The Business Process Transformation Framework Implementation through Metamodel Extension

Vladimir Maruna[*], Tom Mercer[**], Igor Zečević[***], Branko Perišić[***], Petar Bjeljac[***]
[*] MD&Profy, Beograd, Serbia
[**] Value Chain Group, Lawrenceburg, Ohio, USA
[***] Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia
vladimir.maruna@mdprofy.com, tommercer@value-chain.org, {igor.zecevic, perisic, pbjeljac}@uns.ac.rs

Abstract — **The interoperability levels at which complex modeling frameworks may cooperate directly constraint the ease of bidirectional transformation of model artifacts. A particularly challenging approach is to extend targeting modeling framework with another, previously external modeling framework, through metamodel transformations. In this paper, the context and rationales for selecting the metamodel extending approach while embedding The Business Process Transformation Framework methodology implementation into SAP PowerDesigner modeling framework, is described. The paper is focused solely on the first release of the solution, whose main mission was to: define the scope of the concepts belonging to the fundamental methodology dimensions; support the conversion of Value Chain Group Business Process Transformation Framework content from the previous ValueScape implementation to SAP PowerDesigner metamodel extension; and further improvements and extensions of data migrated from the ValueScape to SAP PowerDesigner modeling environment.**

I. Introduction

The Business Process Transformation Framework (BPTF) is a business processes management methodology whose main mission is to improve the alignment between: business strategy, business processes, people, and systems necessary to support that strategy [1]. BPTF is a consequence of research activities focused on switching from the traditional, document based business process design, to the contemporary model based/model driven paradigm [2].

In order to build and maintain a specification of business process design or transformation activities, the model based approach utilizes the predefined library of formally specified building blocks, that may be assembled into arbitrary complex structures representing the information flow among business processes, regardless their scale or abstraction level [3].

The Value Chain Group (VCG) [4] has specified and developed the BPTF methodology with the corresponding Information Model (BPTF IM) [5] which serves as a foundation for all of the methodology concepts definitions. Different implementations of BPTF methodology are provided through mapping of the BPTF IM to: logical database design (for example the entity relationship model) [6]; object-oriented models [7, 8]; or metamodels that are accessible through the BPM/BPA modeling tools [9]. Depending on the implementation of metamodel it is possible to reuse the existing or create a completely new framework that enables: the scope definition, design, versioning and analysis of the digital forms of BPTF artifacts. Prior to the implementation of BPTF with SAP Power Designer integrated modeling environment (PD) [10], VCG has used ValueScape [11], an application that was specifically developed to support the utilization of BPTF IM.

The main focus in this paper is the VCGBPTFwPD integrated modeling environment, which implements and automates the selected use cases of the VCG BPTF methodology on top of SAP PowerDesigner integrated modeling tools. The concrete solution is implemented based on the standard extending mechanism of PD metamodel [12]. The PD Extension, as the central component of VCGBPTFwPD, implements BPTF IM using the ontology mapping approach.

The destination concepts, to which all of the BPTF IM source concepts are mapped, are specified in the form of an object oriented model (the metaclasses model). It represents a domain specific model that is translated into the necessary extensions of PD metamodel.

The main reasons for selecting SAP PowerDesigner among the variety of BPM supporting tools were:

- SAP PowerDesigner has already been in use by the targeted stakeholder groups;
- The existence of extending mechanisms within the PD methodology that enable resolving the potential structural conflicts that may arise when converting a domain model to the corresponding metamodel extensions;
- The existence of standard functions that support data import from external files into PD models, enabling easy and fast creation of the initial BPTF models;
- The existence of PD scripting languages that enable programmatic support for the implementation of selected BPTF methodology procedures.

As a consequence, several beneficial results emerged :

- The possibility of establishing, sharing, and upgrading underlining domain knowledge through modeling process and the utilization of a modeling framework;
- The efficient standardization of methods, activities, artifacts and other elements of VCG's BPTF;
- The possible integration of VCG BPTF methodology artifacts with other development frameworks used either at the enterprise level in context of arbitrary

Enterprise Architecture (EA) project [19], or within the frame of individual development projects.

The VCGBPTFwPD integrated modeling environment is planned to be incrementally implemented through several versions/releases. In this paper, the first incremental version is presented. Its main mission was to: define the scope of the concepts belonging to the fundamental dimension of BPTF IM building blocks; support the conversion of VCG BPTF content from ValueScape to PowerDesigner; and enable further improvements and extensions of data migrated from ValueScape to PowerDesigner integrated modeling environment.

II. The Business Process Transformation Framework Fundamentals

BPTF is expressed through three main dimensions, which are uniformly applicable to arbitrary enterprise (business) systems: Business Building Blocks, Value Chain Segmentation, and Continuous Improvement Programs [1]. Altogether they constitute the core transformation pool.

The Business Building Blocks dimension enables the declaration of basic executable/functional elements that constitute the BPTF. When combined together, they form (build): the Value Streams, the Process Flows, and the Activity Flows, together with the concepts that belong to the Value Chain Segments.

The Value Chain Segmentation dimension defines business processes in the form of patterns or prescriptions that belong to some particular industrial hierarchy, and are qualified for transformation enhancement in order to upgrade the enterprise performance. In BPTF, business processes are defined as linked collections of Value Streams, Process Flows, and Activity Flows.

The Continuous Improvement Programs dimension describes the business process improvement methods that attach the time dependent dynamical behavior to the Value Streams and Business Building Blocks [13, 14]. BPTF models, via the standard building block interconnections, express the value chain segments and their contents that constitute a particular enterprise state in the arbitrary instance of time.

A. The Business Process Transformation Framework Information Model (BPTF IM)

BPTF IM [5] is a VCG specification that describes all of the BPTF concepts, together with the assigned attributes (slots) and their associations. It allows the utilization of the existing BPM/BPA tools for capturing, designing, versioning, and analyzing of the BPTF artifacts in digital form. Different versions of BPTF implementation methodology can be achieved by mapping the BPTF IM into different metamodels.

BPTF IM includes 52 information concepts that are explicitly divided into three groups, which correspond to the previously described BPTF's dimensions. Each BPTF concept is defined in the tabular form through the application of ontology descriptors.

B. The Business Building Blocks

The Business Building Blocks dimension defines standard building blocks of BPTF methodology. Standard building blocks have a normalized definition - each block is defined exactly once in a uniform manner, regardless of the number of referenced value streams. According to that, building blocks can be organized hierarchically and mutually associated according to the current dictionaries ontology.

The BPTF methodology building blocks definitions and business processes vocabulary are developed through model driven approach, which is based on two reference models: the Value Reference Model (VRM) and the eXtensible Object Reference Model (XRM).

VRM is an industry independent, analytical work environment that establishes a classification scheme of business processes using the hierarchical levels and the process connections, which are established through their inputs/outputs [15]. Additionally, this model establishes the contextual links with best practices and metrics, which are referenced in the classification process, to determine the criticality level of company processes. VRM represents the frameworks' spot from which the design of industry-specific processes, i.e. XRMs, begins [15].

XRM represents the domain specific extensions of VRM models. XRM models are industry-specific dictionaries that have to be created by further decomposition of business processes defined in the VRM models. XRM establishes a domain-based view of a company and enables the analysis of industry processes that are specific for that industry branch. XRM also provides a structure for housing private/protected knowledge, while maintaining a standard vocabulary using VRM.

III. VCG BPTF with PowerDesigner

The VCGBPTFwPD integrated modeling environment is composed of five components: BPTF IM Extension, Libraries, Methodology, BPTF Administration, and The Framework Management.

The BPTF IM Extension Component implements VCG BPTF IM in the form of PD metamodel extensions. It allows the representation and management of all BPTF methodology dimensions and includes: necessary extensions of visual and descriptive notation; and procedures that automate certain BPTF methodology segments.

The Libraries Component encapsulates the set of Business Process Models that were created through the BPTF ValueScape tools content migration to the PowerDesigner integrated modeling environment. Migration is performed by expanding the PD metamodel with the developed BPTF IM Extension component. The model reflects the hierarchical structure of the existing VCG library.

The Methodology Component enables the application of the VCG Transformation Lifecycle Management (TLM) methodology [15] through PowerDesigner integrated modeling environment. Relaying on the BPTF IM Extension and Library components, it defines and automates the activities and procedures needed for business process transformation.

The BPTF Administration Component contains procedures that support the administration of user defined and directed development/upgrade of all BPTF element models. It includes the incremental definition and promotion of BPTF methodology changes.

The Management Component supports the VCGBPTFwPD version control mechanisms.

A. *The BPTF IM Object Oriented Model (OOM)*

BPTF IM Object Oriented Model (OOM) is an object representation of the domain specific model that defines syntax and semantics specific for BPTF concepts representation. All of the BPTF IM elements are mapped to a specified OOM. The ontology mapping process is performed according to the predefined set of rules presented in Table I.

The association rules, defined within the BPTF, are established in OOM through the Business Rules Objects [16]. For each rule there is a corresponding Business Rule Object that encapsulates textual description of the constraints applicable to the instances of attributes or associations.

TABLE I.
BPTF IM - PD OOM ONTOLOGY MAPPING

BPTF IM 1.0	Object Oriented Model
BPTF IM 1.0	Object Oriented Model
BPTF Concept	OOM Class
BPTF Concept Attribute	Corresponding Class as Class Attribute
BPTF Concept Description	Class Comment
BPTF Concept Attribute Description	Class Attribute Comment
BPTF Concept Attribute Format	Class Attribute Data Type
BPTF Concept Attribute Option	Class Attribute Standard Checks >> List of Values
BPTF Concept Association	Class Association

In Fig. 1, a high level package model of the OOM is presented. The high level packages follow the BPTF IM dimensions. The individual packages contain only those BPTF IM concepts that belong to the particular VCG BPTF dimension. Each package may reference concepts defined in other packages. The referenced concept appears in the form of a shortcut within the referring package.

Figure 1. The OOM high level package model

Following the prescribed mapping rules, concepts that belong to the Business Building Blocks dimension in BPTF IM release 1.0 are represented by the PD Class Diagram [16]. This class diagram is located in the Building Block Concepts Package of specified OOM. When creating the BPTF IM Extension, BPTF IM OOM was the initial reference to support the transformation of defined concepts to elements of PD metamodel.

B. *BPTF IM Extension*

1) *The General Characteristics of BPTF*

Every BPTF concept contains a unique attribute identifier (ID) and an attribute name that carries the semantics. For the purposes or implementing BPTF IM extensions, the root metaclasses of PD metamodel are used. Attribute ID is mapped to the meta attribute ObjectID of the IdentifiedObject metaclasses. The abstract metaclasses IdentifiedObject is inherited by all of the PD metamodel metaclasses. The attribute name is mapped to the meta attribute Name of NamedObject abstract metaclass. Meta attributes: Comment and Description from the NamedObject metaclass are used for further description of BPTF concepts.

2) *BPTF Category and BPTF SubCategory*

Three BPTF parra-categories and three subcategories belonging to the BPTF IM Extension are mapped to the metaclass ExtendedObject stereotypes with corresponding names: IO Category, IO SubCategory, Metric Category, Metric SubCategory, Practice Category, and Practice SubCategory (Fig. 2). Each of the BPTF subcategories is associated through the ExtendedAttribute to the appropriate BPTF category. This attribute is used in all subcategories to establish and maintain the "belongs to" relationship, which is directed from the subcategory to the corresponding category. The direction of this relation is changeable, meaning that the model user may choose the appropriate model category to which the subcategories belong.

The implemented category and subcategory concepts are not visually presented in diagrams (instances of these stereotypes do not have symbolic views). Their purpose is solely to classify other BPTF concepts, and they need to be present in that context only.

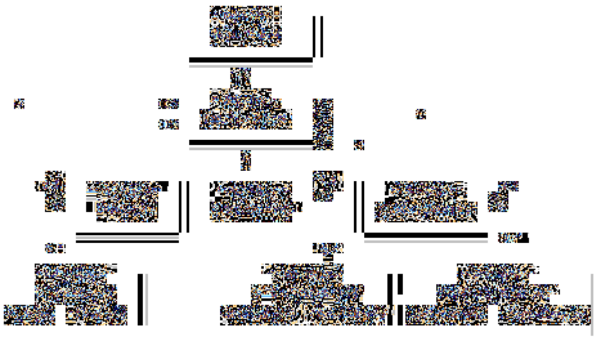

Figure 2. The BPTF Category/SubCategory concepts

3) *BPTF IO, Metric, Practice and Capability*

IO, Metric, Practice and Capability concepts are mapped to the corresponding stereotypes of Data metaclass, that are extended with necessary metaattributes in correspondence to the BPTF IM OOM. This metaclass enables the type of the information that has to be exchanged between business processes at the conceptual level of BPM [17]. It is focused on the information semantics rather than the technical aspects. Each instance of the above BPTF Data stereotypes (IO, Metric, and

Practice) may be associated with exactly one BPTF subcategory of the appropriate type (IO SubCategory, Metric SubCategory, and Practice SubCategory). ExtendedAttribute created in the Data stereotypes is used to establish and maintain the "belongs to" relationship between Data stereotype and the appropriate subcategory.

The additional extensions within BPTF IM Extension were necessary in order to implement the appropriate links between Practice and Capability concepts (Fig. 3). The association between these stereotypes is implemented using a standard PD concept, the ExtendedCollection. This collection, embedded in Practice stereotype, defines the standard set of functions (add, new, remove) that facilitate handling connections between one instance of Practice and a large number of Capability instances. The other side of this association is implemented through CalculatedCollection, enabling the selected Capability object to receive a list of all Practice objects previously associated through the ExtendedCollection.

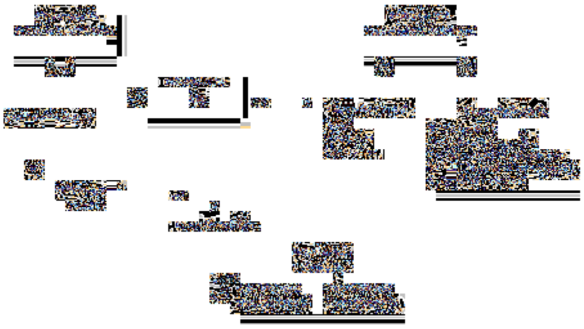

Figure 3. The Association of Practice and Capability Concepts

4) *BPTF Priority Dimension*

The PriorityDimension concept is mapped on to the ExtendedAttributeType metaclass of PD metamodel. Priority Dimension has a predefined set of values: Asset, Brand, Cost, Innovation, Reliability, Sustainability and Velocity. The reason for this mapping type is the relatively static nature of the PriorityDimension concept. Its values can only be changed at VCG level, while at the users level, the only possibility is to make the selection from a predefined list of values.

5) *BPTF VRM /XRM processes*

All of the VRM and XRM processes are implemented as corresponding stereotypes of Process metaclass: VRMLevelOne, VRMLevelTwo, VRMLevelThree, XRMLevelOne, XRMLevelTwo, and XRMLevelThree. Process metaclass is the specialization of the main activity of BPM and enables the creation of entities that deliver a set of services [17]. The standard process decomposition, supported by the PD BPM, completely supports the decomposition of VRM and XRM processes defined in BPTF IM. It is possible to decompose an arbitrary process through the hierarchical structure with corresponding dependences.

6) *Mapping the BPTF VRM to XRM Processes*

Within VCGBPTFwPD, an XRMLevelOne concept provides a standard solution for decomposition, synchronization, and association of VRM and XRM models. This concept is used to connect a generic BPTF environment (VRM) with its implementation, a specific, part (XRM). Processes on the first XRM level are created as synchronized copies of the VRM third level processes.

The associations between operational processes (level three of VRM) and a group of level one XRM processes in the BPTF Extension are established by adding the ExtendedAttribute to XRMLevelOne stereotype. The value of this attribute is automatically assigned in the initial phase of creating XRMLevelOne process instances by cloning the VRMLevelThree processes. XRMLevelOne object inherits all features of the original VRMLevelThree process, including comments, descriptions, and associations with IO, Metric, and Practice objects.

7) *The BPTF Rules*

In addition to the associated slots and associations concepts, BPTF IM 1.0 includes a set of rules that need to be met in order to obtain valid BPTF models. Within BPTF IM Extension, these rules are implemented by the Custom Check, a standard PD metamodel extension mechanism. Custom Check allows the definition of additional model content syntax and semantic validation rules [12]. Business logic, encapsulated in the Custom Check objects, is implemented with custom scripts written in VBScript language [18]. The editing and execution of these scripts are integral functions of PD. Each Custom Check is created as an extension of exactly one metaclass or a metamodel stereotype whose instances are validated.

IV. The Framework Libraries

The Library Component of VCGBPTFwPD is an ordered set of Business Process Models. These Models are the consequence of BPTF content migration from the ValueScape tool to the PowerDesigner modeling environment. The model architecture reflects the hierarchical structure of the existing VCG libraries.

The concepts, belonging to the Business Building Block dimension, are classified into two groups by the BPTF IM. The first group encapsulates categories and subcategories of: Input/Output, Metric, Practices, and Capabilities concepts.

The second group consists of concepts that are used to represent the value flow and /or value changes. This group is also implicitly divided into two categories of processes, which use the concepts defined in the first group. The first category covers the VRM processes, and other processes categories consist of XRM processes. In Fig. 4, a decomposition of the BPTF model is presented. It explicitly states that:

- Following the described internal organization of The Framework, it is obvious that There is exactly one BPTF Dictionary Model;
- There is exactly one BPTF Value Reference Model, which encapsulates generic and abstract experiences and the business process enhancement recommendations;
- There is exactly one BPTF eXtensible Reference Model per problem domain which contains domain specific recommendations, best practices and embedded knowledge. For the end user, XRM is the starting point from which the business process upgrade activities begin.

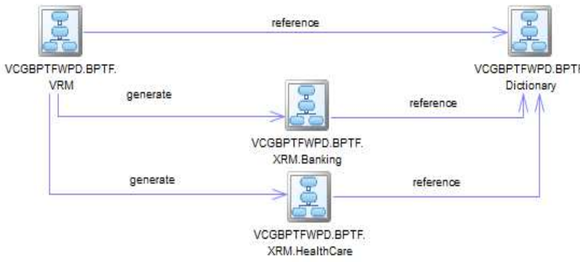

Figure 4. BPTF model decomposition

A. *Load and Transform Procedures*

Before the automation of ValueScape tool contents could start, it was necessary to implement the import procedures that are compliant with the standard PD data import mechanism (PD Import) [12]. In our case, PD Import is extended with two procedures:

- The „Load and Transform Association“ procedure, which supports building BPTF metamodel instances associations with 1: n cardinality for VRM/XRM processes, IO, Metric, Practice, Category and SubCategory objects; and n: m cardinality for Practice-Capability;
- The „Load and Transform Business Process" procedure structures the VRM and XRM processes in a tree hierarchy according to the descriptions of parent and corresponding collection of child processes.

B. *The Dictionary Model*

The Dictionary Model is implemented as a PD BPM with corresponding BPTF IM Extensions. The model contains all of the BPTF methodology element descriptions: Input/Output, Metric, Practice, Category and SubCategory classifiers, as well as the Capability concepts catalog. Objects from the Dictionary Model are created through previously described Load and Transform procedures, and are accessible outside the Dictionary Model. According to the principle of good localization embedded in BPTF concepts, the access to these objects is mitigated by the referencing mechanism represented by the inter-model and cross-model shortcuts. This allows an immediate automatic propagation of each modification to all of the referencing models.

C. *The Value Reference Model (VRM)*

The Value Reference Model is implemented as PD BPM with BPTF IM Extensions. The modeled processes are expressed in three hierarchical segments of VRM Industrial (Plan, Govern and Execute). The VRM32XRM1 Procedure

The VRM32XRM1 procedure generates first level XRM processes based on third level VRM processes, and supports the synchronization of corresponding VRMLevelThree-XRMLevelOne pares. It is implemented as a BPTF IM Extension procedure. In the preparation phase, The Framework user selects one of the possible XRMs that have to be synchronized with the corresponding VRM. In the execution phase, the procedure compares all of the three level VRMs with all of the first level processes of the selected XRM. If the corresponding processes differ, either in attribute values or in the associated objects, the XRMLevelOne process is automatically synchronized. Third level VRM processes without the corresponding copies of XRMLevelOne processes are automatically cloned at the first level of the XRM. The first level XRM processes that do not possess the originating process at the third level of VRM are removed.

D. *The eXtensible Reference Model (XRM)*

In VCGBPTFwPD, the initial eXtensible Reference Model, that contains the three level hierarchy of XRM processes decomposition, is implemented as PD BPM with BPTF IM Extension. It represents the starting point of domain specific technology needs of the particular user.

In the first phase, based on the already formed third level VRM process, the first level XRM's processes are created. For each VRMLevelThree, the VRM32XRM1 procedure has already created the copy of objects in the XRM, with the corresponding stereotype. In the second (content formation) phase, relying on standard PD Import from external files, the second and third level XRM processes are created. In the third phase, the relations between all levels of XRM processes are associated to the objects defined in the Dictionary Model. This is performed by the “Load and Transform Association” procedure. In the final phase of XRM model creation, with the support of the "Load and Transform Business Process", the complete XRM process tree is automatically generated.

E. *The Check Model*

In the model check phase, the validation of all defined constraints is performed at one place, thereby simplifying the use and reusing existing mechanisms to which the PD integrated modeling environment users are already familiar to. In addition to pre-defined rules (for example object name uniqueness), the list of rules may include user-defined rules that are established through the Custom Check object metamodel extensions.

All of the rules that are defined in the BPTF IM Extension through the Custom Check mechanism are appended to the existing ones, via embedded validity checks that exist within PD Business Process Model. The additional checks are available through the Check Model procedure for each extended BPM. The Check Model function is used for all of the syntax and semantic checks for all the automatically generated members of BPTF Libraries component.

V. Conclusions

In this paper, the VCGBPTFwPD, an integrated modeling environment that implements a selected subset of the Value Chain Group Business Process Transformation Framework (VCG BPTF) methodology use-cases via extending the SAP PowerDesigner (PD) integrated modeling tool metamodel, is presented.

VCG BPTF is a model based approach to design/transformation of business processes, which is specified by the corresponding Information Model (BPTF IM). It assumes that concepts and constraints that are described by BPTF IM may be mapped to the metamodel of existing BPM/BPA tools and extended to support the capturing, design, versioning and analysis of BPTF artifacts in digital form. The main component of the described solution is the BPTF IM Extension, implemented through the standard SAP PowerDesigner (PD) extending mechanisms.

The extension is implemented by applying a systematic approach to developing a UML profile based on a domain model or metamodel. In the first phase, an object oriented model (OOM) describing the mapping of BPTF ontology is created. The second phase included a systematic translation of all of specified OOM elements to the appropriate extensions of PD metamodel. The created BPTF IM Extension is applied to the Business Process Model to enable automated artifacts generation from the former VCG ValueScape to the newly created framework in order to form the initial library of BPTF models. All models that were created in such a way were validated over a set of user-defined syntax and semantic checks.

The first version of The Framework is currently in VCG BPTF Training and Certification program, assumed to gradually replace the existing ValueScape application afterwards. The results that we expect to obtain during the testing and certification phases will be used for further evaluation and improvements of the developed framework. The advanced version of The Framework is planned to enrich the BPTF IM Extension by additional mechanisms that would capture the rest of the BPTF IM's dimensions.

Acknowledgment

The paper origins from the results of VCG BPTF with Power Designer project launched by the Value Chain Group in cooperation with the MD&Profy Company and the personal participation of employees at the Computing and Control Department, Faculty of Technical Sciences, University of Novi Sad.

References

[1] T. Mercer, D. Groves and V. Drecun, “BPTF Framework, Part I”, Business Process Trends, September 2010, available at: http://www.bptrends.com/publicationfiles/09-14-10-ART-BPTF%20Mercer%20et%20al-Final.pdf [accessed on 29 January 2016].

[2] T. Bellinson, “The ERP software promise, Business Process Trends”, July 2009, available at: http://www.bptrends.com/publicationfiles/07-09-ART-The%20ERP%20Software%20Promise%20-Bellinson.doc-final.pdf [accessed on 29 January 2016].

[3] T. Mercer, D. Groves and V. Drecun, “Part II - BPTF Architectural Overview”, Business Process Trends, Octobar 2010, available at: http://www.bptrends.com/bpt/wp-content/publicationfiles/THREE%2010-05-10-ART-BPTF%20Framework-Part%202-final1.pdf [accessed on 29 January 2016].

[4] Value Chain Group, http://www.value-chain.org [accessed on 29 January 2016].

[5] Value Chain Group, Bussines Process Transformation Framework (BPTF) Information Model - Building Blocks Segment, http://dx.doi.org/10.13140/RG.2.1.4252.4325.

[6] V. C. Storey, “Relational database design based on the Entity-Relationship model”, Data & knowledge engineering, 7(1), 47-83, (1991), http://dx.doi.org/10.1016/0169-023X(91)90033-T.

[7] J. Rumbaugh, M. Blaha, W. Premerlani., F. Eddy, W.E. Lorensen, “Object-oriented modeling and design”, (Vol. 199, No. 1). Englewood Cliffs: Prentice-hall (1991).

[8] P. Shoval, S. Shiran, “Entity-relationship and object-oriented data modeling - an experimental comparison of design quality”, Data & Knowledge Engineering, 21(3), 297-315 (1997), http://dx.doi.org/10.1016/S0169-023X(97)88935-5.

[9] M. J. Blechar, J. Sinur, “Magic quadrant for business process analysis tools”, Gartner RAS Core Research Note G, 148777 (2008), available at: http://expertise.com.br/eventos/cafe/Gartner_BPA.pdf [accessed on 29 January 2016].

[10] SAP PowerDesigner DataArchitect, available at: http://www.sap.com/pc/tech/database/software/powerdesigner-data-architect [accessed on 29 January 2016].

[11] R. Burris, R. Howard, “The Bussines Process Transformation Framework, Part II”, Business Process Trends (May 2010), available at: http://www.bptrends.com/publicationfiles/05-10-ART-BPTFramework-Part2-Burris&Howard.doc1.pdf [accessed on 29 January 2016].

[12] SAP, Customizing and extending power designer. PowerDesigner 16.5, available at: http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc38628.1650/doc/pdf/customizing_powerdesigner.pdf [accessed on 29 January 2016].

[13] T. Mercer, D. Groves, V. Drecun, “Part III – Practical BPTF Application”, Business Process Trends, November 2010, available at: http://www.bptrends.com/publicationfiles/FOUR%2011-02-10-ART-BPTF%20Framework--Part%203-Mercer%20et%20al%20--final1.pdf [accessed on 29 January 2016].

[14] G. W. Brown, “Value chains, value streams, value nets, and value delivery chains”, Business Process Trends, April 2009, available at: http://www.ww.bptrends.com/publicationfiles/FOUR%2004-009-ART-Value%20Chains-Brown.pdf [accessed on 29 January 2016].

[15] Value Chain Group, Process Transformation Framework, available at: http://www.omgwiki.org/SAM/lib/exe/fetch.php?id=sustainability_meetings&cache=cache&media=vcg_framework_green_09-09.pdf [accessed on 29 January 2016].

[16] SAP, Object-Oriented Modeling. PowerDesigner 16.5. http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc38086.1650/doc/pdf/object_oriented_modeling.pdf [accessed on 29 January 2016].

[17] SAP, Business Process Modeling. PowerDesigner 16.5. http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc38088.1653/doc/pdf/business_process_modeling.pdf [accessed on 29 January 2016].

[18] A. Kingsley-Hughes and D. Read, “VBScript programmer's reference”, John Wiley & Sons, 2004.

[19] H. Peyret, G. Leganza, K. Smillie and M. An, “The Forrester Wave™: Business Process Analysis”, EA Tools, And IT Planning, Q1 (2009), available at: http://www.bps.org.ua/pub/forrester09.pdf [accessed on 29 January 2016].

[20] American Productivity and Quality Center: "Process classification framework", available at: www.apqc.org/free/framework.htm [accessed on 29 January 2016].