REST SOA Orchestration and BPM Platforms

Octavian DOSPINESCU, Cătălin STRÎMBEI, Roxana-Marina STRAINU,
Alexandra NISTOR
Faculty of Economics and Business Administration, AL.I.Cuza University, Iasi
doctav@uaic.ro, linus@uaic.ro, roxana.strainu@gmail.com,
alexandra.anichitoaei@yahoo.com

BPM initiatives and SOA approaches emerged and developed as distinct efforts although they could be perceived as the two sides of the same coin in the context of online business and information systems. Thus, inherently, their modeling perspectives met (or collided) and generated some frameworks and languages like SOAML or SOAMF. Starting from these standards and initiatives we will propose a methodological approach that could lead to a feasible implementation of web service integration in a formal Business Process context.

Keywords*: BPM, SOA, SoaML, Service Oriented Architecture, RESTful Web Services*

1 SOA MODELING

According to [1] the Service Oriented Architecture (SOA) concept is based on the principle of developing reusable business service and building applications instead of building monolithic applications in silos.

SOA has the potential to bridge the gap between business and technology, improve reuse of existing applications and interoperability with new ones. Some authors [2] consider that software services are the building blocks for SOA, and they can be composed to provide a more coarse grained functionality and to automate business processes.

A modeling language is an artificial language that can be used to define system, information in the structure defined by the rules of the system. Rules are used to interpret the association of components in the system. There are two types of modeling languages

- *Graphical Modeling*: it used a diagrammatic approach to represent the system. There are symbols and lines that connect the symbols which is used to display the relationship between the systems. Notations are used to define the constraints in the system.
- *Textual Modeling*: it uses the keywords or special characters with parameters to represent certain information in natural language which can be used by computers for processing [3]. Expressions are used to link the relationship among the text.

Service Oriented Modeling (SOM) is the first step to model services of SOA. According to the specialists [3], the approach for modeling SOA with the viewpoint from services is required. Various methods for service-oriented modeling have been proposed to apply SOA in industrial and business environments by many researchers [4]. They can be seen in Table 1 below:

Table 1. Features of SOA Modeling Methods [4]

Method/year	**Summary of features**
SOA-RM (2006)	No modeling language for service-oriented modeling Used to understand the essence of SOA and core concept Most abstract model relating to SOA in the lowest level Service identification and service composition are not supported
SOA-RFA (2012)	Used to understand the important features of SOA No service identification, but a particular resource in SOA Using UML2 to visualize structured and behavioral architecture conception of SOA
SOA ontology	Corresponding core concept of SOA

DOI: 10.12948/issn14531305/21.1.2017.03

	Using OWL as a modeling language and UML to illustrate classes and properties in SOA modeling No service identification Potential contribution to model-driven SOA implementation
SOMF (2008)	Using specialized modeling notations and support SoaML Service identification in granularity level and relationship between services Covering analysis, design and architecture but doesn't support transformation of existing assets to SOA
PIM4SOA (2007)	Developing a meta-model for SOA Covering essential aspects for SOA (service, process, information and QoS – Quality of services) Supported by WSDL and XSD (web modeling languages) and UML
SoaML (2009)	Using UML Focusing in the basic service modeling concept QoS specification cannot be supported Defining business process choreography and bridging business process to SOA in details
SOMA (2004)	Using SoaML QoS can be supported Service identification, service specification and realization Widely used in industry and business area

Starting from this table we decided, based on the reviewed literature, to pay attention only to specific modeling approaches, which are more common and used today.

1.1. SOAML

UML is used for general purpose modeling of SOA and create visual models. The services were modeled as objects and general purpose of the design was made by it. But when the degree of complexity increased it was not that efficient. This is how SoaML (Service oriented architecture Modeling Language) emerged: as a UML profile and meta-model for the modeling and design of services within a service-oriented architecture [3].

SoaML was created to support the following features:

- Indentifying services and their requirements.
- Defining service consumers and providers as well as the service itself.
- Specifying message pattern and protocols for the services.
- Different Service classification schemes.
- Integration and extension with OMG meta-models and BPMN.

The Object Management Group (OMG) proposed SoaML in 2009 for representing SOA artifacts using Unified Modeling Language (UML) as a core-modeling standard. Moreover, a meta-model and a UML profile are provided in SoaML for the specification and design of service to SOA (meta-model for modeling the requirement for a service and UML for specifying services) [4].

1.2. SOMF

Service Oriented Modeling Framework (SOMF) is a model driven engineering methodology which focuses on business process, and the IT is part of the lifecycle. It can also be used at standalone designing platform with other modeling languages such as UML, BPMN, or SoaML to enrich the language syntax. It can also set software development priorities during life cycle stages, and enhance the overall implementation view of the system [3].

The SOMF offers specialized modeling notation to help model, analysis and identification services that is proposed by Bell. It provides a formal method of service

DOI: 10.12948/issn14531305/21.1.2017.03

identification at different levels of abstraction including meta-model concept and specific notation [4].

1.3. SOMA

SOMA is a modeling technique for developing and building SOA-based systems proposed by IBM in 2004. SOMA activities focuses include: service identification (discovering candidate service and interaction between them), service specification (making decision for exposing services), and service realization (capturing service realization). The main focus of SOMA method is on the service, service components and flows with emphasizing on reusing services [4].

Service-Oriented Modeling and Architecture provides a detailed analysis and design method for the identification, specification and realization of services needed in an SOA, validating every step of the design phase so you are more likely to end up with a fully integrated, flexible and responsive infrastructure for your business. SOMA separates the business logic of a process from its technology platform.

Business process modeling, while as critical for IT as for business, does not typically reach into the IT architecture and implementation domains. Service-Oriented Modeling and Architecture was created to help bridge this gap. Referencing the business analysis, the modeling approach provides a set of techniques to help ensure that the necessary services are provided and used [5].

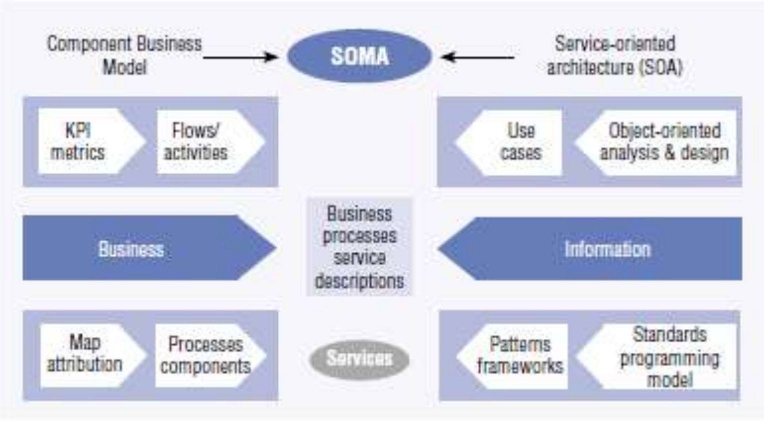


Fig. 1. SOMA integration within the business [5]

As we can see from Figure 1, SOMA is the bridge between Component Business Model and SOA. This helps identifying a proper modeling technique which confirms the conclusions from our reviewed papers: the combination of SOMA and SoaML allows the practitioners to model services in detailed and systematic way [4].

Starting from the specifications presented in table 1 and using the more detailed descriptions from the previous chapter we can conclude that UML and BPM are used as the base for the development of the new modeling techniques for SOA. Going further, we can notice that SoaML, SOMF and SOMA are close to what a SOA modeling needs, but none of them is complete. This is why combinations of these are used in modeling for SOA.

2. SOA & BPM Modeling

SOA is “a software structure for building applications that implement business processes or services, using a set of black-box type software components loosely coupled and orchestrated so that they deliver a well-defined level of services” [6]. In the context of SOA, a *service* is a function performed by an application. A function is coded only once and then reused wherever it is needed. This is a difference from traditional systems, where business functions are hard coded into applications.

DOI: 10.12948/issn14531305/21.1.2017.03

BPM is a methodology, as well a collection of tools that enables enterprises to specify step-by step business processes. Business process management (BPM) addresses how organizations can identify, model, develop, deploy, and manage their business processes, including processes that involve IT systems and human interaction.[1] Business process generally involve running a set of activities and using data to achieve a business objective. Control of process execution, of business data and activities is carried out by different services within the organization [7].

The need for further research arises from the fact that there are gaps in the integration of BPM with SOA, and that there are few studies on the assessment environments of tools and methodologies for modeling of process/cases for collaborative environments, comparative analysis of their use on different types organizations and their use in service oriented environments [7].

Figure 2 shows the relationship between BPM and SOA. As shown in the diagram, BPM does the modeling, simulation, and redesign of processes. SOA infrastructure orchestrates business processes and mediates service providers. Services are exposed, to be used in various processes. Service changes should not impact processes. Process changes reuse various services as needed.

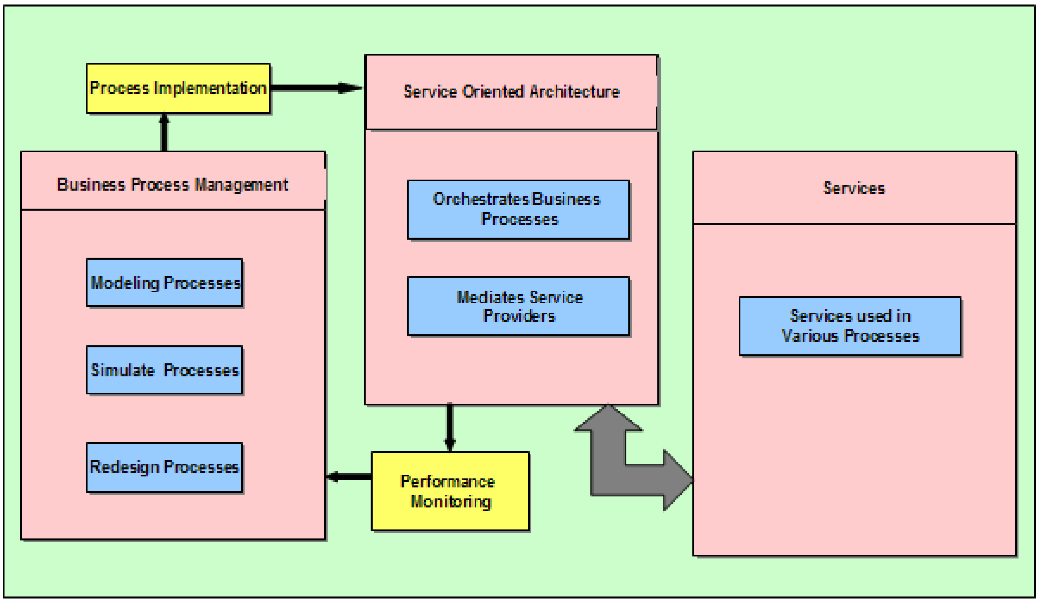

Process Implementation
Service Oriented Architecture
Business Process Management
Orchestrates Business Processes
Services
Modeling Processes
Mediates Service Providers
Services used in Various Processes
Simulate Processes
Redesign Processes
Performance Monitoring

Fig. 2. Relation between BPM and SOA [1]

While the researchers tried to find a way to combine the two concepts by presenting the advantages of this combination, one point was very clear. This may not be completed represented without the help of business rules which represent the multitude of policies, procedures or definitions. These may be external rules, coming from legal regulations that must be observed by all organizations acting in a certain field, or internal rules which define the organization's business politics and aim to ensure competitive advantages in the market [8].

In this context we introduced the three variables we need for this topic: SOA, BPM and BR. The three variables work together in a way which may be seen in Figure 3.

DOI: 10.12948/issn14531305/21.1.2017.03

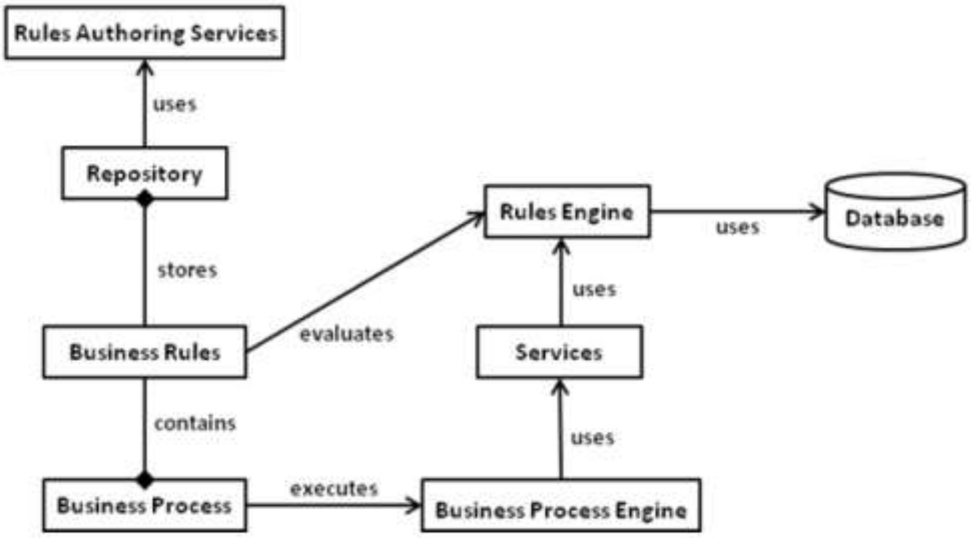


Fig. 3. Usage of business rules and process engines in SOA [1]

The need for business rules was obvious because the processes, the main components of BPM, highly depend on rules. The rules decide how processes communicate with each other or how an activity must be performed. An activity, the atomic part of a business process is implemented (using a SOA approach) with the help of an application which provides services to that activity and to the containing process also.

The literature review also revealed some important differences between the two concepts presented in table 2 below.

Table 2. Differences between SOA and BPM

Characteristics	SOA	BPM
Focus	Creating flexible architecture	Optimizing the way tasks are performed
Perspective	Technological perspective	Business perspective
Approach	bottom-up	Top-down
Work with	Services	Processes
Metaphor	Relational Database Management System	E/R diagram

Researchers tried to identify ways to combine SOA modeling with BPMN and to obtain a framework or a method to automate this combined modeling approach. In these researches, MINERVA (Model drIveN & sErvice oRiented framework for the continuous business process improVement & relAted tools) emerged, a framework for process improvement based on business process lifecycle. Starting from the modeling of business processes in the Business Process Modeling Notation (BPMN), MINERVA automatically obtains from these models service design elements expressed in UML, specifically several diagrams in Service Oriented Architecture Modeling Language (SoaML), to the execution of processes expressed in WS-BPEL or XPDL in a suitable process engine [9]. These steps are presented in figure 4 below.

DOI: 10.12948/issn14531305/21.1.2017.03

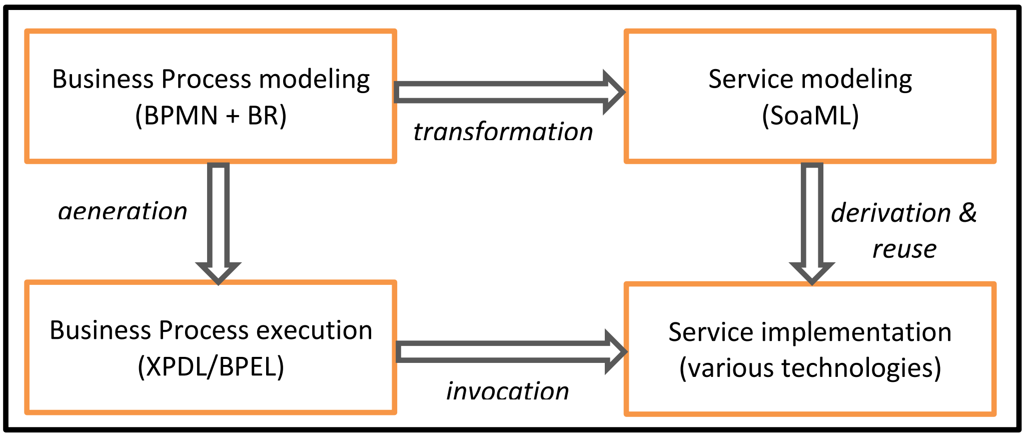


Fig. 4. Business processes and services relationships in MINERVA [9]

The concept of service-oriented business processes was proposed, to take advantage of the principles of abstractness, specialization and separation of service oriented architecture. The proposed concept consists in: business activities and business data. The management of business process activities can be grouped in: design, modeling, deployment, execution, monitoring and optimization. If each BPM activity is treated in terms of service-orientation, the proposal presented in Figure 4 below becomes feasible.

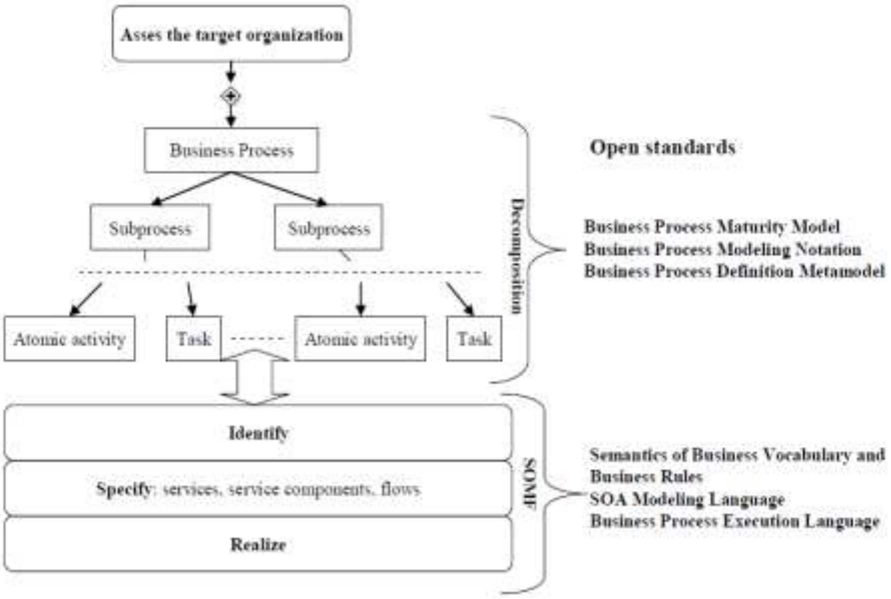

Asses the target organization
Business Process
Subprocess
Subprocess
Atomic activity
Task
Atomic activity
Task
Decomposition
Open standards
Business Process Maturity Model
Business Process Modeling Notation
Business Process Definition Metamodel
Identify
Specify: services, service components, flows
Realize
SOMF
Semantics of Business Vocabulary and Business Rules
SOA Modeling Language
Business Process Execution Language

Fig. 5. Management of services oriented business process [7]

BPMN can be used for modeling business processes in accordance with SOA. BPMN has service modeling capability which allows a participant to use service activities to send messages to other participants. BPMN and Unified Modeling Language (UML) have very different approaches. UML offers an object-oriented approach to the modeling of applications, while BPMN takes a process-centric approach. In the process implementation phase, the key technology of SOA is BPEL (Business Process Execution Language). This language minimizes the semantic gap between the process model and the actual execution code. BPEL enables business processes to be executed directly [7]. SoaML provides a higher-level of abstraction, and more complete service modeling capability that BPMN could benefit from. BPMN can be combined with SoaML for more comprehensive service construction and consumption modeling.

DOI: 10.12948/issn14531305/21.1.2017.03

As a summary we could establish some interconnections between the terms presented here. The business processes are managed using BPM and modeled using BPMN. Each process is decomposed at atomic level in activities which are implemented as functions within applications. As we could see from our definition of SOA, a *service* is a function performed by an application. This means that each activity can be seen as a service. Services are what SOA uses to accomplish its purpose within organization.

Also the second definition states that control of process execution, of business data and activities is carried out by different services within the organization, which in our research are represented by business rules. Business rules can be seen as a separate entity that orchestrates the business processes.

Also we can notice that the BPM modeling should be done according to the business rules which affect each process and each activity. While an author proposes that each process should be treated as an object, we can understand from there that each process should be modeled according to the principles of reutilization. This allows loose-coupled activities which can be reused in other processes.

Finally, from Figure 4 and Figure 5, we could notice that the combination BPMN and SoaML in modeling is presented as the most suited for modeling this combination (SOA&BPM). The literature review revealed that the two modeling approaches are advantaged by minimizing the gap between the modeling techniques used. One important aspect is that service implementation is dependent and strongly influenced by the way that a process is designed. If a process is designed keeping in mind the business rules and process or activity reuse, the service implementation in SOA approach is achieved in a balanced manner.

3. BPM to REST-SOA Modeling and Mapping: Proposed Approach

In the following chapter we will propose a more specific approach in order to achieve a practical and testable workflow inspired from BPM-SOA mapping models like the one proposed by [9] and [7] and pictured in Figure 5 (Management of services oriented business process) and Figure 4 (Business processes and services relationships in MINERVA).

Our approach is based on a multi-step top-down process that aims to establish a comprehensible and, in the same time, a specific enough working procedure. It starts from the generic level of Business Process Model and tries to reach the detailed level of business behavior with REST-HTTP actions and of business data REST-based resource structures. Consequently, in our view the service-support level for BPM workflow will use the REST computing approach for SOA-based information systems.

The proposed workflow consists in the following stages:

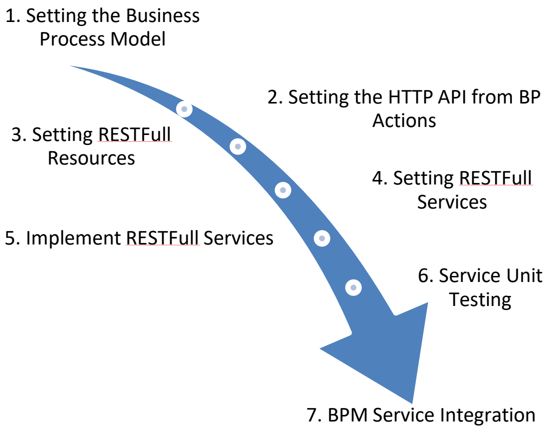

1. Setting the Business Process Model
2. Setting the HTTP API from BP Actions
3. Setting RESTFull Resources
4. Setting RESTFull Services
5. Implement RESTFull Services
6. Service Unit Testing
7. BPM Service Integration

Fig. 6. BPM-SOA Proposal workflow

DOI: 10.12948/issn14531305/21.1.2017.03

1. *Setting the Business Process Model*. BP.Action Design Specifications will take into consideration some general data to identify and qualify business action and also some more specific information about parametrization model. Summing up, the information needed in this stage will require the specification of:
   a. Action identifier
   b. Action Type, e.g.:
      - UIX
      - Atomic Processing
      - Synchronous
      - Timing
   c. Action Data Input
   d. Action Data Output
2. *Setting the HTTP API from BP Action Specifications*. HTTP Action Design Specifications will use a 1-to-1 mapping between BP.Actions and HTTP.Actions. The mapping model will impose a HTTP Action identification through an URI containing the original BP.Action identifier that will keep the trace-back information to original BP Model. Also, the HTTP Action types has to be compatible with the initial BP Action Types and HTTP Action parametrization system has to be compatible with the BP parametrization model. Summing up, the information needed in this stage will require specification of:
   a. HTTP Action identifier: URI (from base URL)
   b. HTTP Action Type:
      - HTTP Action kind, e.g.:
        - Resource-CRUD Action
        - Synchronous Function (RPC)
        - UI/UX
        - Asynchronous Acknowledgement (Event based)
      - HTTP predicate (GET, POST, PUT, DELETE)
   c. HTTP Action Input
      - Input URL Parameters: value parameters
      - Input Request headers: key-value parameters
      - Request Body format: structured (input) parameters by using XSD/XML or JSON documents
   d. HTTP Action Output
      - Output Header: key-value result set (or output parameters)
      - Response Body format: structured results as XSD/XML or JSON documents
      - Response Code: a simple status result
3. *Setting RESTFull Resources as a model of business entities*. In this stage one has to make the transition from process actions to an actually business data model by using HTTP CRUD Action types and identifying the underlying RESTfull resources. We could outline the mapping model in this case by using the expression:
   a. 1 Cohesive CRUD HTTP Action Set – to - … 1 Resource
4. *Setting RESTFull Services to provide RESTfull resources* (or business entities-based model). This stage aims to produce the modularization perspective to be used by the implementation of the underlying software components exposed as RESTfull services through resource access actions. The mapping directives has to settle the distribution of RESTfull resources and other HTTP processing functions to the actually RESTfull services:
   a. 1 Service – per - 1..* Resources
   b. 1 Service – per - 1..* HTTP Functions
5. *Implementation of RESTFull Services*, using platforms like JAX-RS, Spring MVC, etc. The previous stages covered the BPM-SOA mapping process that will finally result into a system design specification that will guide the implementation process of RESTfull services model but without any specific architectural and internal SOA constraints.
6. *Service Unit Testing* (service-level testing). Prior to deploying into a runnable Business Process Platform,

DOI: 10.12948/issn14531305/21.1.2017.03

every RESTFull Service has to be deployed and "to live" into an autonomous executable context/runtime that will allow their validation by modular and unit tests.
   a. 1 service ... 1 test suite
   b. 1 service action (HTTP Action) … 1 unit test
7. *BPM Service Integration and Testing* (process testing: process orchestration testing + service integration testing). Finally, with service components deployed autonomously, the business process actions embodied as HTTP-actions could be (re)integrated and orchestrated within a BP Platform Runtime from where the initial Business Process could be executed and validated by integration tests.

4. Practical Considerations on Modeling BPM to SOA

In the following section we will try to put our BPM-to-SOA in a practical context of a business process targeting the integration of university information systems to support student exchange programs according to the established models and approaches[10], [11]. A simplified BP model for student exchange programs could be described as in figure 7.

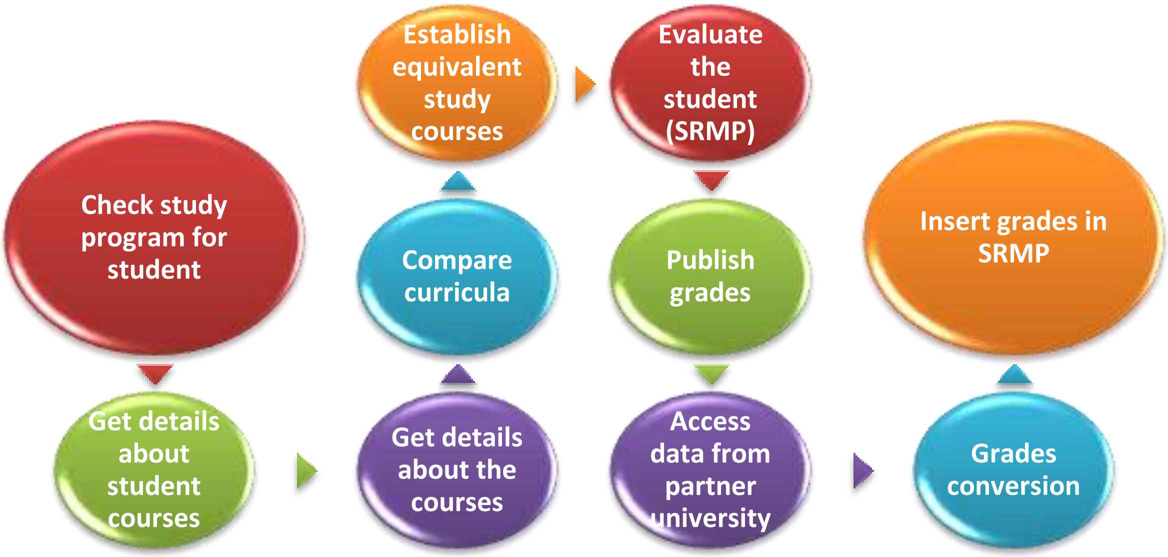

Establish equivalent study courses
Evaluate the student (SRMP)
Check study program for student
Compare curricula
Publish grades
Insert grades in SRMP
Get details about student courses
Get details about the courses
Access data from partner university
Grades conversion

Fig. 7. BP Model and actions

4.1. BPM Specifications

In order to define a context for our business process, one has to start by describing the BMP entities or actors responsible for the BP actions to be mapped by using HTTP services.

4.1.1 BPM Entities/Actors

For our practical case a list of BMP entities/actors could consist in: student, partner university, origin university, exchange program (like Erasmus) etc. Each actor/entity has to be related with its origin system (sun-system) and its SOA-accessible endpoint (as rest URL), like in the following spec:

DOI: 10.12948/issn14531305/21.1.2017.03

Table 3. Systems

System	SubSystem/Service	URL
Partner University	SRMP (Student RoadMaP - Professor, curriculum, study programs, modules, timetable)	./part.univ/SRMP ./part.univ/SRMP/students ./part.univ/SRMP/grades

4.1.2 BPM Action Specification

Before developing HTTP formalization, ones has to complement the BMP model with the action inventory and each BMP action has to get the relevant details like in the following specifications:

Action: <Identify details about the courses that the student attends>

Action Name	Get details about student courses
Action Type	READ
Action Data Input	Courses List from SRMP
Action Data Output	Courses Details from SPC

4.2 HTTP Action Specifications [HTTP API]

Having and extensive inventory of BMP actions, one could progress further with a HTTP specification like in the following spec where every BMP Action produced a HTTP argument:

Action HTTP: < Identify details about the courses that the student attends >

HTTP Action URL	http://server:host/SRS/<sub_module>/…/parent.univ/STX/courses/speciality/specName/courselist
BPM Action Name	Search Student Details at parent university
HTTP Action Type	READ
HTTP Predicate	GET
[Input] URL Parameters	semester
[Input] Request Body	
[Output] HTTP Response Code	200
[Output] Response Body	Courses details (XML/JSON)

4.3 Business Process Integration Model

Starting from the HTTP specification and developing concrete implementations of SOA-REST model one could create a BP integration model from the REST Action Model and a BMP concrete model from a platform like jBPM.

In this context, following the specifications one could create a project on jBMP platform (6.2.0 version) and could obtain a process diagram like the one illustrated in Figure 8.

DOI: 10.12948/issn14531305/21.1.2017.03

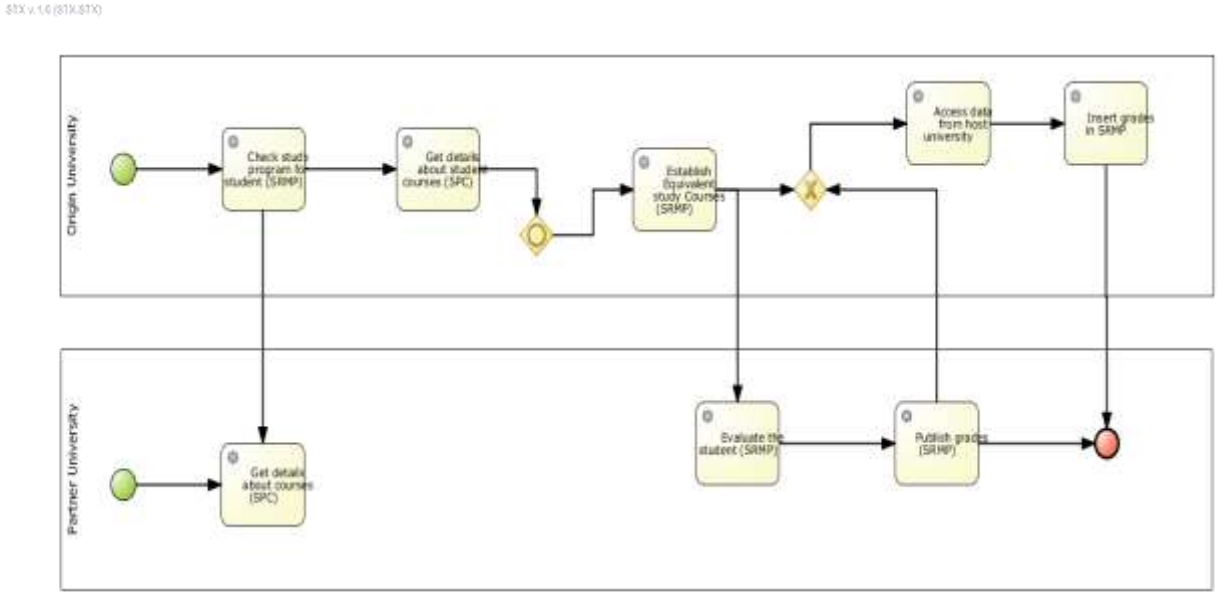
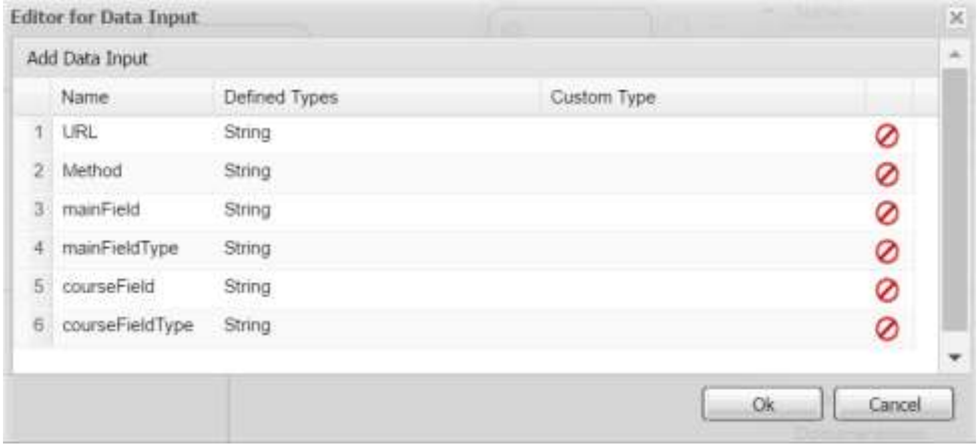

STX v.1.0 (STX.STX)
Origin University
Partner University
Check study program for student (SRMP)
Get details about student courses (SPC)
Establish Equivalent study Courses (SRMP)
Access data from host university
Insert grades in SRMP
Get details about courses (SPC)
Evaluate the student (SRMP)
Publish grades (SRMP)

Fig. 8. BPM.REST Action Model

Each BPM.REST action have some parameters and parameters assignment. The common parameters are URL, that represents the URL of the action endpoint and for each HTTP method of request we have parameter named Method. Bellow we have illustrated the Action Parameters and Action Parameters Assignment under each Action.

BPM.REST Action 2 [Get details about student courses] Specifications

- Action REST Resource Target (from REST Resource Model) – CourseResource;
- Action Parameters;
- Action Parameters Assignment.

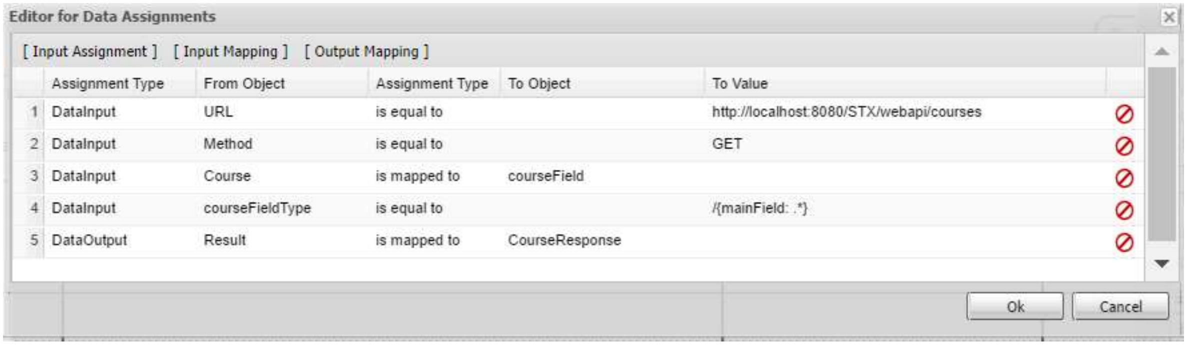

Editor for Data Input

Add Data Input

	Name	Defined Types	Custom Type
1	URL	String	
2	Method	String	
3	mainField	String	
4	mainFieldType	String	
5	courseField	String	
6	courseFieldType	String	

Fig. 9. Data Input Form

DOI: 10.12948/issn14531305/21.1.2017.03



Editor for Data Assignments

	Assignment Type	From Object	Assignment Type	To Object	To Value	
1	DataInput	URL	is equal to		http://localhost:8080/STX/webapi/courses	⊘
2	DataInput	Method	is equal to		GET	⊘
3	DataInput	Course	is mapped to	courseField		⊘
4	DataInput	courseFieldType	is equal to		/{mainField: .*}	⊘
5	DataOutput	Result	is mapped to	CourseResponse		⊘

Fig. 10. Data Assignments Form

5 Conclusions

Once we combine BPM and SOA, process modeling comes to be the expression of SOA, facilitating teams to turn out to be more agile, and agreeing them to see the requirements of the business by retaining a flexible management level under their processes. BPM supports organizations to plan concrete service-oriented architectures by dipping the uncertainty in the effort to grow grained services inside those architectures. Under the proposed approach in previous chapters of the article, combining SOA and BPM generates greater efficiency in the management information systems.

The whole approach depends very much on the first two stages: Setting the Business Process Model and Setting the HTTP API from BP Action Specifications. Starting from them, the methodology allows the implementation of RESTFull services and BPM service integration.

As a future direction we intend to expand the proposed model so that it can be applied to any language that allows the implementation of REST.

Acknowledgments:

„This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number PN-II-RU-TE-2014-4-0748”.

References

[1] K. Gopala, "BPM and SOA: A Strategic Alliance," BPTrends, pp. 1-7, May 2006.

[2] V. Gacitua-Decar and C. Pahl, "Business Model Driven Design of Service Architectures for Enterprise Applications Integration: A Pattern-based Approach.," in Proceedings of the Doctoral Consortium on Software and Data Technologies, Porto, 2008.

[3] M. Kanakia and P. Chawan, "Evolution of modelling techniques for Service Oriented Architecture," Journal Of Engineering Research and Applications, vol. 4, no. 7, pp. 229-232, 2014.

[4] M. Mohsen and M. Muriati, "A review of SOA Modeling Approaches for Enterprise Information Systems," Proceedia Technology, vol. 11, pp. 794-800, 2013.

[5] IBM, "IBM Service-Oriented Modeling and Architecture," IBM Global Services, New York, 2004.

[6] G. Matei, "SOA and BPM, a partnership for Successful Organizations," Informatica Economica, vol. 15, no. 4, pp. 39-54, 2011.

[7] M. Mircea, "Adapt Business Processes to Service Oriented Environment to Achieve Business Agility," Journal of Applied Quantitative Methods, vol. 5, no. 4, p. 680, 2010.

[8] A. Florea, A. Andreescu, V. Diaconita and A. Uta, "Approaches Regarding Business Logic Modeling in Service Oriented," Informatica Economica, vol. 15, no. 3, pp. 88-98, 2011.

[9] A. Delgado, F. Ruiz, I. Guzman and M. Piattini, "A Model-driven and Service-oriented framework for the business process improvement," Journal of Systems Integration, vol. 3, pp. 45-55, 2010.

[10] C. Strimbei, O. Dospinescu, R. Strainu and A. Nistor, "The BPMN Approach of

DOI: 10.12948/issn14531305/21.1.2017.03

the University Information Systems," Ecoforum, vol. 5, no. 2, pp. 181-183, 2016.

[11] N. Dospinescu, M. Tatarusanu, G. Butnaru and L. Berechet, "The Perception of Students from the Economic Area on the new Learning Methods in the Knowledge Society," Amfiteatru Economic, vol. 13, no. 30, pp. 527-543, 2011.

Octavian DOSPINESCU graduated the Faculty of Economics and Business Administration in 2000 and the Faculty of Informatics in 2001. He achieved the PhD in 2009 and he has published as author or co-author over 30 articles. He is author and co-author of 10 books and teaches as an associate professor in the Department of Information Systems of the Faculty of Economics and Business Administration, University Alexandru Ioan Cuza, Iasi. Since 2010 he has been a Microsoft Certified Professional, Dynamics Navision, Trade&Inventory Module. In 2014 he successfully completed the course “Programming Mobile Applications for Android Handheld Systems” authorized by Maryland University. He is interested in mobile devices software, computer programming and decision support systems.

Cătălin STRÎMBEI has graduated the Faculty of Economics and Business Administration of Al.I.Cuza University of Iaşi in 1997. He holds a PhD diploma in Cybernetics, Statistics and Business Informatics from 2006 and he has joined the staff of the Faculty of Economics and Business Administration as teaching assistant in 1998 and as associate professor in 2013. Currently he is teaching Object Oriented Programming, Multi-Tier Software Application Development and Database Design and Administration within the Department of Business Information Systems, Faculty of Economics and Business Administration, Al.I.Cuza University of Iaşi. He is the author and co-author of four books and over 30 journal articles in the field of object oriented development of business applications, databases and object oriented software engineering.

Roxana-Marina STRAINU graduated in 2014 the Master of Business Information Systems at the Faculty of Economics and Business Administration, Alexandru Ioan Cuza University of Iasi. She also graduated the Faculty of Mathematics in the year 2005. She is interested in developing smart systems and mobile applications on Android platform. Now she is a PhD student in the business information systems area.

Alexandra NISTOR graduated the Faculty of Economics and Business Administration in 2011 and the Master of Business Information Systems at the Faculty of Economics and Business Administration in 2013. Her research interests include the use of automated testing in small and medium companies. Now she is a PhD student in the business information systems area.

DOI: 10.12948/issn14531305/21.1.2017.03