Deriving Service Level Agreements from Business Level Agreements

An Approach Towards Strategic Alignment in Organizations

Vitor Almeida Barros[1], Marcelo Fantinato[1], Guilherme M. B. Salles[1] and João Porto de Albuquerque[2]

[1]*School of Arts, Science and Humanities, Univ. of São Paulo, 1000 Arlindo Béttio Av., São Paulo, Brazil*

[2]*Inst. of Mathematical and Computer Sciences, Univ. of São Paulo, 400 Trabalhador São-carlense Av., São Carlos, Brazil*

Keywords: Strategic Alignment, Business Process Management, Service-oriented Computing, Service Level Agreement, Business Level Agreement, WS-Agreement.

Abstract: Business Process Management (BPM) can help organizations in their attempts to align strategies between business and information technology areas. It is not only necessary to address functional properties during the BPM life-cycle, but also process quality and operating constraints, which are usually grouped together as Non-Functional Properties (NFP). However, the most prestigious languages for business process modelling are unable to represent these NFPs, and this creates a gap between the degree of success in identifying functional properties and NFPs as well as between the process modelling and its implementation. We have attempted to fill this gap by proposing the StrAli-BPM (Strategic Alignment with BPM) approach, which is divided in two parts – BLA@BPMN and BLA2SLA: the former seeks to extend the BPMN language by embodying NFPs, in the form of BLAs (Business Level Agreements); and the latter semi-automatically derives a set of SLAs (Service Level Agreements), linked with web services, from a pre-defined BLA. This paper outlines the BLA2SLA part of the StrAli-BPM approach. In addition, it includes a prototype tool developed to validate BLA2SLA and the results of an experiment undertaken to validate it.

1 INTRODUCTION

The business scenario has undergone major changes in the area of Information Technology (IT), since IT has been adding value to business undertakings. Largely as a result of competition, IT has become a means of processing information to support decision-making (Laudon and Laudon, 2009). Organizations regard high investment in IT as vital to obtain products and services supported by this tool, and view it as one of the key factors in achieving success in their strategic goals (Bruin and Rosemann, 2010). When integrating Business and IT, it is an ongoing challenge to prevent the organization's investments in IT from being misdirected through low productivity and poor quality (Shimizu, Carvalho and Laurindo, 2006). Furthermore, with the high degree of dependency on IT and investment in it by organizations, it is essential to have suitable and well-structured methods to ensure the solutions it offers are aligned with corporate expectations and needs.

Business Process Management (BPM) can be applied in this context to transform business processes within organizations (Weske, 2007). From the perspective of BPM, with the support of Service-oriented Architecture (SOA), strategic alignment can be achieved by managing and improving the technological means by which processes can add value to the organization (Allen, 2006). Process modelling is critical to guide strategic alignment and, hence, should not only take account of functional properties but also Non-Functional Properties (NFP) as a means of achieving quality goals effectively and efficiently (Pressman, 2009). Examples of NFPs are reliability, performance and scalability, which directly influence the functional capacity of a process (Sommerville, 2010).

However, there are several issues involved in handling NFPs in this scenario in terms of the available methodologies and tools. For example, process modelling languages, including BPMN (Business Process Model and Notation) (OMG, 2011), are only concerned with functional capacity modelling (Pavlovski and Zou, 2008). In terms of process execution, there are already well-established concepts of Quality of Service (QoS) (Khaluf, Gerth and Engels, 2011) and Service Level Agreement

Barros V., Fantinato M., M. B. Salles G. and Porto de Albuquerque J..
Deriving Service Level Agreements from Business Level Agreements - An Approach Towards Strategic Alignment in Organizations.
DOI: 10.5220/0004887202140225
In *Proceedings of the 16th International Conference on Enterprise Information Systems* (ICEIS-2014), pages 214-225
ISBN: 978-989-758-029-1

(SLA) (Caseau, 2005) for handling NFPs. Yet, from the perspective of a strategic alignment, NFPs are only created during the implementation of the executable business processes, rather than at the beginning of the BPM lifecycle.

It is necessary to take account of NFPs related to organizational goals so that quality constraints can be introduced at the business level; and this can be done by Business Level Agreements (BLA) (Bratanis, Dranidis and Simons, 2010). A BLA can be understood as a set of SLAs; however, whereas each SLA is combined with an electronic service (e-service) and is designed to ensure quality at the service level, a BLA is combined with a part of the process model (i.e. a sub-process) that can be executed by a set of services. Hence, a BLA must formalize a goal at a business level, and this can be expressed by a specific NFP linked with a metric that must be achieved, and represents the broader requirements of the organization.

Currently, there is still no entirely satisfactory solution for enabling business and IT areas to work in alignment with modelling processes and their subsequent execution, which takes account of both functional properties and NFPs. In an attempt to provide a strategic alignment in the context of BPM/SOA, we proposed the StrAli-BPM (Strategic Alignment with BPM) approach, which is formed of two parts: *(i)* BLA@BPMN – an extension of the BPMN notation that is designed to include BLAs and *(ii)* BLA2SLA – a top-down strategy to refine BLAs into SLAs which is combined with the e-services that compose the executable process derived from the BPMN model. This paper's main focus is on the BLA2SLA part since BLA@BPMN has already been outlined (Salles *et al*., 2013).

This paper is structured as follows: Section 2, the main background concepts; Section 3, the methodology employed; Section 4, an overview of the StrAli-BPM approach; Section 5, details of the BLA2SLA part; Section 6, an assessment of the approach being adopted; and, finally, Section 7, conclusions and recommendations for further work.

2 CONTEXTUAL BACKGROUND

In the BPM context, NFPs are constraints that have to be imposed to ensure quality and meet other business process requirements (Pavlovski and Zou, 2008). However, NFPs are not yet widely accepted and often receive less attention than the functional properties in the BPM approaches. BPMN, for instance, in its official version, does not deal with NFPs although there are some limited proposals in the literature. Taking into account some similarities, BPM approaches should be given the same conduct as that given to the functional and non-functional requirements in the Software Engineering approaches (Sommerville, 2010).

The functional performance of IT services depends on a number of NFPs, such as: availability, response time and accuracy. The QoS assurance provided by IT is one of the most important factors for business success, as well as improving competitiveness (Khaluf, Gerth and Engels, 2011). SLAs can formalize and ensure that these operating and quality constraints are overcome. SLAs are thus able to assess whether the service provided complies with the terms agreed between parties (Theilmann *et al*., 2010). With regard to the specific BPM/SOA context, the most commonly languages employed to formalize SLAs are WS-Agreement and WS-Policy, both based on XML and recommended by W3C. SLAs address the QoS attributes and levels, and are linked to the web services that provide an executable process that is implemented to meet the functional requirements of a process model (Caseau, 2005). The SLAs specified in one of these two languages have their own structures and can, for example, be used to define goals, metrics, fines and bonuses.

A BLA, in turn, can be considered to be equivalent to a SLA at a business level. SLA features are, hence, found in a BLA, and include: agreement validity, fines and bonuses (Bratanis, Dranidis and Simons, 2010). Thus, a BLA, linked to a sub-process, can be hierarchically mapped to a set of SLAs. During the process modelling, a BLA is regarded as a "father" agreement that is related to a set of activities in the process model. During the process implementation, this BLA can derive a set of SLAs that are linked to the respective services. These services, in turn, were derived from the business process model to perform its set of activities. This top-down derivation, as illustrated in Figure 1, defines a set of SLAs that, if all are met, allows the BLA goal to be achieved in an induced way (Goel, Kumar and Shyamasundar, 2011).

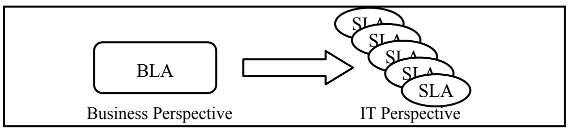

Figure 1: Relationship between BLA and SLAs.

In essence, the structures of the SLAs and BLAs may be similar. The main difference between them is the type of goal that each one defines: SLAs must assure QoS fulfilment, whereas BLAs aim at determining the strategic requirements and target values that the organization has to achieve.

These strategic requirements inside a BLA can be defined through Key Performance Indicators (KPI), which is a management tool to measure business effectiveness (Wetzstein *et al*., 2009). KPIs are used to support the definition and measurement of strategic organizational requirements (Parmenter, 2010). Moreover, the definition of the KPIs depends on the business features and economic or organizational context (Friedenstab *et al*., 2012).

Whereas the SLA concept, as well as the QoS, is firmly established in industry as well as in academia, BLA is still considered to be a new concept, whose structure and usage have not yet been mastered either conceptually or in practice.

A short, illustrative example is given in Figure 2 to clarify the differences between BLA and SLA and give a better explanation of how BLAs can ensure that organizational goals are met through KPIs.

- **Organization**: a financial institution with nearly twenty million customers;
- **Corporative context**: in 2013, priority will be given to customer satisfaction;
- **Process**: loan contracting for account holders;
- **BLA**: following a customer's loan request, a credit contract should be created within a maximum of six hours, including intermediate activities, assuming that the proposal is accepted by the credit board;
  - KPI: processing time;
  - Target value: six hours or less;
- **SLA-1**: activity/service "Send proposal to the credit board for analysis";
  - Service latency ≤ 30 seconds;
  - Service availability ≥ 99,95% (from 10am to 4pm);
- **SLA-2**: activity/service "Analyze proposal data";
  - Service response time (analyze the proposal and enter the results into the system) ≤ 4 hours;
- **SLA-3**: activity/service "Send response of the proposal analysis";
  - Service availability ≥ 99% (from 10am to 4pm);
- **SLA-4**: activity/service "Generate a credit contract";
  - Service response time ≤ 5 minutes;
  - Service availability ≥ 99.95%;
  - Service recovery in case of failure ≤ 25 minutes.

Figure 2: Illustrative example of BLA and SLAs.

3 METHODOLOGY

To ensure that the results of this work were achieved, this study was based on the design science research methodology (Hevner *et al*., 2004). This reflects business needs and finds ways of catering for them by underpinning the investigation with the scientific development of solutions as well as drawing on the results achieved by the research. The purpose of the artefacts that are being developed is to address unsolved problems or those that cannot be solved satisfactorily, which, in this study, is the lack of any formal representation for BLA, the lack of an approach to relate BLA to BPMN and the lack of an approach to derive BLA into SLAs. In this work, the designed-oriented approach was adopted (Osterle *et al*., 2011). This involved conducting an analysis of business needs to investigate both the external and internal factors that influence these issues. As a result, the design and preliminary evaluation of the proposed solution (i.e. both parts – BLA@BPMN and BLA2SLA – of the StrAli-BPM approach) were carried out in the initial stage. Finally, the communication stage was executed to disseminate the results obtained.

On the basis of the methodology outlined above, a theoretical survey and a study on the state of the art of the context discussed were carried out as a part of this study. The theoretical survey included topics such as strategic alignment between business and IT, BPM, SOA, NFP, KPI, BLA, SLA/QoS and the BPMN and WS-Agreement languages. This was followed by a comprehensive development and evaluation of the BLA@BPMN part of the StrAli-BPM approach, although it is not strictly within the scope of this paper since it has already been examined by Salles *et al*. (2013). Following this, it was possible to address, the BLA2SLA part.

A WS-Agreement metamodel was devised, by using a UML Class Diagram, to allow a better understanding of the concepts that underpin the WS-Agreement, and thus link the concepts of the two parts of StrAli-BPM. After this, there was a systematic mapping to make a comparison between the existing BLA@BPMN metamodel and the newly created WS-Agreement metamodel. This enabled us to have an exact understanding of how: information of a BLA that is linked with a BPMN, and exported in a XML format, can be used to provide information for the derived SLAs through the fields specified in the WS-Agreement schema. C# was used to implement a prototype tool for BLA2SLA. This tool uses the XML standard for both the data input (BLAs exported from BLA@BPMN prototype tool) and the data output (WS-Agreement language).

4 StrAli -BPM

The StrAli-BPM approach proposed by Salles *et al*. (2013) employs BPMN as its basis. It was chosen because of its widespread use and standardization. The mapping from processes modelled in BPMN to its executable version in WS-BPEL/WSDL has already been covered by other studies, such as by Mazanek and Hanus (2011), and can be called BPMN2BPEL. StrAli-BPM establishes an analogue mapping, in which the BLAs, which are represented as an extension of the BPMN models, can be mapped to an SLA standard (WS-Agreement, for example). These SLAs, in turn, are combined with the web services that compose the executable process and form relationships that are graphically shown in Figure 3. The focal point of this paper is on how a set of SLAs can be derived from a BLA, i.e. the transformation of the NFPs; and is similar to the way in which functional properties can be transformed by BPMN2BPEL.

The conceptual framework outlined in Figure 3 shows more clearly how StrAli-BPM closes a cycle to achieve the strategic alignment, by using BPMN, BLAs and SLAs. In terms of phases, the first stage (analysis and design) usually comes under the responsibility of business areas, whereas the second (implementation and execution) is usually performed by IT. From another perspective, this approach allows both functional properties and NFPs (represented through BLA and SLA) to be defined in an integrated, top-down manner.

The first part of StrAli-BPM, which is not addressed in this paper, is called BLA@BPMN (BLA at BPMN) and consists of a BPMN extension to represent the BLAs, which use KPIs to define their goals, combined with groups of activities in the process. This extension includes both a metamodel to represent the BLAs and a graphical notation.

The second part of the StrAli-BPM, called BLA2SLA (BLA to SLA), is related to the semi-automatic derivation of BLAs to SLAs, which are combined with the web services that compose the executable process. The next section examines the structure used to represent SLAs, the mapping between the BLAs and SLAs, and the prototype tool employed to validate the BLA2SLA.

In StrAli-BPM, a BLA is accompanied by some general recommendations with regard to its structure (Allen, 2006). The general recommendations aims at identifying: the Goal (and its linked KPI), Penalties and Rewards (Pourshahid et al., 2009; Friedenstab et al., 2012). The proposed structure for a BLA is represented in Figure 4 as a metamodel. A BLA artefact thus has a well-defined attribute structure that can be used as a template to create and store BLAs. Some of the attributes in Figure 4 have similarities to SLAs, albeit at a business level.

As shown in Figure 4, Goal, which uses the KPI concept, represents which property or requirement should be fulfilled or improved by the BLA, together with the target value that must be achieved. KPIs, as discussed in Section 2, must be defined in accordance with business requirements. Penalty and Reward, in turn, represent indemnities which must be paid for by the Customer and Supplier, respectively, depending on the extent to which the Goal is attained. The Penalty attribute can have one or more occurrences in each BLA; and, the Reward attribute, none or more occurrences. The BLA class

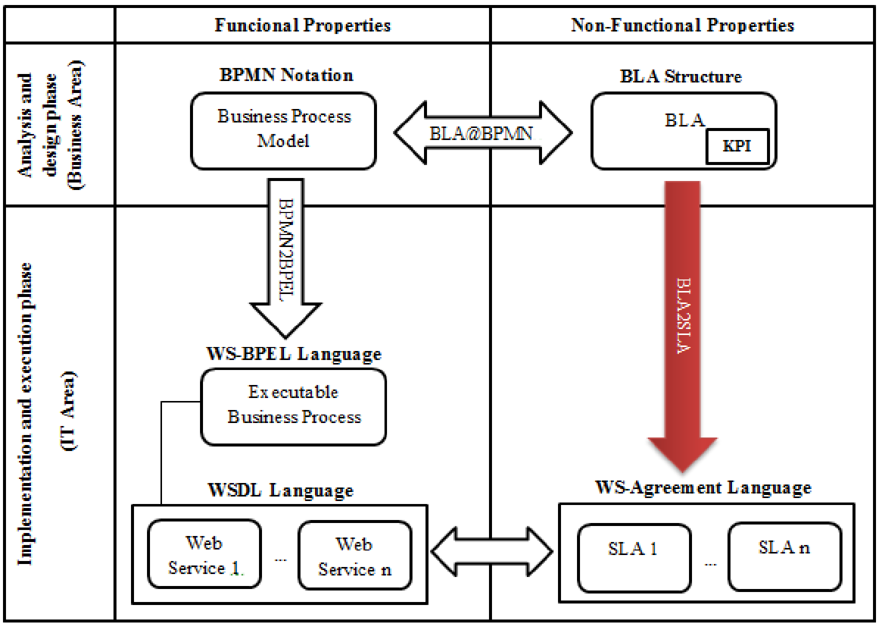

Figure 3: StrAli-BPM Approach - Proposed Conceptual Framework.

relates to a series of class in the BPMN metamodel. These relationships are not scope of this paper.

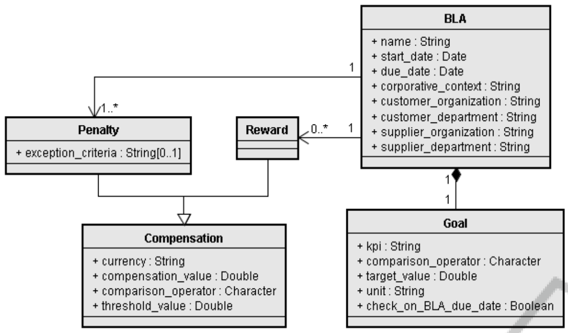

Figure 4: BLA Metamodel.

As regards the BLA@BPMN part, a new object type, designated as BLA, was incorporated into BPMN (version 2.0) as a Custom Artefact, tested in the Bizagi tool. Figure 5 shows the BLA artefact represented in a process model for credit application (like the example in Section 2), according to the visual patterns of Bizagi. The BLA artefact connects to a sub-process through a "Group artefact" and an "Association connector", both of which already exist in BPMN. The BLA object called "6 hours for contracts origination" involves four activities of the process model that are associated with the target goal set, i.e. a BLA goal, which uses the KPI concept, comprises a set of one or more activities (which later will be derived to a set of one or more SLAs). By employing this approach, and taking account of the nature of the BLAs, two or more BLAs can have a non-empty intersection with each other, i.e. the same activity can be contained in two or more BLAs.

5 BLA2SLA APPROACH

In an attempt to achieve an alignment between business and IT, BLA2SLA proposes that the goals established at the business level, that are represented in BLA, be used as input for defining the SLAs in a top-down process. This derivation process requires the work of an IT specialist with knowledge of business needs and especially of the business process domain and web services technology. Thus, after the definition of the KPI-based goals, that are linked to business processes in a BLA-form, these BLA-goals will be divided into specific objectives, (at an operational level), which are represented by the SLAs and use non-functional properties through QoS attributes. The next subsections describe BLA2SLA in terms of: SLA structure, BLA-SLA mapping and as a support tool.

5.1 SLA Structure

The BLA structure defined for BLA@BPMN, through a metamodel, was designed to be flexible enough to allow the BLA-SLA mapping to be undertaken in different SLA and QoS specification languages. In the specific case of the StrAli-BPM approach, the WS-Agreement language was chosen to carry out the mapping.

WS-Agreement aims at defining and monitoring the SLAs that have been established between the customer and supplier for a set of web services (Andrieux *et al.*, 2007). The task of specification is done by using XML and monitoring occurs while the web services are being executed. WS-Agreement thus describes the related web services and the SLAs

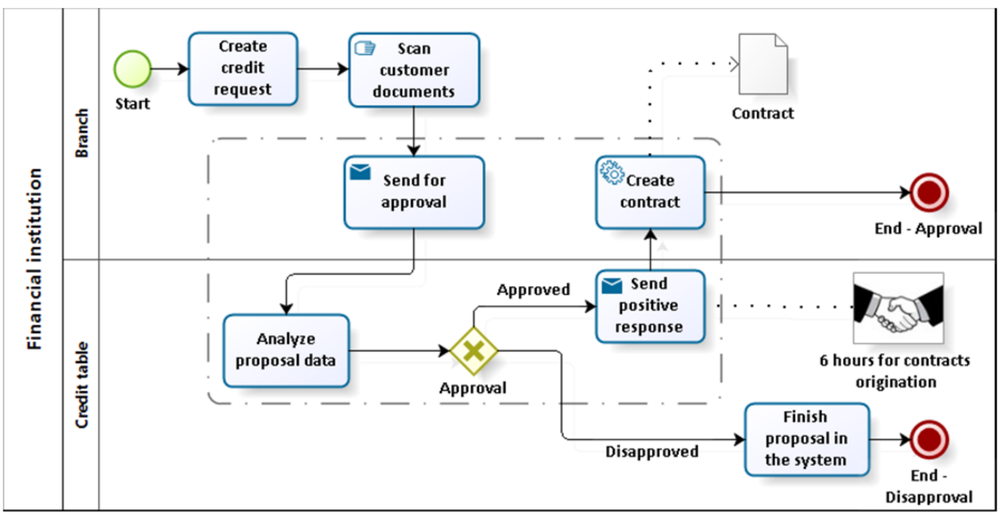

Figure 5: Illustrative example of BLA@BPMN.

established for them, including their QoS attributes and guarantees, such as fines and bonuses, which must be applied in accordance with the fulfilment of the service level agreements.

The basic structure of an SLA specified in a WS-Agreement is illustrated in Figure 6. Name serves as an identifier and Context should mention, at least, the customer, the supplier and the validity of the agreement. Terms represent the formal obligations between the parties. This section should contain at least: *(i)* Service Terms, which define the functional part, i.e. the data with respect to the web services involved in the agreement; and *(ii)* Guarantee Terms, which define the non-functional part, i.e. the SLA goal and the inclusion of penalties and bonuses to ensure that the goals are met.

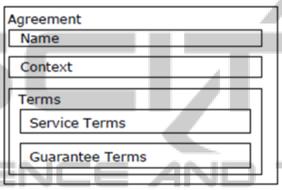

Figure 6: WS-Agreement Framework.

A WS-Agreement metamodel was established to systematize the derivation rules of BLAs to SLAs, since no suitable metamodel was found in the literature. The WS-Agreement metamodel was formed by means of the language specification outlined by Andrieux *et al*. (2007), and maintained by the Open Grid Forum. The resulting WS-Agreement metamodel is shown in Figure 7. This is a slightly simplified version when compared with the language schema, but it meets the needs of StrAli-BPMs. All the required classes and attributes of the scheme were complied with. Additionally, some of the optional elements which are needed in the BLA-SLA mapping were also retained.

As shown in Figure 7, in a WS-Agreement, AgreementId and Name are used as identifiers. In Context, Agreement Initiator defines the initiator of the request for an agreement creation; Agreement Responder defines the entity that responds to the agreement creation request; and ServiceProvider Agreement Initiator/Responder defines which of these parts must be declared as a SLA supplier. Still in Context, ExpirationTime defines the time at which the agreement is no longer valid; and TemplateId and TemplateName optionally define a SLA template in WS-Agreement (set, respectively, to an AgreementId and a Name), which is used to specify an agreement from pre-defined information.

Terms represent the formal obligations between the involved parties. ServiceDescriptionTerm, ServiceReference and ServiceProperties compose the Service Terms (as described in Figure 6), which define the functional part of the SLAs, i.e. the description of web services involved in SLAs. Web services can either be completely described or else only referenced, when they have been previously specified in WSDL. ServiceDescriptionTerm, the only mandatory class, lists all the web services, through ServiceName, which can be used later while creating the Guarantee Terms. ServiceReference

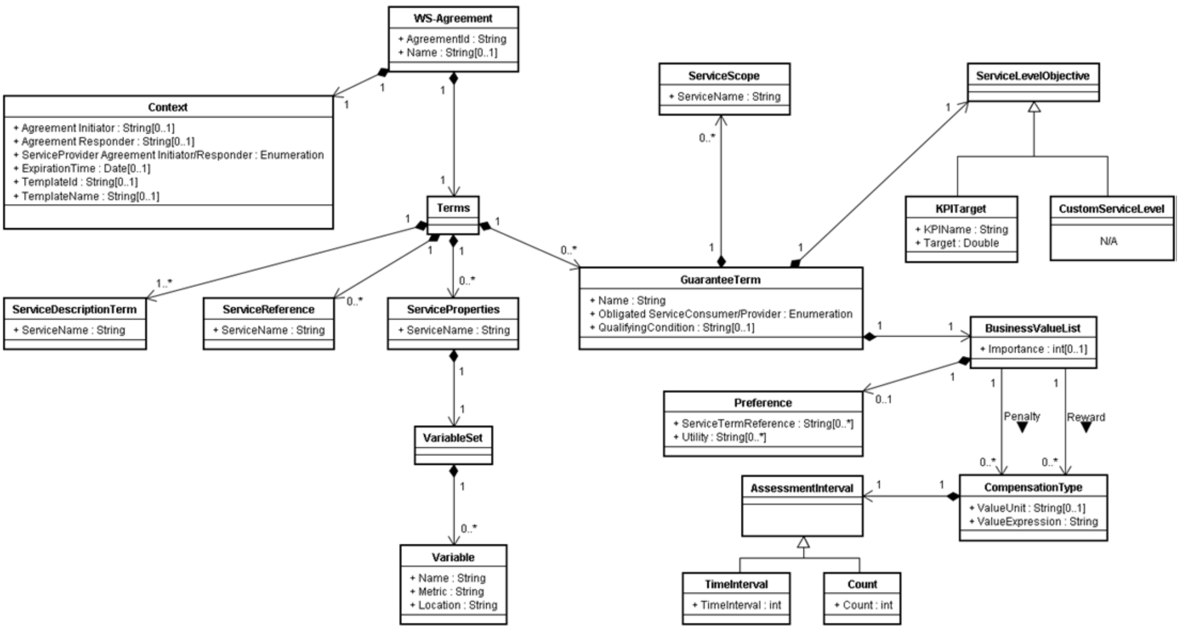

Figure 7: WS-Agreement Metamodel (SLA Structure).

may be optionally used to reference external information of the web services, through ServiceName, such as a WSDL specification. And ServiceProperties may be (optionally) used to specify non-functional properties of the web services, also through ServiceName, which can be used later while creating the Guarantee Terms. ServiceProperties is composed of VariableSet which, in turn, is composed of Variable. In Variable, Name defines the name of the non-functional property (e.g. availability, latency, cost); Metric defines the semantics of the non-functional property (e.g.: percentage, second, dollar); and Location can store a reference in a local/file – usually in the XML structure – that describes features in that non-functional property.

GuaranteeTerm composes the non-functional and hence most important part of the WS-Agreement (as described in Figure 6), i.e. the agreement goals and the possible penalties and bonuses used to encourage the fulfilment of the established goals. In GuaranteeTerm, Name defines the name given to a guarantee, which in practice represents the name of each SLA that is part of a WS-Agreement specification. Obligated Service Consumer/Provider defines what is the part, considering the service consumer and the service provider, responsible for meeting the goal. QualifyingCondition, when used, defines a precondition which is required for the GuaranteeTerm to hold. ServiceScope defines which services, specified through ServiceDescriptionTerm, are included in a given GuaranteeTerm.

ServiceLevelObjective defines the SLA goal in GuaranteeTerm, through two options: KPITarget and CustomServiceLevel. In KPITarget, KPIName and Target defines the SLA goal as, respectively, a KPI expression and a KPI target expression. KPIName can use data (i.e. non-functional properties) defined as Name in Variable. In CustomServiceLevel, any structure of attributes can be generically defined to specify the SLA goals according to needs in exceptional cases.

BusinessValueList defines a set of values for the GuaranteeTerm, each expressing a different aspect of the ServiceLevelObjective. In BusinessValueList, Importance, when present, defines the relative importance of meeting a ServiceLevelObjective. CompensationType defines a Penalty or a Reward, to be applied if the SLA goal is not attained. In CompensationType, ValueExpression and Value Unit define, respectively, the type of currency (e.g.: R$, USD) and amount of money that must be paid in each compensatory action. A Penalty must be paid from the provider to the consumer, and a Reward in reverse. A Penalty or Reward can apply on multiple occasions for the same GuaranteeTerm, in different periods. AssessmentInterval defines the application frequency of a compensatory action, through two options: TimeInterval and Count. TimeInterval defines frequency during a given time interval such as number of days, number of weeks, number of months, etc. Count defines frequency through the number of invocations of the service with regard to the GuaranteeTerm. Preference, when used, defines alternative ways of achieving the associated Service LevelObjective. In Preference, ServiceTerm Reference refers to one or more ServiceName, in ServiceDescriptionTerm, and can be used as alternative ways to achieve the ServiceLevel Objective; and Utility defines respective utilities obtained by achieving this ServiceLevelObjective.

5.2 BLA-SLA Mapping

Each BLA, which contains a single goal, must be derived into a set of SLAs, each with its specific goal. In the light of the use of WS-Agreement to specify SLAs, a BLA structure should physically result in a WS-Agreement file, which in turn can contain various SLAs represented by different GuaranteeTerm sections in the WS-Agreement file.

In deriving BLA to SLAs, a part of the attributes mapping is done directly, i.e. the information specified in the BLA can be automatically forwarded to the SLA specification. In contrast, the other part is done indirectly, i.e. the BLA information should be regarded as simply consisting of suggested values to be used as SLA attributes or values to be used only as a reference for specifying the SLA attributes.

Table 1 shows the direct mapping from the attributes of the BLA metamodel (see Figure 4) to the attributes of the WS-Agreement metamodel (see Figure 7). To compose the Agreement Initiator and Agreement Responder attributes in SLA, a concatenation of, respectively, the customer_organization + customer_department and supplier_organization + supplier_department attributes, from BLA, is proposed.

Additionally, Table 2 shows SLA attributes that can be indirectly created as a suggestion or a reference, from either the information from BLA or the WS-Agreement framework. The information supplied in Table 2 should only be used as reference values, basically to support the specialist work.

Table 1: Direct mapping from BLA to SLA attributes (directly automatable mapping).

BLA (from)			**SLA / WS-Agreement (to)**		
Class	**Attribute**	**Type**	**Class**	**Attribute**	**Type**
BLA	name	String	WS-Agreement	Name	String
	due_date	Date	Context	ExpirationTime	Date
	corporative_context	String	GuaranteeTerm	QualifyingCondition	String
	customer_organization	String	Context	Agreement Initiator	String
	customer_department	String			
	supplier_organization	String		Agreement Responder	String
	supplier_department	String			
Activity	name	String	ServiceDescriptionTerm	ServiceName	String
			ServiceReference	ServiceName	String
			ServiceProperties	ServiceName	String
			ServiceScope	ServiceName	String

Table 2: Indirect mapping from BLA to SLA attributes (not directly automatable mapping).

SLA / WS-Agreement (to)			**Reference – BLA / SLA (from**		
Class	**Attribute**	**Type**	**Option**	**From**	**Value**
Guarantee Term	Name	String	Reference value	BLA / SLA	name attribute from WS-Agreement class or BLA class + KPIName attribute from KPITarget class
KPITarget	KPIName	String	Reference value	BLA	all attributes from the Goal class
	Target	Double			
Compensation Type	ValueUnit	String	Suggested value	BLA	currency attribute from Compensation class
	Value Expression	String	Reference value	BLA	compensation_value attribute from Compensation class
Count	Count	Integer	Suggested value	BLA	Value "1" [when the attribute check_on_BLA_due_date from Goal class is 'equal to false']
TimeInterval	Time Interval	Integer	Suggested value	BLA	Difference (in days) between the due_date from BLA class and the current date [when the attribute check_on_ BLA_due_date from Goal class is 'equal to true']

These values can also be used by the support tool as suggested by the pre-filled values, which can be changed if the specialist does not agree with them.

In an a priori approach, the total value of compensatory actions (ValueExpression) of a set of SLAs must not exceed the value of the compensatory action (compensation _value) of the respective BLA used to derive these SLAs. However, in view of the fact that the application frequency of BLA and SLA actions may differ, this systematic control was not predicted for BLA2SLA.

Although they are described in Figure 7, some classes and attributes of the WS-Agreement metamodel are not addressed in the BLA-SLA mapping shown in this section. They are also not included in this first version of the BLA2SLA approach being proposed. The following items of the WS-Agreement metamodel have still not been addressed: attributes TemplateId and TemplateName from Context class; attribute Location from Variable class; attribute Importance from BusinessValueList class; and the whole of the Preference class.

The BLA goal, which is defined in terms of a KPI, must be converted to non-functional properties at the service level, and this can be represented by QoS attributes. A well-defined list of QoS attributes was used with the aim of standardizing the SLAs created by BLA2SLA. Table 3 lists the QoS attributes used in this study for this purpose.

5.3 BLA2SLA Prototype

The prototype tool for BLA2SLA was designed to act as a semi-automatic converter. It was developed in C# language, given its robustness for processing data in XML format. The tool computationally interprets the BLAs, exported in the XPDL language (based on XML) by the BLA@BPMN tool, and supports the specialist user in the creation of SLAs to be specified in the WS-Agreement language (also based on XML), respecting their pre-defined structure and metamodel (see Figure 7). The XPDL files exported by BLA@BPMN contain information from both the business process model specified in BPMN as well as the linked BLAs.

Table 3: QoS attributes used by BLA2SLA, as proposed by Garcia and Toledo (2006).

QoS attributes	**Unit (metric)**	**Definition**
Accuracy	Percentage	The service error rate over a time interval
Capacity	Integer	The number of concurrent requests a service allows
Reliability	Time	The time for continuity of correct service and for transition to correct state
Cost	Double	The calculation of the costs incurred of using the service
Availability	Percentage	The time a service is operating (in percentage terms)
Scalability	Percentage	The throughput increase rate in a given time interval
Stability	Percentage	The rate of change of service interface
Latency	Time	The time taken to start servicing a service request
Robustness	Integer	The level of service resilience l to incorrect inputs and invocation sequences
Throughput	Percentage	The request processing rate a service supports
Response time	Time	The time a service takes to complete its task

The tool has a main screen, as illustrated in Figure 8, which is split in: (i) the top, which contains general information on the SLAs (i.e. those related to Name, Context and Service Terms according to Figures 5 and 6); and (ii) the bottom, which contains the SLA details (i.e. those related to the Guarantee Terms according to Figures 5 and 6).

The tool operation begins with importing the XPDL file that represents the business process model generated and exported by the BLA@BPMN, which includes all possible BLAs linked with such a model. Then, the list of BLAs present in the process model is displayed in BLA List field. Thus, any BLA listed may be chosen by the specialist so that the SLAs are derived for it.

According to the derivation rules presented in Table 1, the fields SLA Name, Expiration Time, Agreement Initiator and Agreement Responder are automatically filled with information from the imported XPDL file, in line with each selected BLA. This information is ready only in the tool.

The Services list is similarly filled with information from the XPDL file: all activities that have been selected as part of the sub-process under jurisdiction of a BLA are presented as potential services to be implemented as web services. The processing of this type of information in the input file needed to consider details of the language used by the BLA@BPMN tool: for example, the BLAs are linked to activities in the BPMN model using graph coordinates X and Y. Once incorporated by the BLA2SLA tool, these activities have the "WS-" prefix added to their names.

Given that not all activities present in the process model should be computationally executed via web services, the specialist may disregard such activities marking them as Not a service. This action can be useful, for example, to disregard activities from the process model that will be manually executed or via

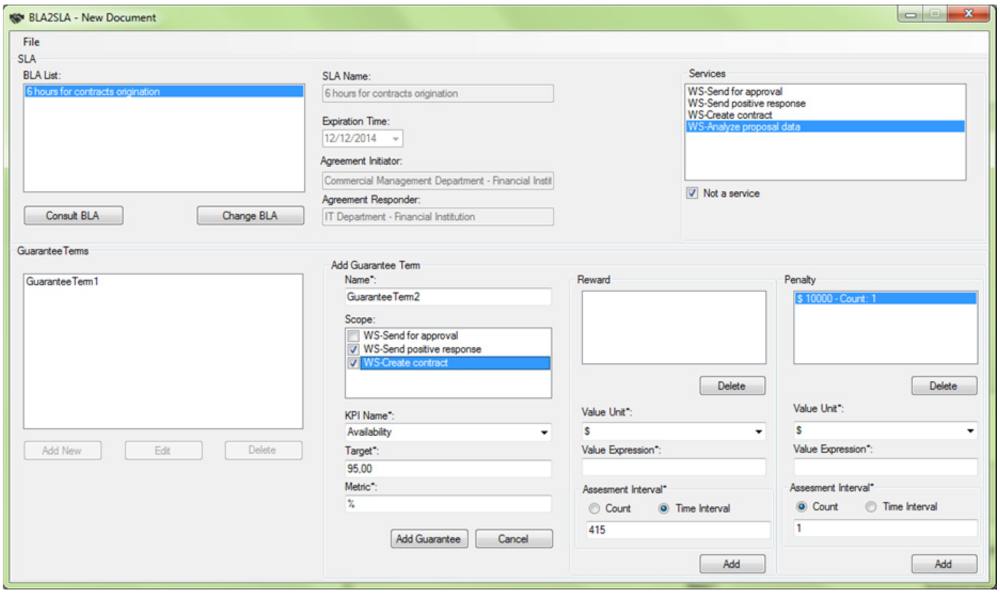

Figure 8: BLA2SLA prototype tool – main screen with an illustrative example.

script; since in this case this approach does not assumes a SLA for them in terms of SOA context. Activities marked as Not a service are not considered by the tool during definition of the Guarantee Terms. For purposes of SLA integrity control, only activities not currently used to create Guarantee Terms may be marked as Not a service.

For each BLA, it is expected that the specialist defines a set of SLAs (i.e. a set of guarantee terms) according to his or her knowledge about this BLA-SLA mapping and the business-IT involved context. The tool helps the specialist to find, in a visual and systematic way, the best set of SLAs that together will cover the respective BLA.

The tool presents a list of the GuaranteeTerms already created for the selected BLA. The existing ones can be edited or new ones can be created. For both cases, a list of web services is presented in the Scope list to be selected or unselected for the scope of the GuaranteeTerm, which means for which web services it applies. Activities marked as Not a service are not presented in the Scope list.

For each Guarantee Term, and all services selected as its scope, information related to the Service Level Objective should be defined. Such information includes: Guarantee Term Name, KPI Name, Target and Metric. CustomServiceLevel option is not available in this tool version. None of this information is direct created by the tool, i.e. they need to be defined by the specialist based on the selected BLA, whose details can be viewed in this tool for reference. For KPI Name, the tool allows the specialist to choose one of those QoS attributes presented in Table 3, and once it is chosen the respective metric unit is presented in Metric. These QoS attributes are physically stored in the structure of the Variable class (see Figure 7). Target must be defined by the specialist.

Also for each Guarantee Term, and the KPI Target (Service Level Objective), a set of Rewards and a set of Penalties can be defined by the specialist. The tool lists all already defined rewards and penalties, which can be viewed or deleted. To add either new rewards or penalties, the specialist must define their Value Unit and their Value Expression. All currency values used in the selected BLA are made available for selection for Value Unit, although any other currency value can be used. Moreover, the specialist must define the rules for the Assessment Interval in terms of Count or Time Interval. In this case, values are suggested according to those proposed in Table 2, and can be changed according to decision of the specialist. The derisions being taken in these cases, by the specialists, should be done considering the selected BLA, whose details can be viewed anytime.

In the illustrative example showed in Figure 8, there is a SLA (or Guarantee Term) is presented, called "GuaranteeTerm1". Moreover, a new SLA, called "GuaranteeTerm2" is being added. "GuaranteeTerm1" has the following associated data (not presented in Figure 8):

- Web service(s) in Scope: "WS-Send for approval";
- KPI Target (Service Level Objective): Response Time in 30 minutes;
- Penalty: $ 5,000.00 [to be charged to each service invocation].

"GuaranteeTerm2", in turn, is being defined with the following data (presented in Figure 8):

- Web service(s) in Scope: "WS-Send positive response" e "WS-Create contract";
- KPI Target (Service Level Objective): Availability in 95,00%;
- Penalty: $ 10,000.00 [to be charged to each service invocation].

Still for this specific case of BLA, a third example of SLA to be added could have the following data (not presented in Figure 8):

- Web service(s) in Scope: "WS-Send positive response" e "WS-Create contract";
- KPI Target (Service Level Objective): Availability in 99,00%;
- Reward: $ 10,000.00 [to be charged to each service invocation].

The tool allows that defined SLAs can be either saved and loaded again or else exported in the WS-Agreement format. If there are more than one BLA in the process model imported via XPDL file, the tool creates a WS-agreement for each BLA during the export procedure.

The following attributes of WS-Agreement language were not incorporated by the tool BLA2SLA: Location (Variable class), Importance (BusinessValueList class), ServiceTermReference e and Utility (Preference class), as presented in Section 5. Thus, these attributes are given the default value "[to be defined]" at the time of SLA generation in WS-Agreement specification. Therefore, an XML file is created in accordance with the official WS-Agreement scheme, which allows the IT area subsequently specify this information as needed.

6 EVALUATION OF BLA2SLA

An experiment was conducted with specialists to analyze the StrAli-BPM approach, including BLA2SLA. According to Sánchez *et al*. (2010), an experiment helps determine the effectiveness of methods and approaches proposed in a previous study, and by reducing uncertainties about the associated theory. Travassos, Gurov and Amaral (2002) state that an experiment should be supported by measurements so that the initiative that is carried out can be characterized, and that these can be both qualitative and quantitative.

The Goal-Question-Metric (GQM) technique was used to systematize the experiment: this is an experimentation process based on the phases of definition and interpretation (Travassos, Gurov and Amaral, 2002). The definition phase adopted a top-down approach: setting goals, formulating questions and defining the metrics. The interpretation phase used a bottom-up approach: measuring the results of the experiment in terms of metrics, formulating answers to the defined questions and grouping the answers to demonstrate the success of the goals.

Six employees of a large scale Brazilian financial institution were involved in the experiment. Only specialists in the application field of StrAli-BPM were chosen to take part in the experiment. Moreover, these specialists represent different management levels (strategic, tactical and operational) as well as both of the different sides – business and IT. The experiment involved a formal introduction and a practical handling of StrAli-BPM and associated tools, including BLA2SLA.

The goals were defined with the aim of understanding the following, from the standpoint of the specialists: level of contribution, practical applicability, effectiveness, viability, usability of the prototypes, and limitations. Questions and metrics were defined to refine these goals so that a roadmap could be derived with four qualitative questions and six quantitative questions to conduct focused semi-structured interviews (Yin, 2005).

As a result of the experiment, a series of data from analyses were collected for StrAli-BPM in general, as well as for BLA@BPMN and BLA2SLA in particular. The results that are solely related to BLA2SLA can be highlighted as follows:

- In terms of how much BLA2SLA contributes towards achieving SLAs that meet the goals set up by business areas; the specialists stated that it provided significant advantages over ad-hoc approaches. On a scale from 1 to 5, the average of their responses was 3.7;
- In terms of the ease of creating SLAs from the goals established in BLAs that are combined with business process models; the specialists thought that BLA2SLA has a good degree of usability. On a scale from 1 to 5, the average of their responses was 3.5;
- The results of the qualitative questions, in general, showed that: on the one hand, BLA2SLA performed very well in the evaluation, and the specialists pointed out that it had a number of potential benefits and advantages; on the other hand, the same specialists also noted a number of limitations since they thought that as the organizational environment of a large-scale company is quite complex, such an approach must be fully mature to allow it to be used effectively.

7 CONCLUSIONS

The StrAli-BPM approach advances the state of the art in the Strategic Alignment between business and IT areas by suggesting that SLAs can be established from BLAs. The BLA2SLA part outlined here was carried out from the BLA@BPMN part, which was previously discussed by Salles *et al*. (2013).

The StrAli-BPM approach is sufficiently general to meet a wide range of needs in business areas and IT infrastructure, since organizations have peculiar features in terms of process modelling, SOA paradigm implementation and KPIs with business-aware capacities. In Salles *et al*. (2013), it has been shown that the use of BLAs and KPIs essentially makes it easier to identify NFPs in the context of BPM, at the process level.

In addition to the mapping to BPMN elements, the BLA structure was prepared to enable the subsequent semi-automatic derivation of SLAs, since the BLA@BPMN prototype tool exports the BLA structure, with all its attributes and values, to the XML format. In view of this, we decided to move towards with this approach, by accommodating the derivation of BLA in SLAs, as well as by developing and evaluating its computational support in greater depth.

The results of the experiment that was conducted to evaluate the whole StrAli-BPM show that this BLA2SLA part was well assessed by all the specialists involved. In summary, they reported that they ignore any approach similar to BLA2SLA and considered it to be an important step towards achieving strategic alignment in organizations.

In future work, we would like to improve the BLA2SLA approach by extending the direct mapping between BLA and SLA attributes, whenever possible. Moreover, we would like to test it by carrying out a controlled experiment in which users could try it out in a case study.

ACKNOWLEDGEMENTS

We thank the State of São Paulo Research Foundation (Fapesp), Brazil, for its support.

REFERENCES

Allen, P., 2006. Service Orientation: Winning Strategies and Best Practices. *Cambridge Univ. Press, New York.*

Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M., 2007. *Web Services Agreement Specification (WS-Agreement)*. World Wide Web Consortium (W3C).

Bratanis, K., Dranidis D., Simons A. J. H., 2010. Towards run-time monitoring of business-level agreements for web services. *Proc. of the Annual South-East European Doctoral Student Conf., pp. 370.*

Bruin, T., Rosemann, M., 2010. Towards understanding strategic alignment of business process management, *Proc. of the Australasian Conf. on Information Systems (ACIS).* Adelaide, pp. 1.

Caseau, Y., 2005. Self-Adaptive Middleware: Supporting Business Process Priorities and Service Level Agreements. *Advanced Engineering Informatics, 19th ed., pp. 199-211.*

Friedenstab, J., Janiesch, C., Matzner M., Müller, O., 2012. Extending BPMN for Business Activity Monitoring. *Proc. of the 45th Hawaii Int. Conf. Syst. Sciences (HICSS), pp. 4158–4167.*

Garcia, D. Z. G., Toledo, M. B. F., 2006. Semantics-enriched QoS Policies for Web Service Interactions. *In: 12th Brazilian Symposium on Multimedia and the Web (WebMedia - ACM), Natal, Brazil,35-44.*

Goel, N., Kumar N.V., Shyamasundar R.K., 2011. SLAMonitor: A System for Dynamic Monitoring of Adaptive Web Services. *IEEE Europ. Conf. on Web Services, 109–116.*

Hevner, A.R., March, S.T., Park, J., Ram, S., 2004. Design Science in Information Systems Research. *MIS Quarterly 28, 75–105.*

Khaluf. L., Gerth, C., Engels, G., 2011. Pattern-Based Modeling and Formalizing of Business Process Quality Constraints. *Proc. of the Int. Conf. on Advanced Infomartion Systems Engineering (CAiSE), London, pp. 521.*

Laudon, K. C., Laudon, J. P., 2009. *Management Information Systems*, PrenticeHall. 11th edition.

Mazanek, S., Hanus, M., 2011. Constructing a bidirectional transformation between BPMN and BPEL with a functional logic programming language. *J. Visual Lang. and Comp. 22, 66–89.*

OMG, 2011. Business Process Model and Notation (BPMN) Specification, Version 2.0. Object Management Group.

Osterle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., Sinz, E.J., 2011. Memorandum on design-oriented information systems research. *Europ. J. Inf. Syst. 20, 7–10.*

Parmenter, D., 2010. *Key Performance Indicators (KPI): Developing, Implementing, and Using Winning KPIs*, Wiley. 2nd edition.

Pavlovski , C. J., Zou, J., 2008. Non-functional Requirements in Business Process Modeling. *Proc. of the Asia-Pacific Conf. on Conceptual Modelling (APCCM), Darlinghurst, pp. 103.*

Pourshahid, A., Amyot, P., Peyton, L., Ghanavati, S., Chan, P., Weiss M., Forster, A.J., 2009. Business Process Management With the User Requirements Notation. *Electronic Commerce Research, 269–316.*

Pressman, R., 2009. *Software Engineering: A Practitioner's Approach*, McGraw-Hill. 7th edition.

Salles, G. B. M., Fantinato, M., Nishijima, M., Albuquerque, J. P., 2013. A Contribution to Organizational and Operational Strategic Alignment: Incorporating Business Level Agreements into Business Process Modeling. *Proceedings of the 10th Int. Conf. on Services Computing (SCC). IEEE Computer Society Press, pp. 17-24.*

Sánchez, L., Garcia, F., Mendling, J., Ruiz, Z, F., 2010. Prediction of Business Process Model Quality based on Structural Metrics. *In: 29th Int. Conf. on Conceptual Modeling (ER), Vancouver, Canada, p. 458-463.*

Shimizu, T., Carvalho, M., Laurindo, F. J. B., 2006. *Strategic Alignment Process and Decision Support Systems: Theory and Case Studies*, IRM Press. 1st edition.

Sommerville, I., 2010. *Software Engineering*, Addison-Wesley. 9th edition.

Theilmann, W., Winkler, U., Happe, J., Magrans, I., 2010. Managing on-demand business applications with hierarchical service level agreements. *LNCS, vol. 6369, pp. 97–106.*

Travassos, G. H., Gurov, D.; Amaral, E. A. G. G., 2002. Introdução à Engenharia de Software Experimental. Relatório Técnico, COPPE-UFRJ (in Portuguese).

Weske, M., 2007. Business Process Management: Concepts, Languages, Architectures, *Springer*.

Wetzstein, B., Leitner, P., Rosenberg, F., Brandic, I., Dustdar S., Leymann, F., 2009. Monitoring and Analyzing Influential Factors of Business Process Performance. *In: 13th Enterprise Distributed Object Comp. Conf. (EDOC), pp. 141–150. Auckland.*

Yin, R. K., 2005. *Estudo de Caso: Planejamento e Métodos*, Bookman. 3rd edition, pp. 212 (in Portuguese).