of alignment [32	The relation between architecture, governance, strategy, and communication issues should be considered.
6	Considering the human and process aspects [33]	The alignment method should be comprehensive and should attend to different aspects of alignment and the relations.
7	Being aware about alignment maturity [34]	Both the current and the target alignment level should be considered.
8	Paying attention to identifying alignment pivot and preventing misunderstanding [35]	The main goal should be determined. It should be specified that the goal is to align IT and business, or it is to use the business for IT alignment (or vice versa).

Table 3. Criteria-Based Evaluation

Approach \ Criterion ID	1	2	3	4	5	6	7	8
Component-Based	*	-	-	*	-	-	-	-
Evaluation-Based	*	-	-	*	-	-	-	-
Process-Based	+	-	-	*	+	+	-	-
Integration-Based	*	*	-	*	+	*	-	-
Model-Based	-	-	-	*	-	-	-	-
Domain-Based	-	+	-	+	-	-	-	-
Proposed Framework	+	+	+	-	+	+	+	+

References

[1] D. Avison, J. Jones, P. Powell, and D. Wilson, "Using and validating the strategic alignment model," *J. Strateg. Inf. Syst.*, vol. 13, no. 3, pp. 223–246, 2004.

[2] A. Wegmann, G. Regev, I. Rychkova, L.-S. Lê, and P. Julia, "Business and IT alignment with SEAM for enterprise architecture," in *Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International*, 2007, pp. 111–111.

[3] P. P. Tallon, K. L. Kraemer, and V. Gurbaxani, "Executives' perceptions of the business value of information technology: a process-oriented approach," *J. Manag. Inf. Syst.*, vol. 16, no. 4, pp. 145–173, 2000.

[4] D. Garlan, "Software architecture: a roadmap," in *Proceedings of the Conference on the Future of Software Engineering*, 2000, pp. 91–101.

[5] N. Anquetil, K. M. de Oliveira, K. D. de Sousa, and M. G. B. Dias, "Software maintenance seen as a knowledge management issue," *Inf. Softw. Technol.*, vol. 49, no. 5, pp. 515–529, 2007.

[6] S. Buckl, F. Matthes, and C. M. Schweda, "Future research topics in enterprise architecture management–a knowledge management perspective," in *Service-oriented computing. ICSOC/ServiceWave 2009 workshops*, 2010, pp. 1–11.

[7] C. M. Pereira and P. Sousa, "Enterprise architecture: business and IT alignment," in *Proceedings of the 2005 ACM symposium on Applied computing*, 2005, pp. 1344–1345.

[8] B. Van Der Raadt, J. F. Hoorn, and H. Van Vliet, "Alignment and maturity are siblings in architecture assessment," in *International Conference on Advanced Information Systems Engineering*, 2005, pp. 357–371.

[9] P. Van Eck, H. Blanken, and R. Wieringa, "Project GRAAL: Towards operational architecture alignment," *Int. J. Coop. Inf. Syst.*, vol. 13, no. 03, pp. 235–255, 2004.

[10] T. Clark, B. S. Barn, and S. Oussena, "A method for enterprise architecture alignment," in *Working Conference on Practice-Driven Research on Enterprise Transformation*, 2012, pp. 48–76.

[11] A. Jansen and J. Bosch, "Software architecture as a set of architectural design decisions," in *5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05)*, 2005, pp. 109–120.

[12] G. Riempp and S. Gieffers-Ankel, "Application portfolio management: a decision-oriented view of enterprise architecture," *Inf. Syst. E-Bus. Manag.*, vol. 5, no. 4, pp. 359–378, 2007.

[13] R. Nicolas, "Knowledge management impacts on decision making process," *J. Knowl. Manag.*, vol. 8, no. 1, pp. 20–31, 2004.

[14] M. Alavi and D. E. Leidner, "Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues," *MIS Q.*, vol. 25, no. 1, pp. 107–136, 2001.

[15] D. Falessi, G. Cantone, R. Kazman, and P. Kruchten, "Decision-making techniques for software architecture design: A comparative survey," *ACM Comput. Surv. CSUR*, vol. 43, no. 4, p. 33, 2011.

[16] L. Halawi, J. Aronson, and R. McCarthy, "Resource-based view of knowledge management for competitive advantage," *Electron. J. Knowl. Manag.*, vol. 3, no. 2, pp. 75–86, 2005.

[17] G. Schiuma, T. Gavrilova, and T. Andreeva, “Knowledge elicitation techniques in a knowledge management context,” *J. Knowl. Manag.*, vol. 16, no. 4, pp. 523–537, 2012.
[18] A. Riege, “Three-dozen knowledge-sharing barriers managers must consider,” *J. Knowl. Manag.*, vol. 9, no. 3, pp. 18–35, 2005.
[19] A. I. Al-Alawi, N. Y. Al-Marzooqi, and Y. F. Mohammed, “Organizational culture and knowledge sharing: critical success factors,” *J. Knowl. Manag.*, vol. 11, no. 2, pp. 22–42, 2007.
[20] T. Mooradian, B. Renzl, and K. Matzler, “Who trusts? Personality, trust and knowledge sharing,” *Manag. Learn.*, vol. 37, no. 4, pp. 523–540, 2006.
[21] J. S. Holste and D. Fields, “Trust and tacit knowledge sharing and use,” *J. Knowl. Manag.*, vol. 14, no. 1, pp. 128–140, 2010.
[22] N. R. Quigley, P. E. Tesluk, E. A. Locke, and K. M. Bartol, “A multilevel investigation of the motivational mechanisms underlying knowledge sharing and performance,” *Organ. Sci.*, vol. 18, no. 1, pp. 71–88, 2007.
[23] P. Sousa, C. Pereira, and J. Marques, “Enterprise architecture alignment heuristics,” *Microsoft Archit. J.*, vol. 4, pp. 34–39, 2004.
[24] G. Plataniotis, S. de Kinderen, and H. A. Proper, “Relating decisions in enterprise architecture using decision design graphs,” in *Enterprise Distributed Object Computing Conference (EDOC), 2013 17th IEEE International*, 2013, pp. 139–146.
[25] G. Plataniotis, S. de Kinderen, D. van der Linden, D. Greefhorst, and H. A. Proper, “An empirical evaluation of design decision concepts in enterprise architecture,” in *IFIP Working Conference on The Practice of Enterprise Modeling*, 2013, pp. 24–38.
[26] J. S. Bradbury, J. R. Cordy, J. Dingel, and M. Wermelinger, “A survey of self-management in dynamic software architecture specifications,” in *Proceedings of the 1st ACM SIGSOFT workshop on Self-managed systems*, 2004, pp. 28–33.
[27] J. Zachman, “The zachman framework for enterprise architecture,” *Zachman Int.*, vol. 79, 2002.
[28] R. Sessions, “Comparison of the top four enterprise architecture methodologies,” 2007.
[29] I. S. Aier and R. Winter, “Virtual decoupling for IT/business alignment–conceptual foundations, architecture design and implementation example,” *Bus. Inf. Syst. Eng.*, vol. 1, no. 2, pp. 150–163, 2009.
[30] J. Saat, U. Franke, R. Lagerstrom, and M. Ekstedt, “Enterprise architecture meta models for IT/business alignment situations,” in *Enterprise Distributed Object Computing Conference (EDOC), 2010 14th IEEE International*, 2010, pp. 14–23.
[31] R. J. Wieringa, P. A. T. van Eck, and D. Krukkert, “Architecture alignment,” *Enterp. Archit. Work Springer Berl.*, 2005.
[32] H.-M. Chen, “Towards service engineering: service orientation and business-IT alignment,” in *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, 2008, pp. 114–114.
[33] U. Seigerroth, “Enterprise modeling and enterprise architecture: the constituents of transformation and alignment
[34] J. Luftman, “Assessing business-IT alignment maturity,” *Strateg. Inf. Technol. Gov.*, vol. 4, p. 99, 2004.
[35] B. Fritscher and Y. Pigneur, “Business IT alignment from business model to enterprise architecture,” in *International Conference on Advanced Information Systems Engineering*, 2011, pp. 4–15.

Razieh Dehghani is a Ph.D. student at Sharif University of Technology, Tehran, Iran. She received the M.Sc. degree in Software Engineering from Sharif University of Technology, in 2014, and the B.Sc. degree in Computer Engineering (Software) from Bahonar University of Kerman, in 2012. Her current research interests include Information System Development Methodologies, Knowledge Management, Situational Method Engineering, and Software Evolution.

Email: email@ce.sharif.edu

Jafar Habibi is a faculty member at the Department of Computer Engineering at Sharif University of Technology and the managing director of Electronic Computing Machines Services. He is the supervisor of Sharif Robo-Cup Simulation Group. His research interests are mainly in the areas of computer engineering, simulation systems, MIS, DSS and evaluation of computer systems performance.

Email: jhabibi@sharif.edu

Paper Handling Data:

Submitted: 03/31/2018
Received in revised form: 05.28.2018
Accepted: 06.02.2018
Corresponding author: Dr. Jafar Habibi
Department of Computer Engineering, Sharif University of Technology, Tehran, Iran