

## SBISAF: A Service-Oriented Business and Information Systems Alignment Method

Aurelijus MORKEVIČIUS<sup>1</sup>, Saulius GUDAS<sup>2</sup>, Darius SILINGAS<sup>3</sup>

<sup>1</sup>*Kaunas University of Technology, Faculty of Informatics, Information Systems Department  
Studentų 50-313a, LT-51368 Kaunas, Lithuania*

<sup>2</sup>*Vilnius University, Kaunas Faculty of Humanities  
Muitinės 8, LT-44280 Kaunas, Lithuania*

<sup>3</sup>*Vytautas Magnus University, Faculty of Informatics, Department of Applied Informatics  
Vileikos 8-409, LT-44404 Kaunas, Lithuania  
e-mail: aurelijus.morkevicius@stud.ktu.lt, gudas@vukhf.lt, darius.silingas@gmail.com*

Received: February 2012; accepted: May 2013

**Abstract.** This paper presents a new approach for the business and information systems (IS) alignment consisting of a framework, metamodel, process, and tools for implementing it in practice. The purpose of the approach is to fill in the gap between the existing conceptual business and IS alignment frameworks and the empirical business and IS alignment methods. The suggested approach is based on the SOA, GRAAL, and enterprise modeling techniques such as TOGAF, DoDAF, and UPDM. The proposed approach is applied on four real world projects. Both the application results and the small example are provided to validate the suitability of the approach.

**Keywords:** enterprise architecture, enterprise modeling, business and IS alignment, SOA, service provisioning.

### 1. Introduction

Modern enterprises are establishing a process management as a must for successive planning and monitoring the performance of the processes. The constant monitoring of the process performance helps to identify bottle necks and their causes that are usually a workflow, software or hardware malfunction, or simply the lack of automation at a certain activity. It implies that the modern enterprise is facing a task of aligning its business needs with the software applications such as ERPs. The alignment of business and information systems (IS) is in contrast to what is often experienced in organizations: information systems and business professionals unable to bridge the gap between themselves. This gap generally results in expensive IS that do not provide expected return on investment (ROI). And the most common way of identifying bottle necks in business and IS alignment is the Enterprise Architecture (EA) which is a blueprint for how an organization achieves the current and future business objectives using information technologies (IT) (Dahalin *et al.*, 2011). For this reason, enterprise architecture programs are established in enterprises wherein the enterprise architects seek to align enterprise processes and infrastructure with their supporting information systems (Wegmann *et al.*, 2005).

The business and IS alignment has been studied in multiple distinct research areas such as an alignment via governance, alignment via communication, and alignment via architecture (Chen, 2008) where each is a subject of the different branch of science. Alignment via architecture is further classified into an alignment via software architecture and alignment via enterprise architecture. Alignment via enterprise architecture utilizes enterprise modeling, model analysis and design techniques and is in interest of our research.

Enterprise Architecture is a hot topic since 1987-ies when the first EA framework has been introduced by Zachman (1987). The EA was not very widely applied in practice due to lack of modeling languages and tools. Two decades have passed till the EA movement has been reinforced by the successful adoption of the Unified Modeling Language (UML) (OMG, 2007) and the Model-Driven Architecture (MDA) (OMG, 2003). There have been multiple attempts to apply Unified Modeling Language (UML) for enterprise architecture modeling (Dalgarno and Fowler, 2008), but many enterprise architects found it too complicated for solving their domain-specific problems (Silingas and Butleris, 2009a). However, the versatility of UML led to the appearance of multiple new modeling languages such as the Unified profile for MODAF and DoDAF (UPDM), System Modeling Language (SysML), Service Oriented Architecture Modeling Language (SoaML) etc (Morkevičius *et al.*, 2010). UML and its compatibility with its extensions allowed integrating different languages based models, thus enabling creation of large and versatile EA models in one repository (Silingas and Butleris, 2009b). It helped solving a wide range of problems: business transformation into knowledge-based business, business and IT alignment, the computerization of business management tasks etc (Gudas, 2009). UML has also been adopted for the other EA modeling techniques such as TOGAF (The Open Group, 2009), which evolved significantly over the last years.

In this paper we are focusing into a subset of business and IT alignment problem; the problem of optimally fitting business and information system architectures together – the business and IS alignment. The purpose of our research is to propose a new approach for business and IS alignment based on the existing conceptual frameworks, theories, and the research of the latest enterprise modeling techniques.

This paper is structured as follows: in Section 2, the related works are analyzed; in Section 3, the proposed approach is presented; in Section 4, experimental evaluation and application of the proposed approach is described; in Section 5, the achieved results, conclusions, and future work directions are indicated.

## 2. Related Work

There are a number of business and IT alignment via architecture methods. All of them are applicable to the business and IS alignment as well. One of the most well known methods is Guidelines Regarding Architecture Alignment (GRAAL). The GRAAL is a conceptual framework providing a collection of concepts and relations among them. It is based on four simple dimensions: (i) system aspects, (ii) system aggregation, (iii) systems process, and (iv) description levels, where the first three dimensions focuses on the

system analysis by its observable properties, composite structure, and life cycle and the fourth one concerns the level of granularity. The goal of GRAAL is to derive operational guidelines for aligning IT architecture with business architecture (Van Eck *et al.*, 2004).

GRAAL framework originated from another well known alignment framework of Henderson and Venkatraman (1999) distinguishing two alignment dimensions, the service provision and refinement. Other related frameworks are Zachman (1987) framework and the Kruchten's (1995) 4 + 1 model.

Systemic Enterprise Architecture Methodology (SEAM) business and IT alignment framework developed by Wegmann (2003) is grounded in General System Thinking (GST) (Weinberg, 1975), and living system theory (Miller, 1995). The two main SEAM concepts used to express the behavior and construction are functional and the organizational levels. Functional level represents the behavioral hierarchy and the organizational level represents the constructional hierarchy. Similar to GRALL it is a conceptual framework.

There are also works extending and applying GRALL (Zarvic and Wieringa, 2006) and SEAM (Wegmann *et al.*, 2005) frameworks. However none of them provides a method to evaluate business and IS alignment in the particular EA model. In other words, all frameworks described above are fully conceptual frameworks; they neither provide a process nor they are adapted using with the most popular enterprise modeling languages, frameworks, and methods differently than our indents are. The integrity with empirical enterprise modeling methods such as DoDAF, MODAF, NAF, TOGAF, and modeling languages such as UML, BPMN, Archimate, EMM (Gudas *et al.*, 2005), and UPDM provides traceability between business and application models; however they do not provide processes and tools of how to verify if business and IS is aligned. The study of empirical mostly used methods gave as a solid background for our research. As a result our proposed approach is based on the integrated metamodel developed on the basis of these methods. However, our goal is not a new modeling language. Our goal is an approach suitable to use in combination with the majority of empirical enterprise modeling techniques.

The closest method to ours is the BITAM (Chen *et al.*, 2005) method based on the SOA. BITAM uses a twelve-step process for managing, detecting and correcting the misalignment at the architecture level. The method is an integration of two distinct analysis areas: (i) business analysis and (ii) architecture analysis – for aligning elements in three layers of a business system: (i) business strategy, (ii) business architecture, and (iii) IT architecture.

Most of the methods described in this chapter are not considering the non-functional aspect of the enterprise model. Quantitative evaluation of business and IS is also not a part of empirical enterprise modeling techniques; however it is important for our research. The GRALL framework defines quality system aspect (Van Eck *et al.*, 2004) classified into for user and for developer sub aspects. The classification is close to the one used in ISO 9126 (ISO/IEC, 2004). ISO 9126 classifies software measurements into external metrics, internal metrics, and quality in use metrics. External and internal metrics are calculated by the software development team and the quality in use metrics are usually acquired

by the user. There are a number of other quantitative schemes to evaluate IS, business and their interrelationship. A detailed study of parameters applying to various enterprise model elements and their inter relationships has been introduced in Gustafsson *et al.* (2009).

Another method for quantitative model driven evaluation of the enterprise model has been proposed by Morkevicius *et al.* (2010). The method is based on the SysML parametric model defined in OMG (2008b) and is executable by using the majority of enterprise modeling tools. Versatility of this method allows it be used with any UML and SysML based domain specific modeling language.

### 3. Integrated Method for Business and IS Alignment

In this section we describe the framework, metamodel, process and related techniques that the proposed method is based on.

#### 3.1. SBISAF Framework

The framework defines four aspects of analysis for the business and IS alignment and utilizes the metamodel described in the next section of this paper.

Similarly to GRAAL, SBISAF defines two system aspects: a service and quality. The service aspect originates from the behavioral nature. In SBISAF it is defining the environment of the service including its provider, consumer and a set of behaviors required for the service realization, thus we are calling it functional. The quality aspect from its origin is structural. In SBISAF it defines quantitative characteristics of a service, thus we are calling it non-functional. Besides the two we also define vertical and horizontal aspects of the business and IS alignment. Vertical and horizontal dimensions in the business and IS alignment have been first mentioned by Labowitz and Rocansky (1997). In SBISAF we have identified vertical and horizontal aspects by analyzing the nowadays enterprise modeling techniques. We consider the verticality as the traces between two different abstraction layers in the enterprise model. In our case it is the traceability between business and IS architectures. The horizontality is considered as the relationships between service provider and the service consumer at the same level of abstraction. For instance a graphical user interface (GUI) component directly accessed by the user and the web service providing the business logic. By not having a detail service contract defined, derivation of the vertical business and IS alignment is not possible.

We are defining four combinations of business and IS alignment aspects: vertical functional, horizontal functional, vertical non-functional and horizontal non-functional.

#### 3.2. SBISAF Metamodel

According to the identified aspects of business and IS, we have clearly separated concepts into belonging to the business architecture and belonging to the information system architecture. We have also identified that some of the concepts do not fit to any

of them. For this reason we have used a concept of solution architecture, which originated from the MODAF architecture framework, where it describes the combination of systems and human resources (together grouped into resource configurations) used to implement business scenarios. As we have also defined a resource configuration and human resource concept that do not fit to the contents of the IS architecture, we are using a solution architecture concept to make a clear separation between the information system architecture and its constructs and a human resource and resource configuration concepts.

Inspired by Morkevicius and Gudas (2012), the Open Group (2009), and OMG (2009), we have defined business service as a service provided by a participant, requested by another participant, and supported by zero or more application services. A participant in the metamodel is a logical business unit which can abstract any of the following: human resource, application component, resource configuration, organization, department etc. An application service is a service supporting one or more business services. Application service that does not support business service we treat as a redundant. An application component is an actor requesting one or more application services from another application component.

We are describing a resource configuration concept to group interacting actors human resource and application together. It is common, especially in business architectures to show that the organizational resource consumes business service. However we are considering it as a bad practice resulting into the inexecutable model. As we have defined a human resource, it is the organizational resource in the solution architecture used as an actor of the IS. It means that the human resource performs one or more tasks with the help of the IS. However it cannot directly consume the application service by himself. It implies a human resource requires a proxy application component to indirectly access the application service to complete its task. A resource configuration is used to group both proxy application components and human resources.

From the behavioral point of view the business service is realized by the business process and the application service is realized by the application function. To follow the same pattern the vertical trace is added between the business process and application function. The trace is called implements.

In addition to already defined concepts we are using a measurement concept, which defines non-functional performance characteristics of a service. We are defining both a business measurement and an application measurement. The business measurement is constrained to measure only the business service and the application measurement is constrained to measure only the application service. The connection between both is called influences, as business measurements are directly dependent from the measurements of information systems. Another addition is the service level agreement (SLA) concept that the service must conform to. Conformance to the SLA is derived from measurements of the service. If the measurement of the service value is in the limits defined by the SLA, service is considered to be conforming to that SLA.

Based on the observations made and the metamodel built (Fig. 1), we are further defining four aspects of the business and IS alignment in details.

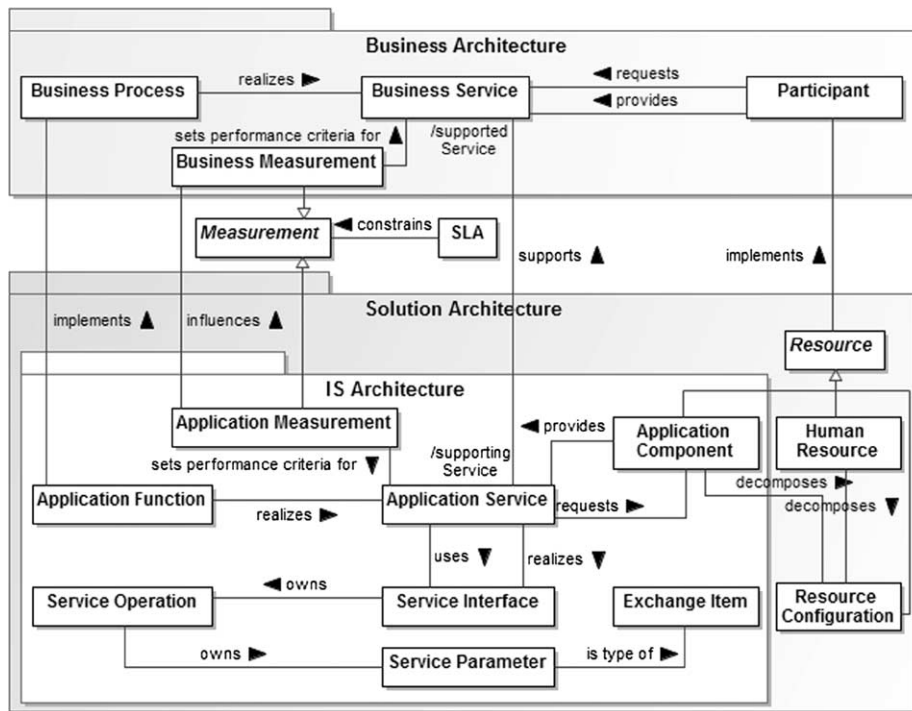


Fig. 1. SBISAF metamodel.

### 3.3. Functional Business and IS Alignment

In Morkevičius and Gudas (2012) a vertical functional business and IS alignment is defined as the alignment of business and application services where application service must serve business service, by supporting business processes performed; due to a business service would be delivered to a service consumer (Fig. 2a). A horizontal functional alignment of the business and IS is defined as a detailed interaction between two information systems where service requests from the requesting system have to be satisfied by one or more application services from the providing system (Fig. 2b).

To check if the vertical business and IS aspect is aligned, we are providing the list of business rules. The alignment is considered achieved only if all of the following rules are satisfied for every instance of a particular element:

1. An application service has to support one or more business services.
2. An application function has to implement one or more business processes.
3. An application component providing the application service has to implement one or more participants.
4. A Service provider (either a participant in case of business service or an application component in case of application service) should perform activities (either a business process in case of participant or an application function in case of application component) required for the service realization.

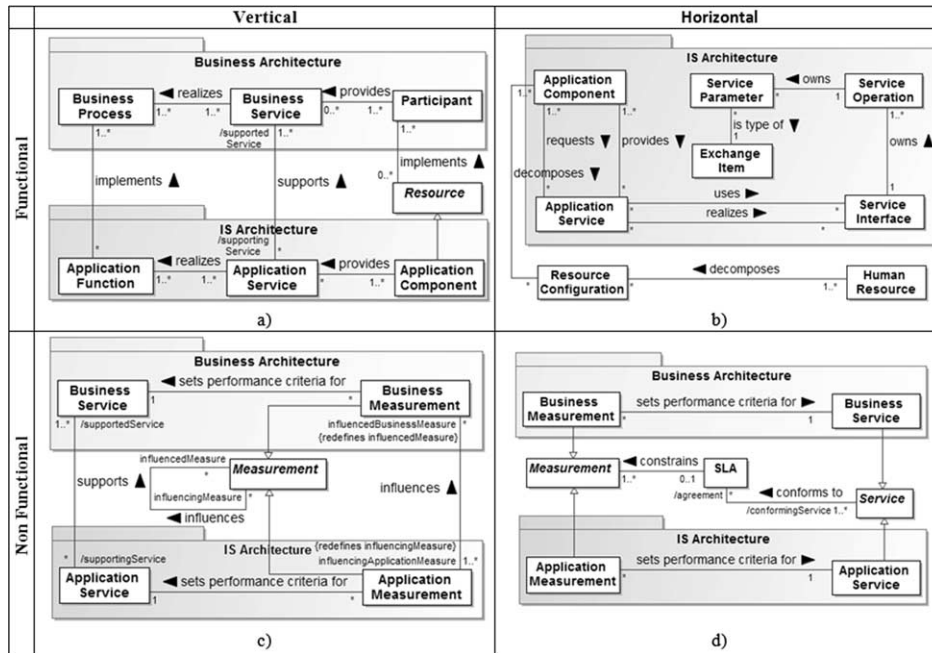


Fig. 2. Aspects of the business and IS alignment: (a) vertical functional, (b) horizontal functional, (c) vertical non-functional, (d) horizontal non-functional.

For the horizontal functional alignment to be achieved a set of rules needs to be satisfied:

1. All application service requests must be satisfied.
2. Requester and provider service interfaces must be compatible.
3. Resource configuration contents should include at least one application component if it is requesting or providing an application service.

### 3.4. Non-Functional Business and IS Alignment

Our proposed approach emphasizes the impact of IS parameters to the parameters of the business. For instance, we may think of measuring business process performance by using the control chart. At a point in time when the business process gets support from the IS its performance drops. As soon as employees learn using the IS the performance of the process starts rising quickly. We can see the direct impact of the IS system to the outcome of a process. The influence may be higher, lower or even critical if the failure of IS stops a process from delivering. It depends on many various circumstances within the enterprise.

The dependency of business measurements from the measurements of the IS we call a vertical non-functional business and IS alignment (Fig. 2c). It expands the service concept by adding performance parameters that we further simply call measurements. A single service can be measured by multiple measurements. We are separating two different

Table 1  
Requirements for each aspect of the business and IS alignment

Alignment level	Requirements
Vertical functional alignment	Each IS service must support at least one business service. Otherwise it is redundant.
Horizontal functional alignment	All service requests must be satisfied.
Vertical non-functional alignment	IS measurements, that the business measurements are influenced by, must be supplied.
Horizontal non-functional alignment	All SLAs for each of the services must be met.

kinds of measurements. One is the application measurement used to define the performance of the application service and the other is the business measurement used to define the performance of the business service. Each application measurement attached to the application service must influence at least one business measurement. The constraint applies that the measured business service would be supported by the measured application service.

To check if business and IS are vertically non-functionally aligned we are coming up with the rule: an application measurement has to influence at least one business measurement.

Values required achieving by the application service is specified in the SLA between the requester and the provider of a service. SLAs can be defined in both the business and the IS architectures. We consider that the achievement of the SLA at the application layer implies the achievement of the SLA at the business layer if the vertical non-functional alignment is achieved. The achievement of the SLA is the horizontal non-functional business and IS alignment (Fig. 2d). We are defining the rule to check if the model is horizontally non-functionally aligned: a measurement value should not exceed limits defined in the SLA.

To conclude, we have defined four aspects of business and IS alignment (Table 1).

### 3.5. A Process of Verifying Business and IS Alignment

The method is incomplete without the process of application and tools. We define the process of applying the proposed business and IS alignment method in this section.

A Functional and non-functional alignment aspects take the similar role in the IS engineering and reengineering processes; both need to be achieved before the development of the IS is started. We are recommending to perform functional and non-functional alignment checks as soon as the high-level system design (as a part of EA program) is developed. Tests for the horizontal non-functional alignment additionally requires test data and is much more difficult to perform especially before or at the early phases of the information system engineering process. Thus we are proposing to perform horizontal non-functional alignment tests as early as possible, but no later than the IS testing phase.



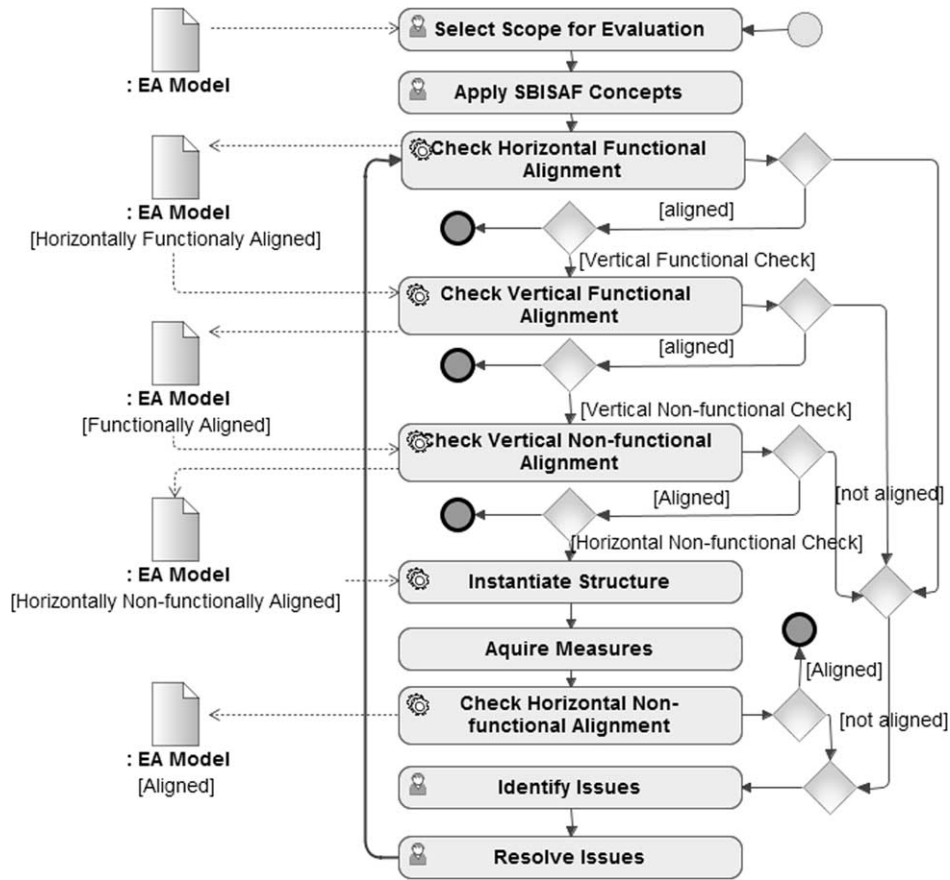


Fig. 3. Business and IS alignment process.

Looking wider to the enterprise information systems architecture where more than one system interacts to each other, both the functional and non-functional tests should be performed in every iteration of the EA development method to detect possible bottlenecks that may lead to the reengineering of one or more information systems.

The process of alignment of the business and IS starts from defining a model scope we want to check in order to reduce the complexity if checking the whole model at once. We also need to make sure that SBISAF concepts are well established before proceeding. This is important for both the functional and non-functional alignments; however it may be performed once before checking the functional business and IS alignment. When these two prerequisites are complete, we start with the evaluation of the horizontal functional alignment. The whole process is provided in the Fig. 3. If the model is horizontally functionally aligned we are proceeding with the vertical functional alignment check. As soon as we found issues making the model unaligned, we identify the source for each of the issues and resolve them. We restart the loop to check if all issues have been resolved and if no new issues are found. If there are no issues found we proceed to the verification of

the vertical non-functional alignment. As soon as we found issues making the model unaligned, we identify the source for each of the issues and resolve them. If there are no new issues found, the model is treated as a completely vertically non-functionally aligned and we continue with the horizontal non-functional alignment check which is done similarly. However, it includes two additional steps: (i) instantiation of the service structure and (ii) acquisition of measurement values.

By having multiple business and application services and measuring their alignment, we can come up with various statistics such as the percentage of overall alignment, percentage of functional alignment etc.

#### 4. Experimental Evaluation

As we have already defined the process, we need to define tools. We encourage using UML based standards for the enterprise modeling. Our choice has been influenced mostly by the number of UML based tools in the market. The decision of using UML led us to the realization of SBISAF in UML – we have developed UML profile for SBISAF. The verification of the alignment of the business and IS is performed by using object constrain language (OCL). For instance checking of whether the application function is implementing at least one business process is performed by executing the following OCL expression:

```
Context ApplicationFunction Inv: let i:Set (UML2_Metamodel
::Dependency) = self.clientDependency->select(e|e.
oclIsTypeOf(SBISAF_Profile::Implements)) in let p:Bag(
UML2_Metamodel::NamedElement) = i.supplier->select(e|e.
oclIsTypeOf(SBISAF_Profile::BusinessProcess)) in not p
->isEmpty()
```

A set of executable OCL expressions have been developed to check if all four aspects of the conceptual business and IS alignment are modeled correctly. Together with the SBISAF profile, both are forming the contents for the plug-in for the MagicDraw CASE tool. For the experimental evaluation we are using the implemented plug-in with the MagicDraw UPDM tool. Our model is based on the UPDM modeling standard with slight extensions required by SBISAF and not supported by the UPDM standard. For instance our extension profile allows classification of services and measures into business and IS domains and provides relationships between the domains such as supports and influences. For this reason we have used mapping between the SBISAF profile and the UPDM language defined in Morkevičius and Gudas (2012). Additionally we have expanded mapping table with non-functional business and IS alignment concepts (Table 2).

##### 4.1. Case Study

For the case study we have modified scenario of an e-shop defined in Morkevičius and Gudas (2012) to show the applicability of SBISAF. The *retail unit* in the e-shop consists

Table 2  
SBISAF profile to UPDM elements mapping

SBISAF element	UPDM element
Business Service	Service Access provided in the Operational Viewpoint
Application Service	Service Access provided in the Systems Viewpoint
Participant	Node Role typed by a Performer
Human Resource	Resource Role typed by a Person Type or Organization Type
Application Component	Resource Role typed by Software
Resource Configuration	Resource Role typed by a Capability Configuration
Consumes	Request port owned by a Performer or Software and typed by a Service Access
Provides	Service port owned by a Performer or Software and typed by a Service Access
Business Process implemented by Application Function	Implements
Participant implemented by Resource	Implements
Business Measurement	Property owned by Service Access provided in the Operational Viewpoint
Application Measurement	Property owned by Service Access provided in the Systems Viewpoint
Influences	–
SLA	Default value for the Property
Conforming Service	Default value for the Property owned by a Service Access
Service Architecture	Logical Architecture. Used to bind business measurements and application measurements to the SysML parametric model based calculations.

of the following participants: *sales unit*, *tech. support unit*, *supplier*, *supply unit*, and *post unit*. It also interacts with the external participant – *customer*. We have built the DoDAF operational resource flow description (OV-2) model to show the logical architecture of the e-shop *retail unit* (Fig. 4a). Shortly describing the model itself, it is used to show resource flows (in our case information flows only) between business units (participants) in the particular context. In this diagram the context is the *retail unit*.

Participants in the example model are providing and requesting business services. They are communicating through service channels. For example in Fig. 4b *customer* and *tech. support unit* are participants where the square on the border of *customer* is a request port requesting the *customer support business service* and the square on the border of *tech. support unit* is a service port providing the *requested customer support business service*. Participants are also exchanging information where *issue* goes to *tech. support unit* and *resolution* goes to *customer*.

This kind of modeling approach of services is defined in SoaML specification (OMG, 2008a). A subset of SoaML concepts are used in the UPDM modeling language (OMG, 2009).

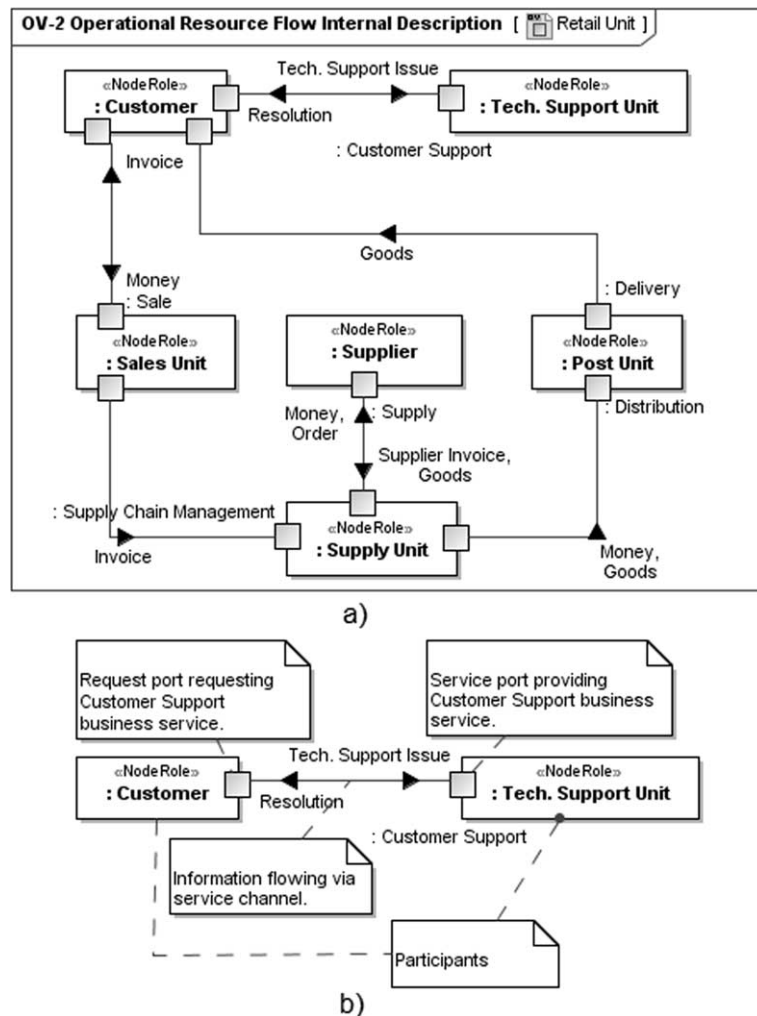


Fig. 4. (a) Architecture of retail business unit; (b) service channel.

We have also built DoDAF systems interface description (SV-1) model to show how human resources are interacting with the application components used in business (Fig. 6). In the systems layer we have used application services. We have also defined provided and required interfaces. For instance in Fig. 5 the *tech. support* application component is providing the *customer support* application service which realizes multiple service interfaces. A service interface provision is shown as a lollipop on the service port. There are also two request ports; one attached to the *tech. support staff* and the other to the *customer* human resource. A service request is shown as a socket on the request port. Both request ports are requesting the same service but if you take a closer look you will find requests for different service interfaces.

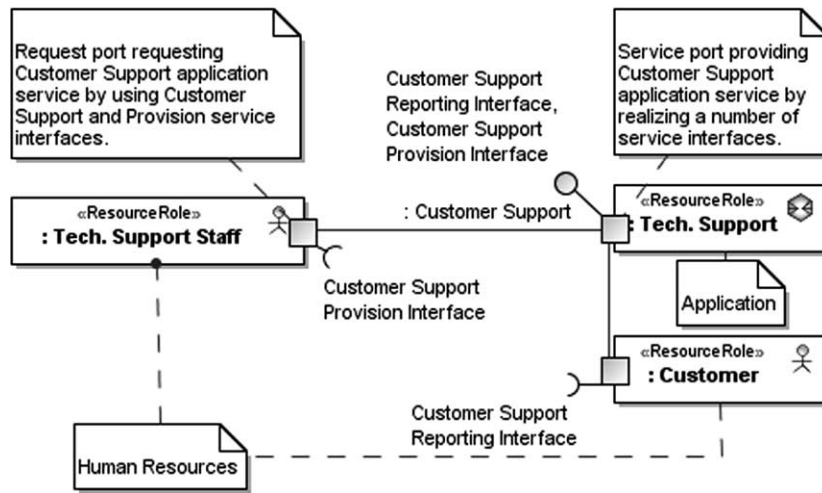


Fig. 5. Application service provision.

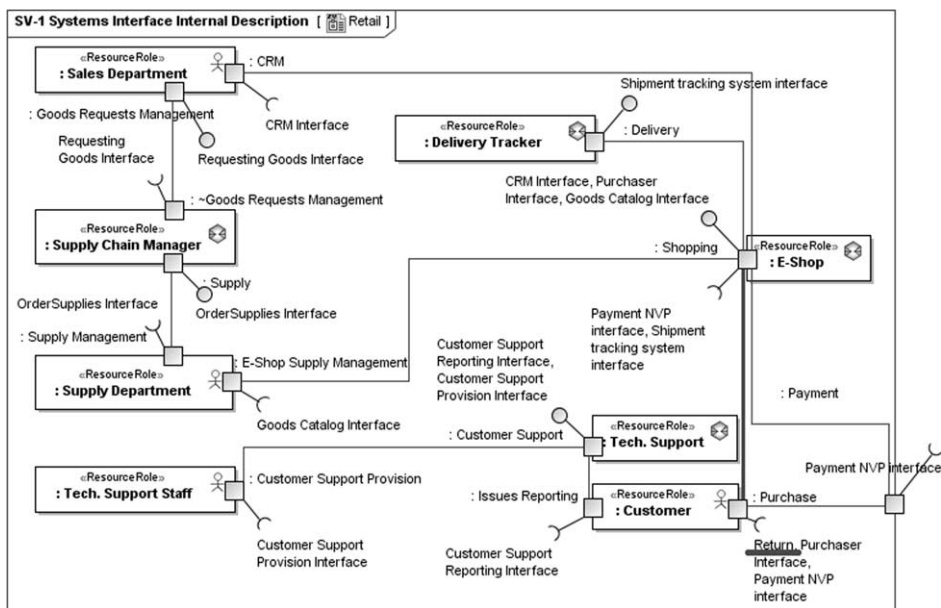


Fig. 6. Systems interface description model.

#### 4.2. Functional Business and IS Alignment Evaluation

First, we have checked the model by the horizontal functional alignment rule: all required interfaces are either provided in the particular context or delegated to the outside.

If looking to the systems interface description diagram (Fig. 6) note that the service channel between the *e-shop* and *customer* is highlighted. This is how the MagicDraw

	Customer Support	Delivery	Distribution	Sale	Supply	Supply Chain Management
SvcV-2	1	1	1	3	1	1
CRM				↗		
Customer Support	↗					
Delivery		↗	↗			
Goods Requests Management						↗
Payment				↗		
Purchase				↗		
Return						
Supply					↗	

Fig. 7. Business and application services alignment matrix.

tool indicates the failure of the executed OCL rule. It also provides a list of failures in the validation results pane (Fig. 8). In Fig. 6 the horizontal functional alignment rule failed because of incompatible interfaces at the service and request ports. In this particular case the *return* service requested by the *customer* is not provided by any of the information systems. By identifying this business and applications alignment gap in the e-shop enterprise we identified the model as not horizontally functionally aligned. As a resolution, we have integrated the *return* service to the model by providing all required relationships to make sure the model is horizontally functionally aligned.

Second, to make sure the business and IS is vertically functionally aligned we have validated the model by the following rule: each application service is supporting at least one business service.

According to the SBISAF metamodel, business services are supported by the application services. We have built a matrix of services to analyze if it is correct in our case (Fig. 7). We have identified that the *return* application service is not supporting any of the business services in the architecture. By performing detailed analysis it was indicated that there are no business processes realizing the *return service*. As a resolution, missing relationships have been added into the model. After making the model functionally aligned we are further continuing with the check of the non-functional business and IS alignment.

#### 4.3. Non-Functional Business and IS Alignment Evaluation

By having the functionally aligned architecture, first, we check the vertical non-functional alignment to make sure the relationships between business and application measurements are established and the horizontal non-functional alignment check can be performed. As a prerequisite for the check we have defined the availability measurement for each service

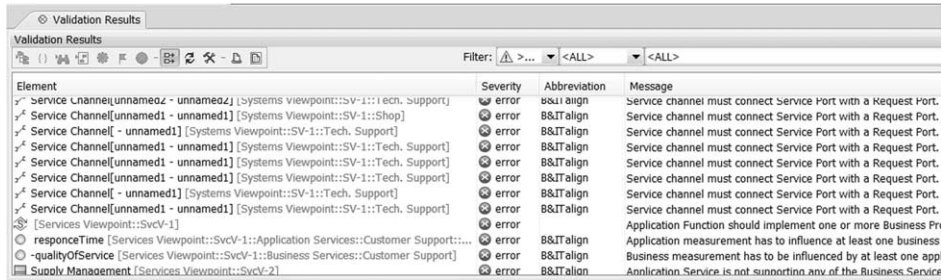


Fig. 8. Validation results pane in MagicDraw.

Purchase	Available		Unavailable	
	Available	Unavailable	Available	Unavailable
Payment	1	0	0	0
Unavailable	0	1	1	1

Fig. 9. Conditional probability matrix for the sale service.

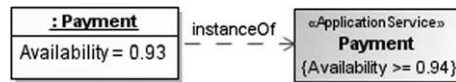


Fig. 10. Availability criteria conformance to the SLA.

in the e-shop enterprise. We have also defined the SLA value for each of the measurements. Conceptually the service is complying with the SLA, however in the model we are defining SLA values as the control limits for each of the measurements we want to check.

For the influence specification between measurements we are using an approach inspired by Jonson *et al.* (2007). The proposed approach uses the influence diagram that is a network used for modeling uncertain variables and decisions, consisting of a directed graph  $G = (N,A)$ , where the set of nodes is  $N$  and the set of arcs is  $A$ . For each node probability distributions are represented in conditional probability matrices. We consider each measurement as a node and each influences relationship as an arc. By creating matrices (Fig. 8) for all chance nodes Bayesian network is established as it is shown in Fig. 11. Please note that this is the instantiated service model so the influence relationships between measurements are not displayed. Instead the links between service instances are shown.

For calculations we use a model driven approach described in Morkevicius *et al.* (2010). We have built the SysML parametric diagram for the *retail* business service. We have also instantiated the service structure Fig. 11. By executing the parametric diagram using the MagicDraw tool, we have calculated that the *retail* service availability is 0.774% (Fig. 11). However the minimal expected value defined in the SLA is 0.97%. It means that the horizontal non-functional alignment in the scope of the retail architecture model is not achieved with the particular configuration of services.

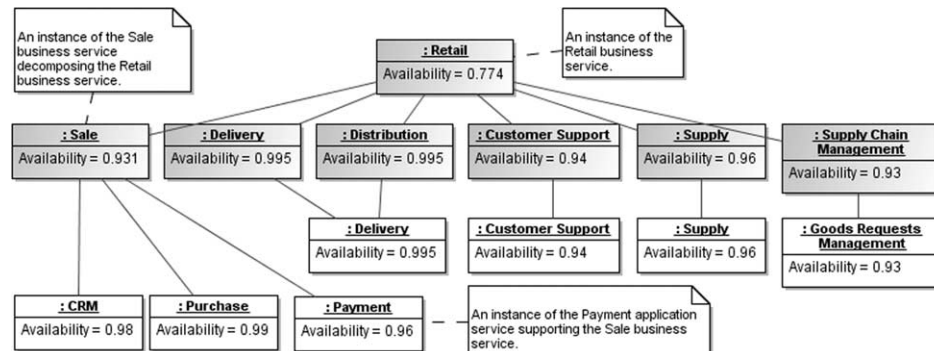


Fig. 11. Bayesian network of instances of business and application services.

A horizontal non-functional alignment is achieved if the availability values of services match the minimal expected values at a particular point in time. For instance if the availability of customer support service is expected to be not less than 0.94%; the value of 0.93% at a present point (Fig. 10) is less than expected. It means the SLA has been broken and the model is not horizontally non-functionally aligned.

Concluding the example provided, in the scope of the *retail* business service we have discovered multiple issues preventing the complete alignment of the business and IS. Functional alignment issues have been fixed, however non-functional alignment issues require changes in the SLA or the deeper investigation of their causes and the impact to the business. It is also recommended to perform more checks with alternative sources of data. Finally, all issues discovered need to be resolved and the model needs to be revalidated.

#### 4.4. Application of SBISAF

We have applied SBISAF on four real world industry projects A, B, C, and D. All projects are under the non-disclosure agreement, thus we are providing only the results of the business and IS alignment checks. All projects are UPDM projects. Three of them contain around 100,000 elements and one contains around 300,000 elements. The project A consists of 9 business services and 8 application services. The project B consists of 2 business services and 10 application services. Project C consists of 24 business services and 86 application services. Project D consists of 8 business and 5 application services. Below are the results containing the number and percentage of executions and violations of the rules for the SBISAF application on all four projects (Fig. 12).

Based on the results we can do early conclusions that the largest number of rules is executed during the horizontal non-functional alignment checks (45% of all executed rules). The least number of rules is executed during the vertical non-functional alignment checks (4% of all executed rules). In comparison the number of violations is highest for horizontal functional alignment checks (46% of all executed rules). The least number of violated rules is detected during the vertical non-functional alignment checks (3% combined).



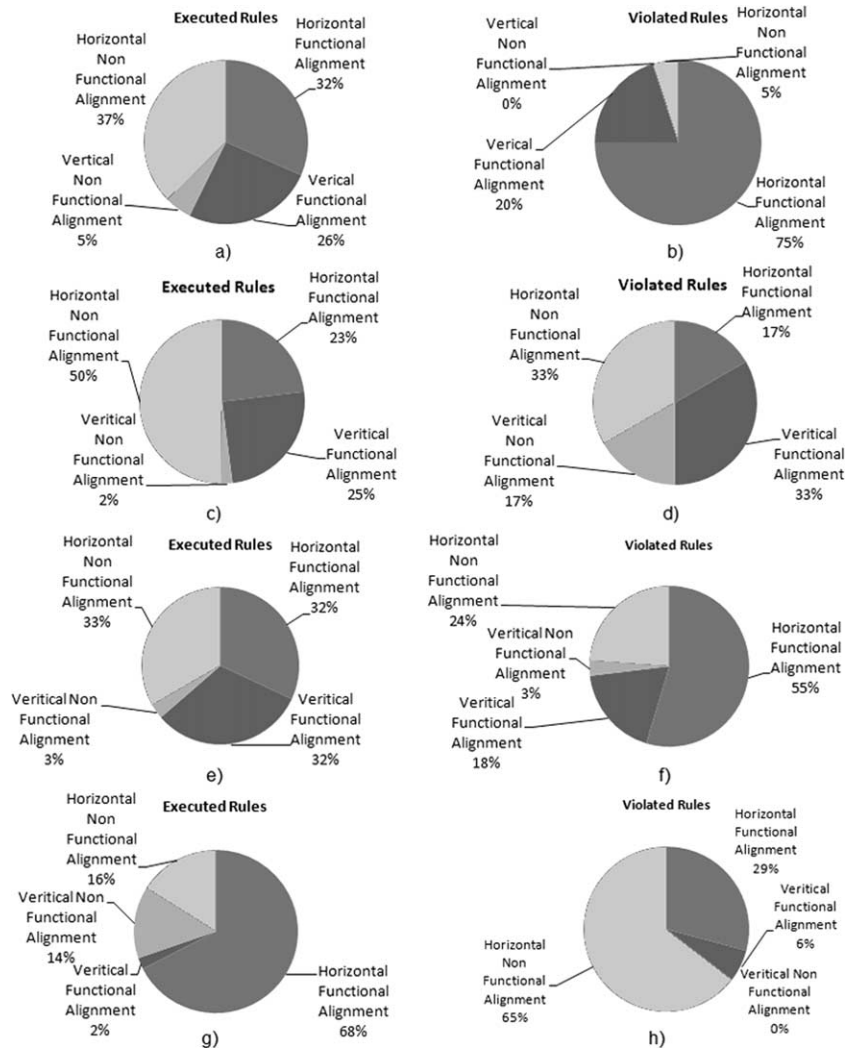


Fig. 12. (a) Executed rules in A; (b) violated rules in A; (c) executed rules in B; (d) violated rules in B; (e) executed rules in C; (f) violated rules in C; (g) executed rules in D; (h) violated rules in D.

The violated per executed rules ratio being the highest for the horizontal functional alignment aspect allows for identifying it as the most vulnerable aspect of the business and IS alignment. It means the violations in it are the most likely to be detected. The most solid aspect with the lowest violated per executed rules ratio is vertical non-functional. Comparing functional versus non functional alignment aspects, it is determined that 51% of all executed rules belong to the functional aspect. It is also determined that 59% of all violated rules belong to the functional aspect.

If comparing projects, according to violated per executed rules ratio, project C gets the highest score equal to 0.093. Scores for other projects are as follows: A = 0.015,

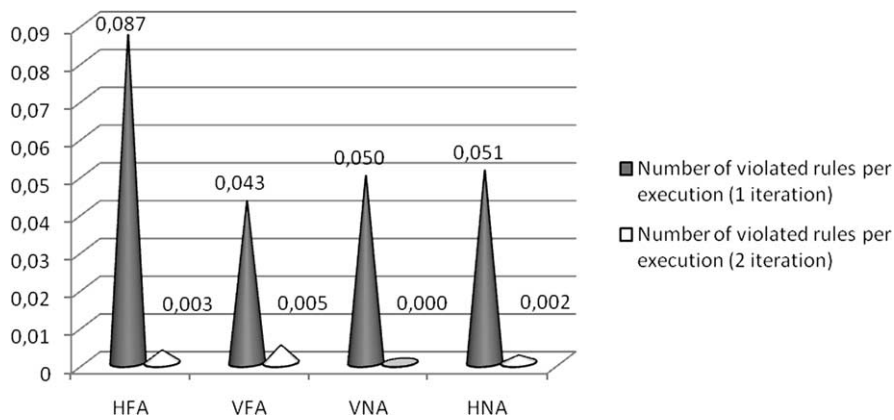


Fig. 13. Violated per executed rules ratio: first and second iterations.

$B = 0.007$ ,  $D = 0.016$ . The ratio equal to 0 shows that there are no violations detected and the model is completely aligned. The lower the value is the higher level of alignment is achieved in the enterprise model. The ratio value to be achieved is the subject to choose for a particular organization.

We have fixed the errors detected during the first iteration of business and IS alignment checks and performed the second iteration of checks. We have calculated the overall violated per executed rules ratio for each business and IS alignment aspect for both the first and the second iterations (Fig. 13). We have also calculated the overall violated per executed rules ratio during the second iteration (0.003) and compared it to the one calculated during the first iteration (0.061). The difference is the increment in 0.058 (0.06). This allows us to conclude that iterative application of SBISAF allows increasing the business and IS alignment level in the enterprise model until the targeted level is reached.

## 5. Conclusions and Future Works

The analysis of existing business and IS alignment methods revealed us that there are multiple different methods available. We have also identified that the majority of the existing methods are conceptual and thus do not provide the alignment process and are not applicable to the existing enterprise modeling techniques. To bridge this gap we have proposed the business and IS alignment method consisting of four aspects of alignment (framework), metamodel, and process. We have implemented the method using UML in the MagicDraw CASE tool. We have also showed its integrity with the existing and widely used UPDM enterprise modeling language.

SBISAF advantages over other methods are the alignment process, repeatability, metamodel, and applicability to multiple enterprise modeling techniques. As the disadvantages we have identified the lack of quantitative approach to measure business and IS alignment and the focus to only IS architecture by ignoring other domains of IT such as the technology infrastructure.

The proposed approach is already applied on four real world projects. While applying SBISAF we have determined the following:

1. The horizontal functional business and IS alignment aspect is the most vulnerable (failures are the most expected).
2. The horizontal non-functional business and IS alignment aspect is the most stable (failures are the least expected).
3. The second iteration in applying SBISAF revealed that we have managed to reduce the ratio of violated per executed rules by the average value of 0.06. This allows us to conclude that the iterative application of SBISAF allows increasing the business and IS alignment level in the enterprise model.
4. According to the experience of the enterprise architect or any other role solving business and IS misalignments, number of alignment iterations required reaching the desired level of business and alignment may vary.

According to the conclusions made we have indicated the following improvement areas for SBISAF: (i) propose modifications for existing enterprise modeling methods to provide views for the better business and IS alignment specification, (ii) describe the way to calculate the business and IS alignment index, (iii) apply method to even more real world EA projects.

## References

- Chen, H.M. (2008). Towards service engineering: service orientation and business-it alignment. In: *Proceedings of the 41st Hawaii International Conference on System Sciences*, pp. 114.
- Chen, H.M., Kazman, R., Garg, A. (2005). Managing misalignments between business and IT architectures: a BITAM approach. *Journal of Science of Computer Programming*, 57(1), 5–26.
- Dahalin, M.Z., Razak, A.R., Ibrahim, H., Yusop, I.N., Kasiran, K.M. (2011). An enterprise architecture methodology for business-it alignment: adopter and developer perspectives. *Communications of the IBIMA*.
- Dalgarno, M., Fowler, M. (2008). UML vs. domain-specific languages. *Methods and Tools*, 16(2), 2–8.
- Department of Defence (2009). *DoD Architecture Framework, Version 2.0, Volume 2: Architectural Data and Models Architect's Guide*. USA.
- Gudas, S. (2009). Enterprise knowledge modelling domains and aspects, *Technological and Economical Development of Economy*, 15(2), 281–293.
- Gudas, S., Lopata, A., Skersys, T. (2005). Approach to enterprise modelling for information systems engineering. *Informatica*, 16(2), 175–192.
- Gustafsson, P., Höök, D., Ericsson, E., Lilliesköld, J. (2009). Analyzing IT impact on organizational structure: a case study. In: *PICMET 2009 Proceedings*, Portland, Oregon, USA.
- Henderson, J.C., Venkatraman, N. (1999). Strategic alignment: leveraging information technology for transforming organizations. *IBM Systems Journal*, 38(2–3), 472–484.
- ISO/IEC. (2004). *ISO/IEC TR 9126-4: Software Engineering – Product Quality – Part 4: Quality In Use Metrics*. International Organization for Standardization – ISO, Geneva.
- Johnson, P., Lagerstrom, R., Narman, P., Simonsson, M. (2007). Extended influence diagrams for enterprise architecture analysis. *Information Systems Frontiers*, 9(2–3), 163–180.
- Kruchten, P. (1995). The 4 + 1 view model of architecture. *IEEE, Software*, 12(6), 42–50.
- Labowitz, G., Rosansky, V. (1997). *The Power of Alignment*. Wiley, New York.
- Lankhorst, M. (2009). *Enterprise Architecture at Work*. Springer. Berlin.
- Luftman, J., Mclean, E.R. (2004). Key issues for IT executives, *MIS Quarterly Executive*, 3, 89–104.
- Miller, J.G. (1995). *Living Systems*. Boulder, University of Colorado Press.
- Morkevicius, A., Gudas, S. (2011). Enterprise knowledge based software requirements elicitation. *Information Technology and Control*, 40(3), 181–190.

- Morkevičius, A., Gudas, S. (2012). An approach: a service-oriented functional business and IT alignment. *Information and Software Technologies Communications in Computer and Information Science*, 319, 162–175.
- Morkevičius, A., Gudas, S., Silingas, D. (2010). Model-driven quantitative performance analysis of UPDM-based enterprise architecture. In: *Proceedings of the 16th International Conference on Information and Software Technologies*, Kaunas, pp. 218–223.
- Nadler, D., Gerstein, M., Shaw, R. (1992). *Organizational Architecture: Designs for Changing Organizations*. Jossey-Bass, San Francisco.
- OMG (2003). *MDA Guide Version 1.0.1*. Needham, MA, USA.
- OMG (2007). *Unified Modeling Language (OMG UML) Infrastructure, v2.1.2*. Needham, MA, USA.
- OMG (2008). *Service Oriented Architecture Modeling Language (SoaML)*. Needham, MA, USA.
- OMG (2008). *Systems Modeling Language, Version 1.1*. Needham, MA, USA.
- OMG (2009). *Unified Profile for the Department of Defense Architecture Framework (DoDAF) and the Ministry of Defence Architecture Framework (MODAF)*. Needham, MA, USA.
- Silingas, D., Butleris, R. (2009). Towards customizing UML tools for enterprise architecture modeling. In: *Information Systems 2009: Proceedings of the IADIS International Conference*, Barcelona, Spain, pp. 25–27.
- Silingas, D., Butleris, R. (2009). Towards implementing a framework for modeling software requirements in MagicDraw UML. *Information Technology and Control*, 38(2), 153–164.
- The Open Group. (2009). *TOGAF, Version 9*. Van Haren, Zaltbommel.
- Van Eck, P., Blanken, H., Wieringa, R. (2004). Project GRAAL: towards operational architecture alignment. *International Journal of Cooperative Information Systems*, 13(3), 235–255.
- Wegmann, A. (2003). On the systemic enterprise architecture methodology (SEAM). In: *Proceedings of the 5th International Conference on Enterprise Information Systems*, pp. 483–490.
- Wegmann, A., Balabko, P., Le, L., Regev, G., Rychkova, I. (2005). A method and tool for business-IT alignment in enterprise architecture. *CAiSE'05 Forum*, Porto, Portugal.
- Weinberg, G.M. (1975). *An Introduction to General Systems Thinking*. Wiley, New York.
- Zachman, J.A. (1987). A framework for information systems architecture. *IBM Systems Journal*, 26(3), 276–292.
- Zarvic, N., Wieringa, R. (2006). An integrated enterprise architecture framework for business-IT alignment. In: *CAiSE'06*, Luxembourg, pp. 262–270.

**A. Morkevičius** received degree of master of business informatics in 2009. He is a doctoral student since 2009 at Kaunas University of Technology Information Systems Department (ISD). He is currently working as a solutions architect at No Magic Europe, a vendor of famous modeling platform MagicDraw. He is also one of the head architects for the UPDM standard in the Object Management Group (OMG). His research interests include enterprise architecture frameworks, enterprise modeling, enterprise model analysis and simulation, and business and IT alignment.

**S. Gudas** is a doctor habilitatus of computer sciences, associate professor of Information Systems Department of Kaunas University of Technology and Kaunas Faculty of Humanities of Vilnius University, dean of Kaunas Faculty of Humanities of Vilnius University. His research interests include computer-aided information systems engineering methods and tools, enterprise modelling for information systems engineering.

**D. Silingas** received his PhD in informatics from Vytautas Magnus University in 2005. He is currently working as a head of Solutions Department at No Magic Europe, a vendor of famous modeling platform MagicDraw. Darius is also a part-time associate professor at Vytautas Magnus University and visiting lecturer at ISM Executive School. His research interests include model-based system engineering and business process management.

## **SBISAF: Į paslaugas orientuota architektūra pagrįstas veiklos ir informacinių sistemų suderinamumo metodas**

Aurelijus MORKEVIČIUS, Saulius GUDAS, Darius ŠILINGAS

Straipsnyje pristatomas naujas veiklos ir informacinių sistemų suderinamumo metodas susidedantis iš karkaso, meta modelio, proceso ir įrankių metodo realizacijai. Metodo tikslas užpildyti spragą tarp konceptualių veiklos ir IS suderinamumo karkasų ir empirinių veiklos architektūros modeliavimo ir analizės metodų. Pristatomas metodas yra pagrįstas SOA, GRAAL ir veiklos architektūros modeliavimo metodais, tokiais kaip TOGAF, DoDAF ir UPDM. Metodas pritaikytas keturiems realaus pasaulio projektams. Straipsnyje pateikiami metodo taikymo rezultatai ir pavyzdys.