Information and Software Technology 72 (2016) 171–188

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

# Managing the alignment between business processes and software systems

Lerina Aversano, Carmine Grasso, Maria Tortorella*

*Department of Engineering, University of Sannio, Via Traiano 1, Benevento, 82100, Italy*

ARTICLE INFO

*Article history:*
Received 24 April 2015
Revised 17 December 2015
Accepted 17 December 2015
Available online 13 January 2016

*Keywords:*
Software evolution
Business process modeling
Alignment evaluation
Software quality

ABSTRACT

*Context:* The alignment degree existing between a business process and the supporting software systems strongly affects the performance of the business process execution. Methodologies and tools are needed for detecting the alignment level and keeping a business process aligned with the supporting software systems even when they evolve.
*Objective:* This paper aims to provide an adequate support for managing such a kind of alignment and suggesting evolution actions if misalignment is detected. It proposes an approach including modeling and measuring activities for evaluating the alignment level and suggesting evolution activities, if needed.
*Method:* The proposed approach is composed of three main phases. The first phase regards the modeling of business process and software systems supporting it by applying a modeling notation based on UML and adequately extended for representing business processes. The second phase concerns the evaluation of the alignment degree through the assessment of a set of metrics codifying the alignment concept. Finally, the last phase analyses the evaluation results for suggesting evolution activities if misalignment is detected.
*Results:* The paper analyses the application of the proposed approach to a case study regarding a working business process and related software system. The obtained results provided useful suggestion for evolving the supporting software system and improving the alignment level existing between them and the supported business process.
*Conclusion:* The approach contributes in all phases of the process and software system evolution, even if its improvement can be needed for identifying the impact of the changes. The proposed approach facilitates the understanding of business processes, software systems and related models. This favors the interaction of the software and business analysts, as it was possible to better formulate the interviews to be conducted with regard to the objectives and, thus, to collect the required data.


## 1. Introduction

Issues regarding the alignment were mentioned for the first time in the late 1970s and, since then, studies and researches were conducted highlighting the alignment concerns [52]. In addition, researchers, practitioners and companies demonstrated through case studies, surveys and empirical approaches that the business and IT (Information Technology) performances are tightly coupled [7,12,18,23,25,31,33,46,54], and enterprises cannot be competitive if their business and IT strategies are not aligned. Indeed, high degree of alignment positively influences IT effectiveness and leads to higher business performance. Nevertheless, in the best of the authors' knowledge, industry has only marginally addressed these aspects in a quantitative way, while the evaluation of the alignment represents a top concern issue.

In the literature, alignment is described at two different abstraction levels, i.e. strategic and functional [15,30], and it involves different concepts, such as enterprise goals, business entities, strategies and processes, technology, information system and data. In [3,4], the authors proposed a literature analysis highlighting that the large part of the proposed approaches are mainly focused on the modeling of business and software assets and few attention is paid to the evaluation of the alignment degree and identification of the evolution tasks to be performed for improving it. Actually, software engineers very often have to deal with cases in which misalignment occurs, and, as a consequence, a business process is not effectively supported by the used software systems. Misalignment may be due to a misuse of a software system or rapid changes of the operating environment [58]. It must be detected and evolution activities must be identified and executed.

* Corresponding author. Tel.: +39 0824305554.
*E-mail addresses:* aversano@unisannio.it (L. Aversano), grasso@unisannio.it (C. Grasso), tortorella@unisannio.it (M. Tortorella).

http://dx.doi.org/10.1016/j.infsof.2015.12.009

They can involve one or more entities of the analyzed business process, which are mainly activities and components of the supporting software systems. The authors performed a study pointing out that evolution activities are more effectively performed if the business process using a software system to be changed and the software system itself are known [6]. This requires the continuous monitoring of the existing alignment degree between business processes and software systems and involves measurement activities to be continuously executed in the operative business context, for catching possible misalignment and detecting the needed evolution activities.

This paper proposes an approach for managing, evaluating and maintaining the alignment between a business process and the supporting software systems at the functional level. The definition of alignment considered in this paper refers to the linking existing between business entities and software components. A business process is **functionally aligned** with a set of software systems if the measure of their alignment achieves an established threshold. Given the model of a business process and one representing the software systems supporting it, the measure of their alignment indicates at which extent each business entity (activity, artefact or operation) is supported by a set of software entities (software components), which have the same intended meaning. A more formal definition of the used concepts concerning the alignment is given in following section.

This paper introduces different and complementary aspects of the alignment concept, relying on: (i) the use of a modeling notation to represent the software systems and business processes; (ii) the definition of the types of links existing between these models; (iii) the quantitative analysis of the alignment relation; and (iv) the exploitation of the analysis results for identifying evolution changes. The approach the paper proposes is based on the coarse grained strategy proposed in [2], where the technological coverage and technological adequacy were first introduced. The paper improves the preliminary proposed solution by introducing a new set of metrics allowing the evaluation of different and complementary aspects of the alignment relationship. The new metrics consider both static and dynamic aspects of the modeled software and business entities and their reciprocal relationships. This paper also improves the used modeling notation, and includes additional concepts for better relating all the involved business entities.

The rest of the paper is organized as follows: Section 2 discusses the analysis of the literature, outlines the motivations of the approach definition and provides the definition relating the considered alignment concepts; Section 3 describes the proposed approach for the alignment evaluation and management; Section 4 presents the software tool supporting the application of the approach; Section 5 discusses a case study aiming at highlighting the applicability of the approach; and final remarks are given in the last section.

## 2. Literature review

A unique definition of the alignment does not exist in literature. Various alternative terms are used for referring to it [17]: fit in [19,30]; bridge in [22]; integration in [30]; harmony in [39]; linkage in [48]; fusion in [53]; and further definition and terms are in [17].

The literature regarding the alignment topic differs in the abstraction level and treated aspects it considers [58,59]. A study of the literature allowed to classify the papers addressing alignment in two main groups regarding *strategic alignment* and *functional alignment*. Strategic alignment of IT exists when goals, processes and activities of a business organization are in harmony with the information systems supporting them [41]. The functional level regards the alignment existing between business processes and software systems and aims at optimizing the effectiveness of the software support during the business process execution.

The next section describes the analyzed papers, while the subsequent one discusses preliminary results of a performed quantitative review, and provides the definition of alignment considered in the proposed approach, and additional needed definitions.

### 2.1. Alignment studies

The literature analysis regarding the alignment is described in the following two subsections. The first subsection concerns strategic alignment, while the second one regards functional alignment.

#### 2.1.1. Strategic alignment

In [45], the information technology alignment requires the definition of a planning process, helping to identify IT strategies, IT projects and Information Systems from the requirements of each business unit and to create links between the strategic and operational levels. In [50,51], a view of business and technological alignment is defined considering at which extent the business mission, objectives, and plans are supported by information technology mission, objectives, and plans. Concepts of “fit” and “integration” among business strategy, IT strategy, business infrastructure, and IT infrastructure are discussed in [30,44].

The Business and Information Systems MisAlignment Model – BISMAM – is proposed in [16,57], to understand, classify and manage misalignments in medical science context and uses a metaphor between misalignment and disease. In particular, the BISMAM modeling approach establishes a misalignment classification scheme that links enterprise architecture views, misalignment symptoms and causes, and defines techniques to detect, correct and prevent misalignments.

The association between strategic alignment, management of ERP projects, business process changes and business performance of ERP systems is analyzed in [60]. The results showed that the more the ERP system strategy is aligned with the business strategy, the more likely the ERP project is completed on budget and on time.

The IT requirement analysis can also impact the organization alignment level. In [13], a requirement engineering framework is presented that enables verification and validation of requirements in terms of alignment with the support of business strategies. In [42], business process models, normally used during the first stages of software requirements gathering, are considered as a valuable source for performing functional size measurement. SEAM – systemic enterprise architecture methodology – is a family of methods for strategic thinking, business/IT alignment, and requirement engineering [61,62]. In particular, SEAM for business is a method designed for analyzing the competitive environment of an organization, including its relationships with its customers, partners, and market regulators. SEAM also regards the alignment between enterprise environment and marketplace. The marketplace is also considered in [35], where the authors describe a multi-perspective framework and tool for supporting business IT alignment in a scenario of software service providers (e.g., ASP’s or web service yellow pages). The approach considers all the relevant dimensions for the sourcing choice, that is services available in the marketplace that are aligned with the organizational goals.

The platform S2AEA (strategic alignment assessment based on enterprise architecture) is presented in [26] for modeling enterprise architecture and assessing strategic alignment based on internal enterprise architecture metrics, such as the degree of consistency between business strategies and information systems. In [20], the dynamic capabilities perspective is applied to analyze the strategic information system alignment process and aids to understand how firms develop IT and align it with business strategy.

In [39], a model, based on CMM with five levels and some intermediate phases, is proposed for measuring the alignment between corporate strategies and information technology strategies. It considers the correlation existing between the maturity of IT-business alignment and IT organizational structure and organization performance.

A problem for the alignment management regards the recovering and elaboration of the needed information. In [56], the authors showed that the organizations are not always able to systematically evaluate whether alignment exists, mainly because of the lack of documentation on the strategic alignment. They propose an approach, called INtenional STrategic ALignment (INSTAL), for describing organizations' strategic objectives and information systems, aiming at documenting and analyzing the strategic alignment.

At the strategic level, a big effort has been performed for evolving the GQM – Goal, Question, Metric – paradigm toward the GQM+Strategies, defined for being applied in software-intensive organizations [10,11]. The GQM+Strategies allows linking high level business goals and strategies of a software organization to software-specific goals and measurement goals. Then, it supports the translation of the business goals and strategies in specific software measurement programs. The GQM+Strategy has been further exploited for considering business value analysis by integrating aspects of business value expressed by the goal owners and stakeholders [36].

*2.1.2. Functional alignment*

The cited approaches mainly concern the analysis of an organization's strategic alignment and propose new modeling techniques and/or use existing ones. Likewise, the papers dealing with the functional alignment mainly include modeling practices [4] and very often exploit modeling languages already known in literature, such as the unified modeling language (UML) and/or its modifications, or propose new modeling approaches. Few papers propose techniques for the evaluation of the alignment level. The modeling and alignment evaluation activities of the analyzed papers are described in the following.

*2.1.2.1. Modeling the alignment.* In [27], the author introduces the service oriented architecture modeling language (SoaML) and uses the business process model and notation (BPMN) language, and the business motivation model (BMM) language for defining the business motivation models. In [34], the author used the UML 2 profile for event driven process chains (EPCs) to connect the business with software requirements and components.

In [63], the authors propose ProQAM (process-oriented questionnaires for analyzing and modeling scenarios) for aligning business process models and software requirements. In [68], the double-state based business process description model (DSBPDM) is proposed for modeling applications called form-centered application systems (FAS), very important for the Chinese e-government affair systems. In [40], the authors deal with the functional alignment of business processes and ICT-infrastructure in health care organizations and use the dynamic essential modeling of organization (DEMO) methodology for representing the business processes and UML for modeling the ICT infrastructure. In [67], an approach is proposed for checking the alignment existing between a value web and the IT supporting its realization represented by an e3-value model and use case diagrams, respectively. In [48], the tool "Efficient" is used as automatic support for analyzing the alignment between business and software services. It uses a UML activity diagram for representing the choreography of business activities and information exchanged between them, and UML class diagrams for representing the structure of this information. In [14], it is claimed that the current business processes modeling approaches lack links with the ontological items, i.e., the tasks and business entities of information systems. These links are important because they permit to adequately represent the information flow and human decisions in the process model, otherwise missing. Then, the author proposes a notation, called TOMP task-oriented modeling of business processes to overcome that limitation.

In [38], Lonchamp uses the UML notation for modeling, and considers a modeling approach based on SPEM, software process engineering meta-model [55], from the OMG.

The alignment at the functional level is often used for identifying the software requirements of software systems, with reference to the service-oriented environments. In [19], the business-IT alignment capabilities are discussed to gather the benefits of the SOA (service oriented architecture) paradigm for developing service-based systems. In [24], a model-driven approach is presented to deal with the alignment between high-level business specifications and lower-level information technologies, that arises when adopting service-oriented approaches for software. Service-oriented development method (SOD-M) is considered to define a service-oriented approach for the development of information systems. In [64], the authors present a framework for analyzing the alignment problem and propose an approach to application architecture design with reference to the business context. In [8,9], the concept regarding enterprises business objects and their implementation as specialized services having separate concerns in the information system for guaranteeing enterprise agility to the evolution, is introduced.

The E-BPMS (business process management systems) modeling framework was discussed in [32], taking into account the requirements of a complex e-business application development. Different aspects of alignment were treated with reference to strategic decision, design, resource allocation, and performance evaluation processes. In addition, the use of some key indicators is proposed as linkage to E-strategy model. In [68], an approach is presented for automatically recovering business process entities from multi-tiered business applications, and for establishing links between the process and their software entities.

Finally, BPEL, XPDL, and BPMN are the most diffused notation for process modeling [43,49,65,66]. However, they are process definition languages, suitable for modeling a new business process, while the evaluation of the alignment requires a notation that is not executable but more close to the organizational view of the business processes already executed.

*2.1.2.2. Evaluating the alignment.* Besides modeling, an important goal of alignment analysis is understanding it. A way for reaching this goal is measuring the alignment level existing between the involved entities. Unfortunately, few researches addressing this aspect were developed.

Etien and Rolland proposed criteria and associated generic metrics to quantify at which extent there is a fit between the business process and software systems [28]. The authors showed how to derive a set of specific metrics from the generic ones. In [52], the author also addressed the evolution of the analyzed context and treated issues such as relationship modeling and refining, engineering class change identification, requirements change specification and propagation. In [37], some metrics were proposed for analyzing the alignment with reference to ontology.

Finally, the authors of this paper proposed in [1] a coarse grained strategy for measuring the alignment at the functional level and detecting misalignments. The alignment was measured by considering two attributes: Technological Coverage and Technological Adequacy. A more detailed analysis of these aspects is provided in [2], where an initial set of metrics is proposed.

**Table 1**
Alignment activities treated in the Science Direct papers.

| Journal | Count | Modeling approach | | | Alignment evaluation | | |
|---|---|---|---|---|---|---|---|
| | | Count | % | Prop. | Count | % | Prop. |
| Journal Information & Management | 16 | 12 | 75.0% | 21.1% | 5 | 31.2% | 12.8% |
| International Journal of Production Economics | 10 | 3 | 30.0% | 5.3% | 5 | 50.0% | 12.8% |
| Journal of Strategic Information Systems | 9 | 7 | 77.8% | 12.3% | 3 | 33.3% | 7.7% |
| Journal of Operations Management | 6 | 4 | 66.7% | 7.0% | 3 | 50.0% | 7.7% |
| Industrial Marketing Management | 6 | 4 | 66.7% | 7.0% | 2 | 33.3% | 5.1% |
| International Journal of Project Management | 6 | 2 | 33.3% | 3.5% | 3 | 50.0% | 7.7% |
| Expert Systems with Applications | 5 | 4 | 80.0% | 7.0% | 3 | 60.0% | 7.7% |
| Journal of Business Research | 5 | 2 | 40.0% | 3.5% | 1 | 20.0% | 2.6% |
| Information and Software Technology | 5 | 1 | 20.0% | 1.7% | 1 | 20.0% | 2.6% |
| Long Range Planning | 4 | 1 | 25.0% | 1.7% | 3 | 75.0% | 7.7% |
| Government Information Quarterly | 4 | 2 | 50.0% | 3.5% | 0 | 0.0% | 0.0% |
| International Journal of Information Management | 4 | 2 | 50.0% | 3.5% | 1 | 25.0% | 2.6% |
| Computers in Industry | 4 | 2 | 50.0% | 3.5% | 0 | 0.0% | 0.0% |
| Hospitality Management | 3 | 2 | 66.7% | 3.5% | 2 | 66.7% | 5.1% |
| Journal of Purchasing & Supply Management | 2 | 1 | 50.0% | 1.7% | 0 | 0.0% | 0.0% |
| Information and Organization | 2 | 0 | 0.0% | 0.0% | 0 | 0.0% | 0.0% |
| International Journal of Systems and Software | 2 | 1 | 50.0% | 1.7% | 0 | 0.0% | 0.0% |
| Decision Support Systems | 2 | 0 | 0.0% | 0.0% | 1 | 50.0% | 2.6% |
| Others | 19 | 7 | 36.8% | 12.3% | 6 | 31.6% | 15.4% |
| Sum | 114 | 57 | | 100% | 39 | | 100% |

### 2.2. Preliminary results of a quantitative review

In [3,4], the results of a preliminary quantitative review with reference to the alignment approaches were discussed. It regarded the most representative journals including papers concerning alignment. They were identified by examining the papers found through a manual inspection resulting by querying the Science Direct database. A characterization framework proposed in [3] was used for guiding the analysis.

The results published in [3,4] confirmed that the large part of the analyzed papers never provided an operative support for quantifying the alignment level. Moreover, no paper used the analysis results for identifying and implementing improvements of the operative context. These data are confirmed by the current research the authors are performing with reference to the papers published from 2003. The preliminary results are shown in Table 1 and concern 114 papers published till year 2014.

Table 1 shows that the large part of the analyzed papers regards the alignment aspect with reference to the modeling task and uses different modeling languages. A closer analysis showed that the most popular language used to model alignment is UML, associated with other languages for representing the business processes. Few papers address the evaluation task and none of them considers the evolution aspects. In particular, even if the table indicates some papers concerning the evaluation, the large part of them mainly discusses results of industrial surveys and does not propose metrics for quantitatively evaluating the alignment.

The analysis also showed that few papers proposed automatic tools for supporting the alignment treatment, but, even in this case, just the modeling of the involved entities were considered. Finally, the cited analysis highlighted that the research regarding the alignment at functional level is immature. The large part of the analyzed papers is theoretic and few of them completely face the alignment concept. Moreover they present only examples and simple case studies for analyzing the application of the proposed approaches.

This paper proposes an approach that models, evaluates and evolves the alignment concept. The aim is to overcome the problems above by considering the limitations identified in the other approaches. Also the proposed approach adopts UML, as the authors of the paper believe that it is a good candidate for its ease of use and dissemination. The UML language is suitably enlarged for modeling the business concepts and it is used for modeling both software systems and business processes for directly mapping their entities. Moreover, the presented approach proposes a set of metrics at different granularity, with the aim of identifying which business entities are less automatically supported, and understanding which changes need to be performed. An automatic support to help the application of the approach is also proposed.

The formal definition of alignment considered in the proposed approach requires the specification of what is intended as business process and software system.

A *business process* is a collection of related, structured activities or tasks and produces a specific service or product. Each activity is executed by specific actors and manipulates and produces specific artifacts. The term artifact is used for pointing out information considered in the business process at any granularity level: variable, structured and unstructured documents, images, and so on. Then, the set of business entities regards *business activities, actors* and *artifacts*. The relationships existing among the business entities are: dependence relations among the activities, as an activity can be executed when other activities are completed; execution relations among actors and activities, as one or more actors executes one activity; and use relations among activities and artifacts, as one activity uses one or more artifacts.

A *software system* is intended as a set of software components (i.e., packages, classes, functions, methods, variables, and so on) and related relationships, that together implement a set of services and functionality that can be useful for helping the execution of the business activities. The relationships existing among the software components can be of three types: functional, data or control.

As already stated in the introduction, the alignment between a business process and the supporting software systems is defined as a one-to-many function:

$$alignment : BusinessEntities \rightarrow SoftwareComponents*$$

where:

- *BusinessEntities* is the set of all the entities of the business process;
- *SoftwareComponents* is the set of all the software components of the supporting software systems.

Thus, ***functional alignment*** maps each element of *BusinessEntities* to a subset of elements of *SoftwareComponents*, also the empty

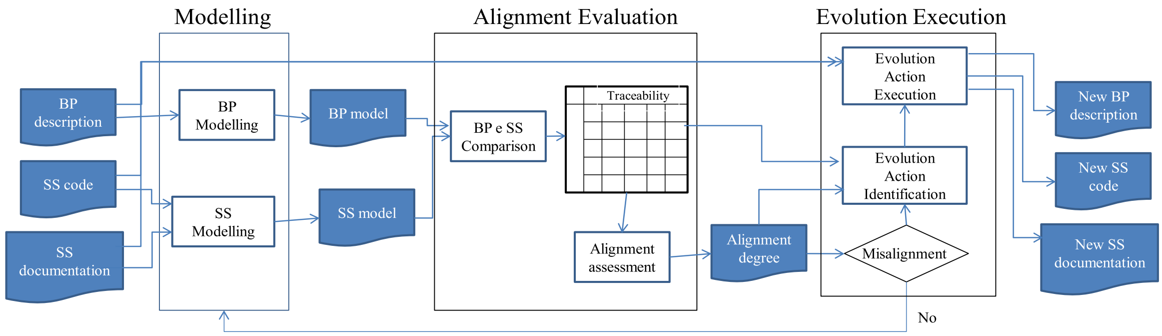



**Fig. 1.** Phases of the proposed approach.

set when a business entity does not correspond to (i.e., it is not supported by) any software component.

## 3. Proposed approach for the alignment evaluation and management

The proposed approach is structured in three phases as depicted in Fig. 1. They are the following:

1. **Modeling.** The models of the business processes and supporting software systems are defined, through the representation of all their entities. A big attention is paid to the business entities that are useful for indicating software functionality to be implemented for improving the alignment level.
2. **Alignment evaluation.** A set of metrics is proposed to objectively measuring the alignment level existing between the business processes and supporting software systems. The alignment assessment requires the comparison between business processes and software systems and the elaboration of a mapping between software and business entities. This mapping is represented through a traceability matrix that is obtained by performing a semantic analysis of the information encoded in the models [5].
3. **Evolution execution.** If the previous phase highlights that the business processes and supporting software systems are not aligned, the evolution actions to be performed for increasing the alignment level are identified and executed. This could require the implementation of new software classes and/or methods, and/or modification/deletion of the existing software components.

The iteration of these phases allows a continuous monitoring and management of the alignment level. In the following, the modeling and alignment evaluation phases will be described.

### 3.1. Modeling

The modeling phase exploits the constructs of the UML language and enhances and combines them for identifying the relation existing between the business processes and supporting software systems. The aim is to identify all the software functionality that should be modified and/or implemented for adequately supporting the business processes. This activity should be executed when the alignment degree needs to be evaluated and each time evolution actions are required.

A software system is typically modeled by using class, state, sequence and use case diagrams provided by the UML language. A business process can be described by activity diagrams that allow to model: activities performed during the process execution; actors executing the activities; control flow among the different activities; data flow indicating the input required to execute each activity and output it produces. Nevertheless, the activity diagram is not enough for depicting all the concepts involved in the alignment. For this reason, additional diagrams have been introduced. The following of the section describes the proposed additional notations.

#### 3.1.1. Activity and states

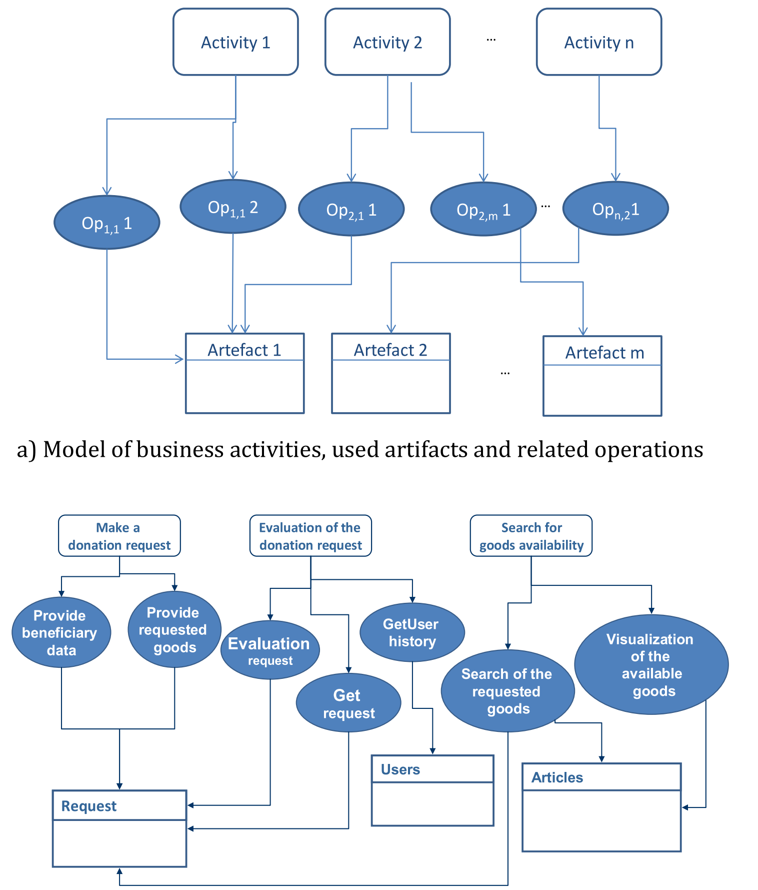
The basic business process model is integrated to highlight the state transition that happens when the activities are performed. In particular, the states are referred to the artifacts manipulated in the process execution, and exchanged as input/output among the activities. Fig. 6 shows an activity diagram of the analyzed case study. It also highlights the change of the artifact states after the execution of an activity.

#### 3.1.2. Activities, operations and artifacts

An activity of the process model represents a part of the work performed during the process execution, by using some business artifacts. Then, the description of a business activity also requires the identification of the actors executing it, artifacts it manages and operations it executes. In the proposed notation, a business artifact is modeled as a class. A business activity is associated to the artifact classes through operations corresponding to the elementary tasks the activity performs on them. Fig. 2a shows the diagram depicting the relationships existing between activities, class artifacts and operations. It shows that *Activity 1* is just associated to *Artifact 1* through operations $Op_{1,1}1$ and $Op_{1,1}2$, while *Activity 2* is associated with *Artifacts 1* and *m* by means of $Op_{2,1}1$ and $Op_{2,m}1$, respectively; and so on. The diagram can also include the actors performing the activities. Fig. 2b reports, as an example, a portion of diagram of the discussed case study. In particular, the arrows connecting operations and artifacts in Fig. 2b indicate the use/definition relations between them. Thus, activity *Make a donation request* performs the operations *Provide beneficiary data* and *Provide requested goods* on the *Request* artifact, and activity *Evaluation of the donation request* executes operations *Evaluation request* and *Get request* on the *Request* artifact and operation *GetUser history* on the *User* artifact, and so on.

#### 3.1.3. Activities and artifacts

The diagram in Fig. 2 can be synthesized in the diagram of Fig. 3, which models the coupling of an activity with the business artifacts it uses. Fig. 3a shows the generic diagram instantiated in Fig. 3b by considering the diagram of some activities of the case study. Fig. 3 shows that a relation exists between an activity and an artifact if the activity performs at least one operation on the artifact. The set of operations indicated in an artifact of Fig. 3 is the set of all the operations performed on that artifact by all the activities. Thus, the proposed notation allows to model the business processes in terms of executed activities and used artifacts. This diagram can include the actors executing the activities. If a software




a) Model of business activities, used artifacts and related operations

b) Example of model with business activities, used artifacts and related operations

**Fig. 2.** Models of activities, artifacts and operations.

system is modeled by using a UML class diagram including a set of classes implementing a set of methods, a comparison between the model in Fig. 3 and the software system class diagram can indicate the business operations that are not implemented by the software system.

#### 3.1.4. Goals, artifacts and states

A business goal is a set of stable states that have to be valid after the business process is executed. A goal of a process is represented in an activity diagram and formulated as a Boolean function in the notation LTL (linear temporal logic), concerning the states of a set of artifacts and other states that have to be reached at the end of the process. As an example a goal of the presented case study is *Managing user*. It can be represented as: $((User.state_t=\text{'created'})\vee(User.state_t=\text{'recovered'}))\wedge(User.state_{t+1}=\text{'notified'})$, that indicate that the management of a user is performed if he has been created in or recovered from the database, and then a notification has been sent to him. In general terms, the goals are formulated combining initial and final states of the artifacts.

### 3.2. Alignment evaluation

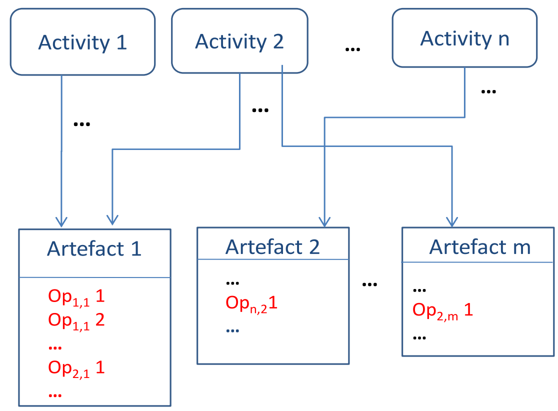
The alignment evaluation entails the measurement of a set of metrics used to obtain an objective value of the alignment degree. This requires the identification of the links existing between business and software entities. These links can be obtained by performing a semantic analysis on the business and software models. The two following subsections describe the semantic analysis to be performed for constructing a traceability matrix and the way to assess the alignment level.

#### 3.2.1. Semantic analysis

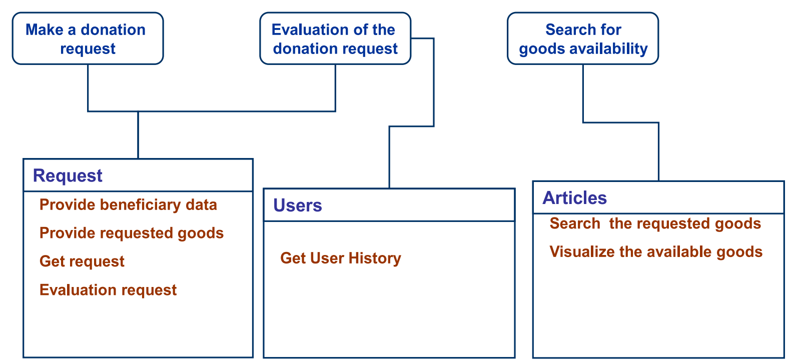
The semantic analysis is aimed to retrieve the traceability links between the business process activities and supporting software system components. It relies on the extraction of the identifiers from the business process and software system models and on the identification of the software components involved in a business activity. It is based on two processing phases:

- *Semantic Processing* regards the extraction of semantic information from the business process and software system models and the source code;
- *Matching Processing* aims at discovering the matching existing between the business information and software system components.

The *Semantic Processing* phase receives as input the business process model containing all the relevant information regarding the business activities, extracts all the identifiers composing the name of each activity and enriches them with semantic information. Its aim is to improve and complete the description of each business process activity, enhancing it with a set of additional information that can be extracted from a lexical ontology. A pre-processing of the extracted information is first executed by performing the following steps: *Tokenization*, to split the source text in tokens and transform them in lowercase letters; *Stopword elimination*, to eliminate the useless terms, such as articles, programming language keywords, and so on; *Lemmatization*, to remove inflectional endings and consider just the base or dictionary form of a word (named lemma); *Stemming*, to chops off the ends of words. The enhancement of the business process activities with additional




a) General diagram with business activities, and used artifacts




b) Instantiation of the diagram with business activities and used artifacts

**Fig. 3.** Models of activities and artifacts.

information is performed by exploiting WordNet[1] for retrieving a set of terms semantically related to each activity. The retrieved terms are synonymous and words contained in the description of the identifiers. If an identifier is a verb, the corresponding nouns are also extracted. The result is the business process (BP) vocabulary.

The *Matching Processing* phase is aimed at verifying the similarity between the syntactical and semantic information previously extracted, and discovering links between business process activities and software components. The result is a traceability matrix highlighting the software components potentially used during the execution of each business process activity.

Evaluating the similarity between couples of terms depends on the adopted information retrieval – IR – model. The proposed approach uses a vector space IR model [29]. It considers the elements of the business process model and software system components organized as vectors in a $n$-dimensional space, where $n$ is the number of the indexing features. These features are the identifiers of the entities extracted from the BP Vocabulary. The similarity criteria are used for identifying links between business process activities and software model entities, and these ones and software source code components. In particular, first, the product of the probabilities that each identifier contained in an entity node of the BP Vocabulary appears in a business activity name is computed; then, the product of the probabilities that each identifier in an entity node of the BP Vocabulary is also an identifier of a source code component is evaluated. The probabilities will be referenced in the following as weights. They are computed by using the well-known IR metric called *term frequency–inverse document frequency* (*tf–idf*) [29]. According to this metric, the vector of weights of the BP Vocabulary terms over the BP activities, $w_a = [w_{i,j}]$, is made of the weights $w_{i,j}$, indicating how important is the $i$-th term for the $j$-th activity. It is derived from the *term frequency* measure, $\boldsymbol{tf}_{i,j,}$ of the $i$-th term in the $j$-th activity, and the *inverse activity frequency, id$\boldsymbol{f}_{ij}$*, of the $i$-th term in the whole set of activities.

In the same way, the weights vector of the terms over software components, $w_k = [w_{i,k}]$, is built by computing the frequency, $\boldsymbol{tf}_{i,k}$, of the $i$-th term in the $k$-th software component, and the *inverse activity frequency, id$\boldsymbol{f}_{ik}$*, of the $i$-th term in the entire set of software components. The term frequency $\boldsymbol{tf}_{i,j}$ is the ratio between the number of occurrences of the $i$-th word over the total number of terms contained in the $j$-th activity. The inverse activity frequency $id\boldsymbol{f}_{ij}$ is defined as follows:

$$idf_{ij} = \frac{\#Activities}{\#ActivitiesContainingTerm_i}$$

The vector element $w_{ij}$ is represented as $w_{ij} = \boldsymbol{tf}_{i,j} \times \log(id\boldsymbol{f}_{ij})$, likewise $w_{ik}$. The term $\log(id\boldsymbol{f}_{ij})$ represents a weight of the frequency of an entity identifier in an activity (software component): a higher weight means that an entity identifier is more specific to that activity (software component). By experience, it is assumed that an entity identifier which reaches a weight equal or greater than 0.35 against an activity (software component) description can be considered as related to the activity (software component), and, then, the correspondence can be considered.

### 3.2.2. Metrics for the alignment evaluation

Once the semantic analysis is completed, the output is used for assessing the alignment level. This involves the assessment of

[1] WordNet – http://wordnet.princeton.edu/.

**Table 2**
Metrics of technological coverage.

| Metric | Formula | Description |
|---|---|---|
| *ActivityCoverage* | $ActivityCoverage = \frac{\#BPAS}{\#BPA}$ | It measures the percentage of process activities supported by the software systems. |
| *BPA* represents the set of the business process activities<br>*BPAS* is the subset of the business process activities that are supported by the software systems.<br>*ActivityCoverage* is evaluated in terms of number of the business process activities supported respect to the total number of process activities. | | |
| *ActorCoverage* | $ActorCoverage = \frac{\#BPActorsS}{\#BPActors} = \frac{\#(BPActors \cap SSActors)}{\#BPActors}$ | It measures the percentage of actors involved in the business process and whose activities are supported by the software systems |
| *BPActors* is the set of the business process actors involved in the business process activities.<br>*BPActorsS* is the subset of Business Process Actors involved in the business process and whose activities are Supported by the software systems. *BPActorsS* is a subset of *BPActors* as some actors could be forced to manually execute their activities. In particular, the business actors supported by the software system are identified through the intersection of the sets of actors included in the software system design, *SSActors*, and *BPActors*.<br>*ActorCoverage* is computed as the ratio between the number of the actors supported by the software systems and the number of actors involved in the business process. | | |
| *ArtifactsCoverage* | $ArtefactsCoverage = \frac{\#BPAtfS}{\#BPAtf} = \frac{\#(BPAtf \cap SSAtf)}{\#BPAtf}$ | It measures the percentage of the business process artifacts implemented in the software systems. |
| *BPAtf* is the set of the business process artifacts used/defined in the business process activities and modeled in the business process model.<br>*BPAtfS* is the set of business process artifacts that are also implemented by the software Systems. *BPAtfS* is a subset of *BPAtf*, as some artifacts could not be implemented by the software systems. This subset is calculated as the intersection of the *BPAtf* set of business artifacts and the set *SSAtf* of the artifact implemented by the supporting software systems through classes modeled in the software system class diagram.<br>*ArtifactCoverage* is computed as the ratio between the number of the business artifacts used/defined by the software systems and the number of business artifacts of the business process. | | |
| *GoalCoverage* | $GoalCoverage = \frac{\#BPGS}{\#BPG}$ | It measures the percentage of process goals supported by the software systems. |
| *BPG* represents the set of the business process goals<br>*BPGS* is the subset of the business process goals that are Supported by the software systems.<br>*GoalCoverage* is evaluated as the ratio of the number of the business process goals supported and the total number of process goals. | | |
| *PathCoverage* | $PathCoverage = \frac{\#BPPathS}{\#BPPath}$ | It measures the percentage of process path supported by the software systems. |
| *BPPath* represents the set of the business process paths<br>*BPPathS* is the subset of the business process paths that are supported by the software systems.<br>*PathC* overage is evaluated as the ratio between the number of business process paths supported and the total number of process paths.<br>*# is used for indicating the cardinality of a set* | | |

the *Technological Coverage* (TC) and *Technological Adequacy* (TA) [1], by considering metrics regarding the links existing between business entities and software components. The evaluation requires the identification of the metrics to be measured for assessing those aspects from the point of view of the software engineer and the business analyst. Therefore, each of them is refined into metrics and is brought to the essential aspects involved in a process model, classified in activities and resources involved in the activity execution, such as human resources (*Actors*), and input/output data required for performing the activities (*Artifacts*).

For evaluating TC, the coverage level for each considered aspect of a business process is evaluated. Both static and dynamic aspects are considered. The metrics regarding the static aspects are: *ActivityCoverage, ActorCoverage, ArtifactCoverage*. While the dynamic behavior is considered by analyzing metrics *GoalCoverage* and *PathCoverage*. They are evaluated from the technological support point of view, expressed in terms of number of the automatically supported activities, actors, artifacts, goals and dynamic paths. For example, with reference to Fig. 3b, if only artifact *Articles* is implemented, just Activity *Search for goods availability* of the process represented in the figure is automatically supported. Then, the TC metric of the process assumes value 0.33. Table 2 lists the considered metrics and how to calculate them in terms of lower level metrics. Once the metrics are evaluated, TC is calculated as an average value. On the basis of the enterprise's needs TC can be calculated giving different weights to the metrics.

For evaluating TA, the adequacy of the support provided by the software systems to the business process is evaluated. Even in this case, static and dynamic concepts are analyzed. The considered static metrics are: *ActivityAdequacy, ArtifactAdequacy, ActorAdequacy*. Metric *GoalAdequacy* is analyzed for the dynamic aspect. The metrics regard the technological adequacy on each single business activity, artifact, actor and goal. For example, with reference to Fig. 3b, if only Operation *Search the requested goods* is implemented by the software system, Activity *Search for goods availability* is not completely automatically supported, and its technological adequacy assumes a value equal 0.5. Obtained the technological adequacy of each business entity, the value of TA for the full process is obtained by using the metrics and related formulas of Table 3.

Once evaluated the adequacy metrics, the final value of TA is evaluated as an average of the obtained values. Even in this case, different weights can be attributed to the metrics on the basis of business needs.

Once the assessment is completed, the values achieved for TA and TC have to be compared to the alignment objective of the organization in terms of desired software support. Thus, such thresholds need to be defined:

- *Thre_TC,* indicating the minimum level of *TC* that the organization aims to achieve;
- *Thre_TA*, indicating the minimum level of *TA* that the organization aims to achieve. Additional thresholds of the technological adequacy, $Thre\_TA_i$, have to be defined also for each activity *i*.

The values of the *Thre_TC* and *Thre_TA* have to be indicated by both organization and process executors. They do not necessarily reach the highest level, as the business processes can include activities that cannot be automatically executed. If it occurs that $TA < Thre\_TA$ or $TC < Thre\_TC$, then changes requirements on the system have to be identified to increase the alignment level. Similarly, a change can cause $TA < Thre\_TA$ or $TC < Thre\_TC$. This can be due to a management (introduction and/or modification of an activity) or technological (changes in the used software system) innovation and it can decrease the technological coverage and adequacy, potentially causing a misalignment. If this happens, the comparison with the thresholds is performed at the activity level. For example, the change performed on activity *i* can cause the decreasing of the technological adequacy for that activity, that is $Thre\_TA_i > TA_i$. This indicates misalignment of activity *i*. If the change does not decrease the values *TA* and *TC*, the alignment is preserved. For

**Table 3**
Metrics for evaluating the technological adequacy.

| Metrics | Formula | Description |
|---|---|---|
| *ArtifactAdequacy* | $ArtefactsAdequacy = \frac{\sum_{\forall i \in BPAtfS} AtfA_i}{\#BPAtf}$<br>where each component $AtfA_i$ is defined as:<br>$AtfA_i = \frac{\#BOAtfS_i}{\#BOAtf_i} = \frac{\#(BOAtf_i \cap SMClass_j)}{\#BOAtf_i}$ | It expresses how adequate is the automatic support offered by the software systems to the management of the artifacts of the business process |
| *ArtifactAdequacy* is evaluated as the average of the automatic support adequacy offered to each business artifact modeled in the considered process model. The business artifact includes as many methods as the operations performed by the business activities execution with reference to the considered artifacts, as modeled in the business process model.<br>$AtfA_i$ is the automatic support adequacy of artifact *i*. It is computed by considering the operations of the business artifact *i* that are implemented in the corresponding class of the supporting software systems. It uses the following metrics:<br>– $BOAtf_i$ is the set of business operations performed on business artifact *i*;<br>– $BOAtfS_i$ is the set of business operations performed on artifact *i* and implemented by the software system. $BOAtfS_i$ is calculated as the intersection of $BOAtf_i$ and the methods implemented in the corresponding software class $SM\text{Class}_j$ | | |
| *ActvityAdequacy* | $AA = \frac{\sum_{i \in BPAS} AA_i}{\#BPA}$ where each component $AA_i$ is defined as:<br>$AA_i = \frac{\sum_{j \in BPAtfS_i} AtfA_{ij}}{\#BPAtfA_i}$ | It expresses how adequate is the automatic support offered to the business activities. |
| *Activity Adequacy* is evaluated as the average of the adequacy of the automatic support offered to each business activity.<br>$AA_i$ is the adequacy of the automatic support to the execution of activity *i*. $AA_i$ is computed by considering the automatic support adequacy of each business artifact *j* used/defined in activity *i* and implemented by the software systems. It uses the following metrics:<br>_ $BPAtfS_i$ is the set of the artifacts used/defined in activity *i* and implemented by the software systems.<br>_ $AtfA_{ij}$ is the automatic support adequacy of each business artifact *j* used/defined in activity *i*<br>_ $BPAtfA_i$ is the set of business artifact used/defined in activity *i* | | |
| *ActorsAdequacy* | $ActorAdequacy = \frac{\sum_{i \in BPActorS} ActorA_i}{\#\text{BPActors}}$<br>where each component $ActorA_i$ is defined as:<br>$ActorA_i = \frac{\sum_{j \in BPAS_i} AA_{ij}}{\#BPA_i}$ | Actor adequacy expresses how adequate is the automatic support offered to the business actors. |
| *ActorsAdequacy* is evaluated as the average of the automatic support adequacy offered to each business actor.<br>$ActorA_i$ is the adequacy of the automatic support offered to actor *i*. It is computed by considering the automatic support adequacy of each activity *j* the actor executes and that the software system supports. It uses the following metrics:<br>_ $BPAS_i$ includes all the activities actor *i* executes and supported by the software system<br>_ $AA_{ij}$ is the automatic support adequacy of each business activity *j* executed by the actor *i*<br>_ $BPA_i$ is the set of business activity executed by the actor *i* | | |
| *GoalAdequacy* | $GoalAdequacy = \frac{\sum_i GoalAdequacy_i}{\#BPGoal}$<br>where $GoalAdequacy_i$ is defined as follows:<br>$GoalAdequacy_i = \frac{\sum_{ij} statePathAdequacy_{ij}}{\#statePath_i}$<br>where $StatePathAdequacy_i$ is defined as follows:<br>$statePathAdequacy_{ij} = \frac{\sum_{jk} ActivityAdequacy_{ijk}}{\#ActivityPath_{ij}}$ | It expresses how adequate is the automatic support offered to the achievement of the business goals. |
| *GoalAdequacy* is evaluated as the average of the adequacy of the automatic support for reaching each business goal. In particular, it considers all the possible paths that could occur for reaching each goal.<br>*#BPGoal* is the number of business process goals of the analyzed business processes<br>$GoalAdequacy_i$ indicates the technological adequacy of the technological solution with reference to *Goal i*. It uses the following metrics:<br>– $\#statePath_i$ is the total number of state paths making true Goal i. A path state that makes true a goal is a set of activities connecting a set of artifact states that verifies the logical expression of the goal.<br>– $statePathAdequacy_{ij}$ is the technological adequacy of the *j*th state path of *Goal i*. It considers the following metrics:<br>– $\#ActivityPath_{ij}$ is the number of activities belonging to $statePath_{ij}$<br>– $ActivityAdequacy_{ijk}$ is the adequacy of activity *k* of $statePath_{ij}$. | | |

example, this can happen if a not automatically supported activity is deleted or technological innovations increase both *TC* and $TA_i$, for some *i*.

## 4. A software support

An automatic support was implemented for facilitating the application of the proposed approach, and automatically carrying the modeling and measurement of the alignment existing between business processes and software systems. Fig. 4 shows the architecture of the implemented tool. It is composed of two macro-components: **Modeling** and **Alignment Evaluation Engine**. The figure also shows the technologies used for developing each component. The tool has been developed in Java as a plug-in of the Eclipse environment,[2] in particular, the Eclipse Modeling Framework (EMF),[3] a personalization of the OMG's MOF (Object Management Group MetaObject Facility),[4] was used. The Graphical Editing Framework (GEF)[5] has been used on the basis of EMF for creating the modeling components. All the components have been developed by using SWT/JFACE,[6] and the MySql Database has been used for data storage. Furthermore, WordNet Semantic DataBase is used for performing the semantic analysis embedded in the Alignment Evaluation Engine.[7]

The **Modeling** component aims at producing the needed models. It consists of two main modules: Business Modeling and

[2] ECLIPSE – Open Source IDE, independent development language but mostly provided in Java. www.eclipse.org/.

[3] EMF – Eclipse Modeling Framework Project (EMF) http://www.eclipse.org/modeling/emf/.

[4] MOF – The MetaObject Facility Specification http://www.omg.org/mof/.

[5] GEF (Graphical Editing Framework) – http://www.eclipse.org/gef/.

[6] SWT and JFace are Eclipse's graphical libraries – http://www.eclipse.org/resources/resource.php?id=191.

[7] WordNet – http://wordnet.princeton.edu/.

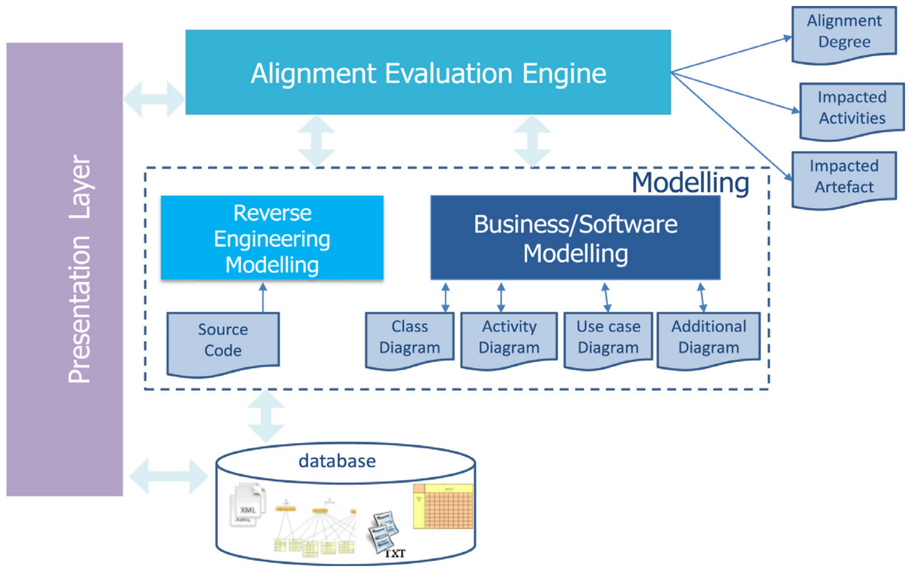



**Fig. 4.** Alignment management supporting tool.

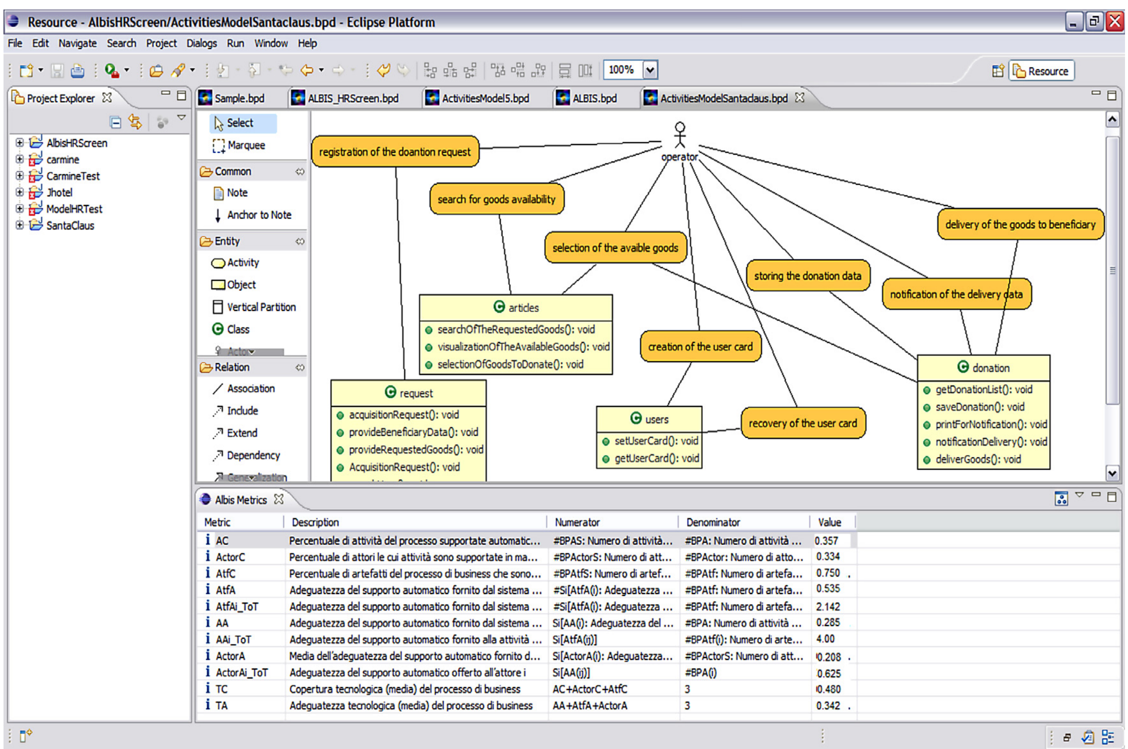



**Fig. 5.** View of calculated metrics.

Reverse Engineering Modeling. The first module uses the business documentation and process owner's knowledge and supports the definition of the business model. It permits the definition of both standard UML diagrams and new diagrams, such as those presented in Section 3.1. In particular, Fig. 5 shows one diagram represented by using the proposed notation. The figure shows a screenshot containing a diagram illustrating the actors involved in the process activities, and the set of operations performed by each activity on the business artifacts. Moreover, the diagram discussed in Section 3.1 synthesizing the interaction of each activity with the artifacts, is automatically elaborated. The reverse engineering modeling can be used when the UML models for the software systems are not provided. It analyses the source code and performs reverse engineering for obtaining the UML model [21,47]. When this operation is not feasible, the tool supports the user to model the software system.

The **Alignment Evaluation** component assesses the degree of alignment. It considers the links connecting business activities and software components stored in the database and calculates the Technological Coverage and Technological Adequacy. The outcome of this component is the synthetic evaluation of the two parameters and the analytical measurement of the lower level metrics, regarding business actors, activities, artifacts and goals. Details are shown at the bottom of Fig. 5 in a view of Eclipse. In particular, the name of each metric is displayed with its brief description, numerator and denominator of the mathematical formula and its synthetic value.

Once this mapping has been identified and the evaluation of the metrics has been performed, evolution activities must be identified and executed on the software systems and the business process itself for increasing the alignment level and improving the business process execution. This step cannot be performed without the experience and knowledge of the process stakeholders and software engineers.

## 5. Case study

In order to understand the applicability of the proposed approach in an operative context, it was applied to a case study. This required the planning of the needed set of activities, described in the following subsection, while the subsequent one discusses the execution of the planned activities.

### 5.1. Case study planning

The case study execution was structured in the following three main phases: preliminary analysis, identification of the case study context, case study execution. Their description follows.

#### 5.1.1. Preliminary analysis

It required the identification of the participating actors, methods and tools to be used, and training activities to be performed.

The participating actors was composed of a team of three researchers with adequate knowhow regarding modeling techniques and metrics. They followed a training of two-week regarding the application of the proposed approach and modeling techniques, and the use of the software tool.

#### 5.1.2. Identification of the case study context

This phase required the analysis of possible operative contexts to be considered and, within them, the identification of the specific business processes to be analyzed. The context was chosen on the basis of the possibility of modeling and changing a business process within the executive environment. Two main criteria were used for choosing the business process: the existence of software systems partially supporting it; and the availability of the business process stakeholders to provide quantitative information useful for evaluating the alignment degree.

This phase required the cooperation of the researchers identified in the previous phase and two process stakeholders identified on the basis of their knowledge concerning the chosen business process.

#### 5.1.3. Case study execution

This phase was performed by a full team composed of five participants: three researchers and two process stakeholders.

The planned activities were the following:

- Business process and software components modeling
- Evaluation of the alignment
- Analysis of the alignment results
- Identification and execution of evolution actions
- Evaluation of the alignment after the evolution action execution

These activities follow the structure of the proposed approach. The results obtained by their application are reported in the next subsections.

### 5.2. Case study execution

The execution of the case study entails the plan described above. Before applying the approach, the case study context and the business process to be analyzed were identified. A number of operative context were considered and the voluntary charity association Beneslan (http://beneslan.it/) was chosen because it was more flexible and welcomes changes, being a non-profit association. In addition, Beneslan was well motivated to improve the execution of its processes. The considered business process within Beneslan regarded the management of object donations for needy children. The considered business process was supported by a software system, named SantaClaus. Obviously, the experience can be easily extended to the case in which a business process is supported by more than one system, as, both alignment definition and evaluation is performed at the software component level, regardless the software system the components belong to.

A description of the execution of the planned activities follows.

#### 5.2.1. Business process and software components modeling

The business process was analyzed with the help of the business process stakeholders. The modeling activity permitted to identify all the business activities. In addition, the analysis of each activity permitted to classify its actors. The analysis of the SantaClaus software system and available documentation allowed the extraction of a similar set of information. Fig. 6 depicts a part of the considered donation business process, which is the main process executed by the organization.

Figs. 7 and 8 show the other diagrams drawn by using the proposed notation. They describe the interaction among the business activities of the activity diagram and business artifacts. The actors involved in the process execution were the *Beneficiary, Operator* and *Administrator* of the voluntary association. The involved business artifacts were four: *Request, Articles, Donation* and *User*. Fig. 8 also shows the operations of the business activities. Fig. 9 shows the class diagram of SantaClaus.

The comparison of the models in Figs. 8 and 9 highlights that the software system did not completely support the analyzed business process, as SantaClaus did not implement some business operations. For example, it did not implement the *Request* artifact required by some business activities. This was also confirmed by the results obtained by performing the semantic analysis tracing the links between business entities and software components. A fragment of the traceability matrix is reported in Table 4.

Table 5 lists the goals of the business process represented as Boolean expressions in the notation LTL. For example, the table shows that the first goal of the business process is goal *Treating Request*. It is reached by the process execution if a donation request was made, registered and evaluated, and after that, it was rejected, or accepted or cancelled. The goal definition and activity diagram permit to identify the state transition paths. Table 6 lists the state transition paths with reference to the artifacts states. Each state transition path is defined in terms of the business activities that have to be performed for passing from an initial state of an artifact to its final state. The paths have been recovered by the activity diagram in Fig. 6. The fourth column of Table 6 lists the paths, while the subsequent column indicates which of them are technologically covered.

#### 5.2.2. Evaluation of the alignment

The modeling activity was followed by the alignment evaluation. The obtained data are listed in Tables 7 and 8.

The second column of Table 7 lists the analytical data measured for each activity, artifact, actor and goal. Table 8 includes just a summary of the evaluation. The tables show that TC is evaluated at 0.407 and TA reaches a value of 0.349. These data were obtained with the support of the software tool.

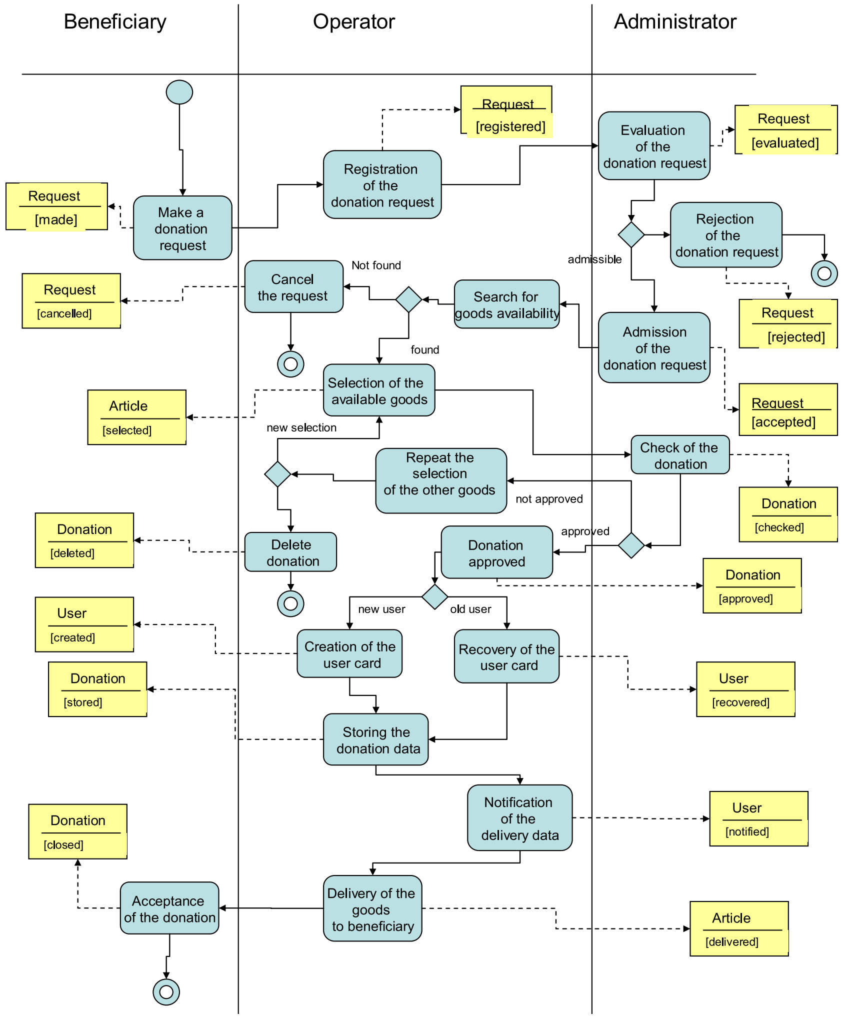



**Fig. 6.** Model of the business process that uses SantaClaus.

**Table 4**
Traceability matrix obtained by semantic analysis.

| Classes | article | | | | | | | | | | | | province | comune | | donation | account | user | | | | | | category | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Business Activities** / Methods | index | indexUser | search | searchUser | post_articoloUser | save_posted | save_Edit | edit | remove | remove_user | assignation | saveAssignation | getProvincesList | getListByProvince | returnListByProvince | getDonationsList | .... | index | getUserByAccount | addUser | getAllUserList | getUserById | checkAccount | getRootCategoriesList | getCategoriesListByRoots | getAllCategoriesList | | getCategoriesList2 | getCategoriesList3 | search | index | edit | remove |
| Make a donation request | | # | | x | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search for goods availability | # | x | # | # | | | x | # | | | # | | | | | # | | | | | | | | x | x | x | | x | x | # | | # | |
| Selection of the available goods | | | | | | | # | # | | | x | x | | | | # | | | | | | | | x | x | x | | x | x | # | | # | |
| Recovery of the user card | | | | | | | | | | | # | x | x | x | x | | | | # | x | # | # | | | | | | | | | | | |
| Creation of the user card | | | | | | | | | | | | | # | # | # | | | | | # | | | | | | | | | | | | | |
| Storing the donation data | | | | | | | | | | | x | # | | | | | | | | | | | | | | | | | | | | | |

x Not Relevant Retrieved
\# Relevant Retrieved

**Table 5**
Definition of the goal for the considered process.

| | |
|---|---|
| Treating request | ((Request.state$_t$=‘made’) ∧ (Request.state$_{t+1}$=‘registered’) ∧ (Request.state$_{t+2}$=‘evaluated’)) ∧ ((Request.state$_{t+3}$=‘Rejected’) ∨ (Request.state $_{t+3}$=‘accepted’) ∨ (Request.state $_{t+3}$=‘canceled’)) |
| Treating donation | (Request.state $_t$=‘accepted’) ∧ ((Article.state $_{t+1}$ =‘selected’) ∧ (Donation.state $_{t+1}$=‘checked’) ∧ ((Donation.state $_{t+2}$=‘approved’) ∧ (Article.state $_{t+3}$=‘delivered’) ∧ (Donation.state $_{t+4}$=‘closed’)) ∨ (Donation.state $_{t+1}$=‘deleted’)) |
| Managing user | ((User.state$_t$=‘created’) ∨ (User.state$_t$=‘recovered’)) ∧ (User.state$_{t+1}$=‘notified’) |

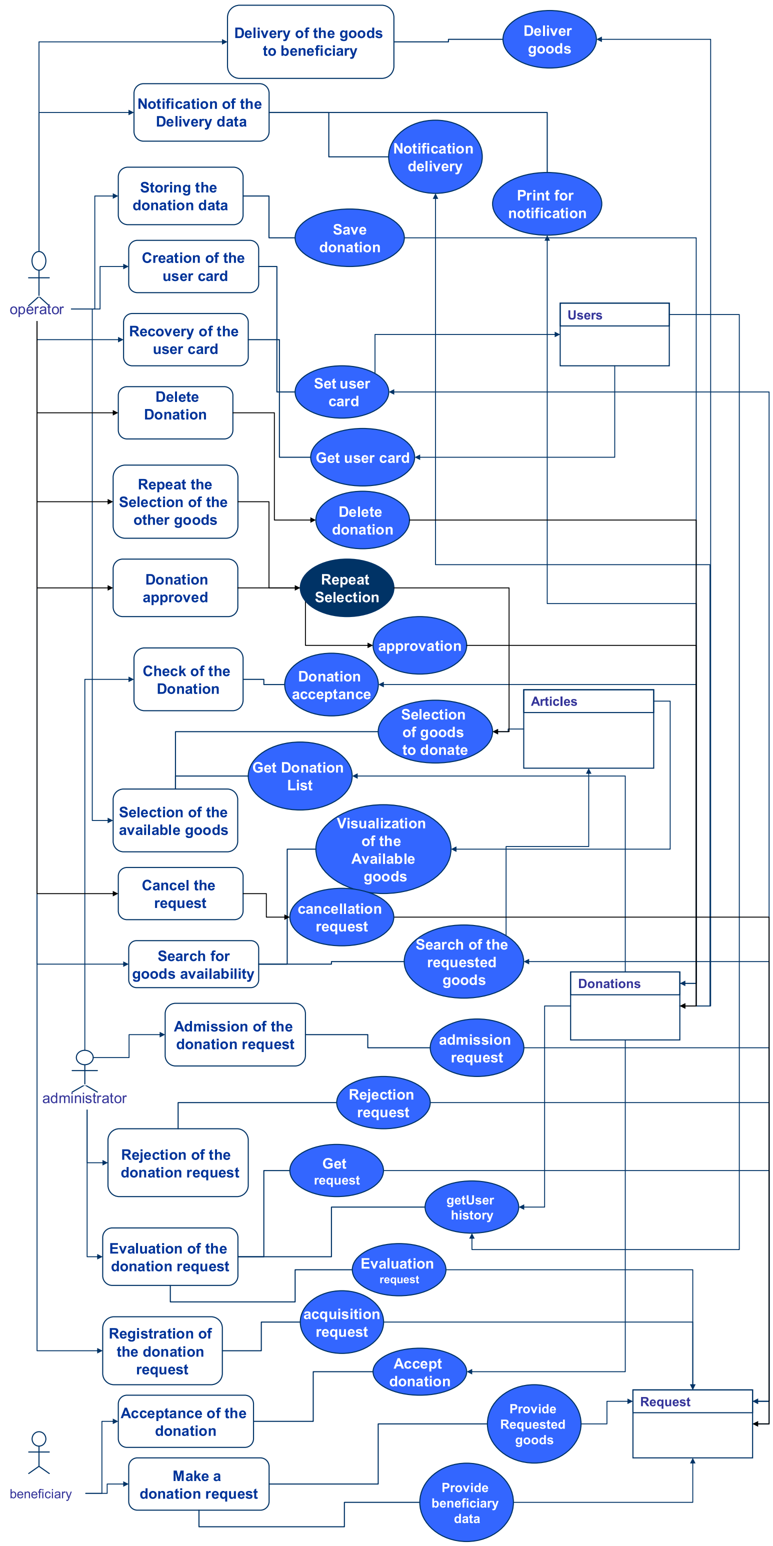



**Fig. 7.** Connection model of the donation business process.

### *5.2.3. Analysis of the alignment results*

The values included in Tables 7 and 8 point out that the support of the SantaClaus software system did not reach a good level of coverage and adequacy. In particular, the values of the metrics indicate that the main lack was related to the way the activities were supported. Actually, the supported activities were just 6 on 18, as many of them were manually executed. On the other hand, the automated activities were completely supported, as their technological adequacy achieved the maximum value. This can be observed also in Figs. 8 and 9, where: activities *Search for goods availability* and *Selection of the available goods* interact with the *Articles* artifact through operations implemented by the *Article* software class, as indicated in Fig. 9; and business activities *Recovery of the user card* and *Creation of the user card* access the *User*

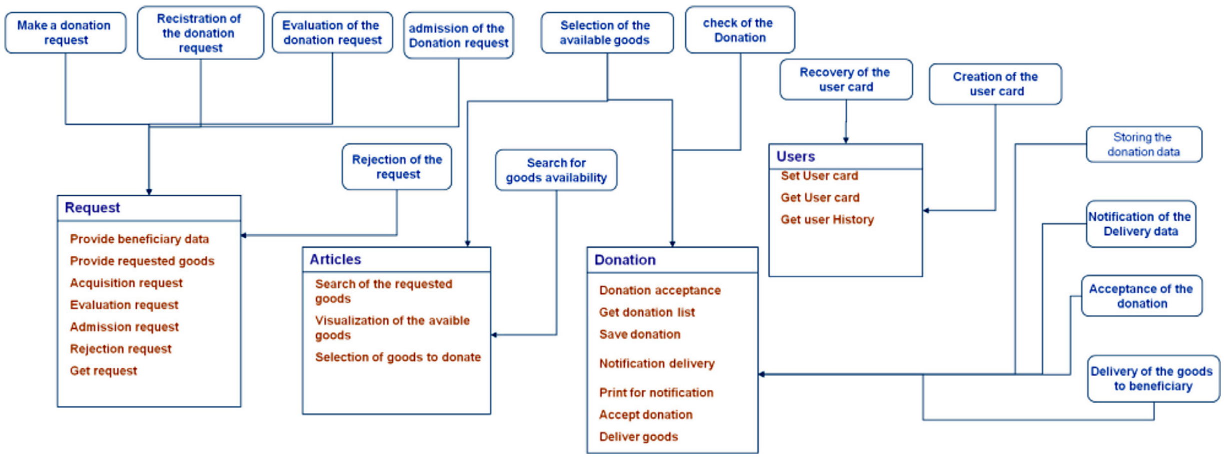



**Fig. 8.** Synthetic model of the donation business process.

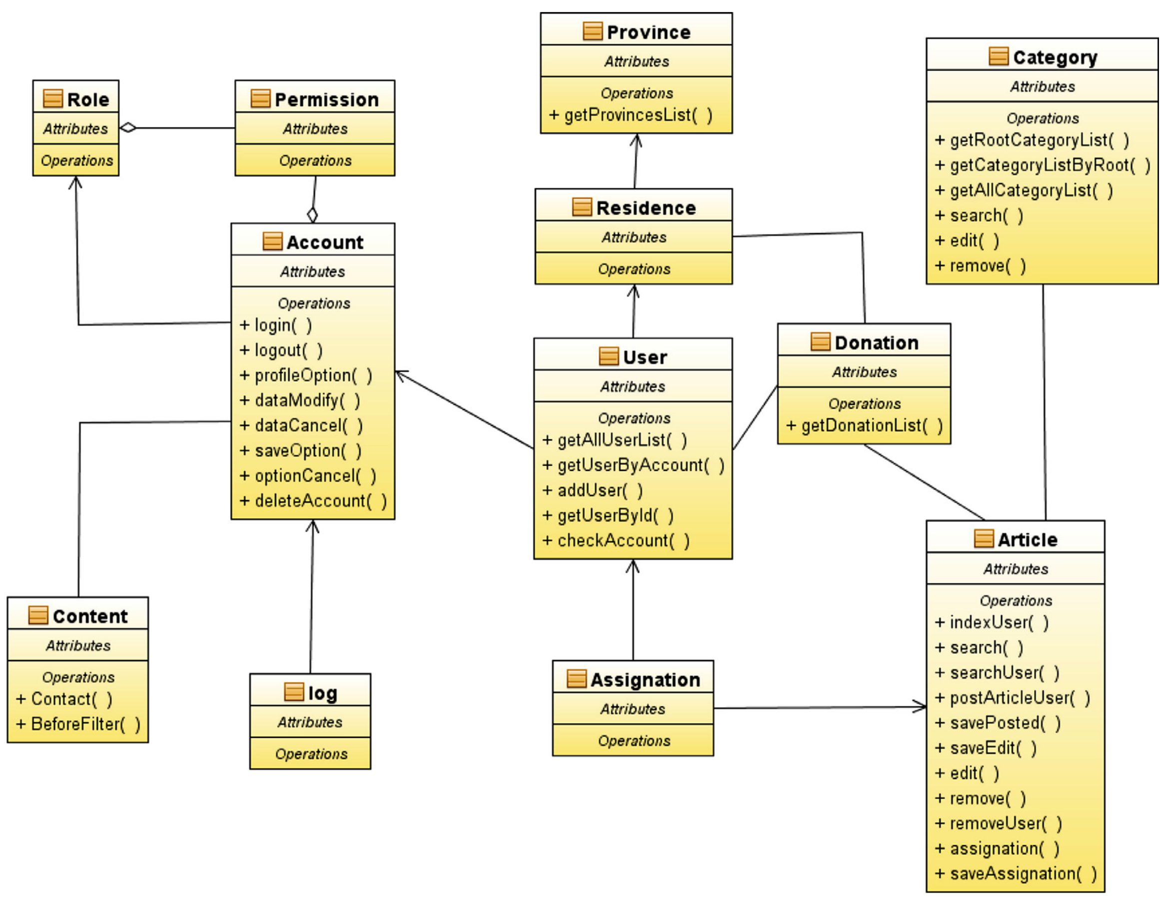



**Fig. 9.** SantaClaus software system class diagram.

business artifact by executing operations of the *User* software class of Fig. 9.

With reference to the goals, the technological adequacy of Goal *Managing user* is high, because all states its definition includes and considered in the business process are covered by the software system. Unfortunately, this goal is not considered relevant by the stakeholders. On the other side, the other two goals, *Treating request* and *Treating donation*, are considered relevant, but their coverage values are equal to 0 as showed in Table 7.

Table 7 highlights that the *Request* artifact was used by the business activities, but Fig. 9 shows that it was not implemented by the software system. This negatively impacted on the support offered to the business activities performed by actors *Beneficiary* and *Administrator*, implying the null value of their technological adequacy. Then, the overall activity adequacy value is 0.334 in Table 8. The table indicates that the highest adequacy value was 0.535, and it was reached by *ArtifactAdequacy*. Table 7 also indicates that the *Donation* artifact was considered by the business process, but some operations needed for automatically managing it were not implemented in the corresponding software class. Then, the *Article* business artifact was adequately supported, but the positive values of its technological adequacy and coverage was negatively affected by those ones of the *Donation* and *Request* artifacts.

**Table 6**
State transition coverage before/after change.

| | Start state | Final state | Path | Coverage before change | Coverage after change |
|---|---|---|---|---|---|
| 1 | Request.made | Request.registered | <Make a donation request, Registration of the donation request> | No | Yes |
| 2 | | Request.evaluated | <Make a donation request, Registration of the donation request, Evaluation of the donation request, > | No | Yes |
| 3 | | Request.rejection | <Make a donation request, Registration of the donation request, Evaluation of the donation request, Rejection of the donation request> | No | Yes |
| 4 | | Request.accepted | <Make a donation request, Registration of the donation request, Evaluation of the donation request, Rejection of the donation request, Admission of the donation request > | No | Yes |
| 5 | Request.registered | Request.evaluated | <Registration of the donation request, Evaluation of the donation request> | No | Yes |
| 6 | | Request.rejected | <Registration of the donation request Evaluation of the donation request, Rejection of the donation request> | No | Yes |
| 7 | | Request.accepted | <Registration of the donation request, Evaluation of the donation request, Rejection of the donation request, Admission of the donation request > | No | Yes |
| 8 | Request.evaluated | Request.rejected | <Evaluation of the donation request, Rejection of the donation request> | No | Yes |
| 9 | | Request.accepted | <Evaluation of the donation request, Rejection of the donation request, Admission of the donation request > | No | Yes |
| 10 | Request.rejected | Request.accepted | < Rejection of the donation request, Admission of the donation request > | No | No |

**Table 7**
Detailed measures for evaluating TA of SantaClaus.

| Metrics | Values before changes | Values after changes |
|---|---|---|
| Actors Adequacy: ActorAdequacy$_i$ | | |
| ActorA$_1$: *Operator* | 0.625 | 0.750 |
| ActorA$_2$: *Beneficiary* | 0.000 | 0.500 |
| ActorA$_3$: *Administrator* | 0.000 | 0.750 |
| Sum: | 0.625 | 2.000 |
| Artifact Adequacy: ArtifactAdequacy | | |
| AtfA$_1$: *Articles* | 1.000 | 1.000 |
| AtfA$_2$: *Donation* | 0.142 | 0.142 |
| Atfa$_3$: *User* | 1.000 | 1.000 |
| AtfA$_4$: *Request* | 0.000 | 0.875 |
| Sum: | 2.142 | 3.017 |
| Activity Adequacy: ActivityAdequacy | | |
| AA$_1$: *Make a donation request* | 0.000 | 1.000 |
| AA$_2$: *Registration of the donation request* | 0.000 | 1.000 |
| AA$_3$: *Evaluation of the donation request* | 0.000 | 0.667 |
| AA$_4$: *Admission of the donation request* | 0.000 | 1.000 |
| AA$_5$: *Rejection of the donation request* | 0.000 | 1.000 |
| AA$_6$: *Search for goods availability* | 1.000 | 1.000 |
| AA$_7$: *Selection of the available goods* | 1.000 | 1.000 |
| AA$_8$: *Check of the donation* | 0.000 | 0.000 |
| AA$_9$: *Recovery of the user card* | 1.000 | 1.000 |
| AA$_{10}$: *Creation of the user card* | 1.000 | 1.000 |
| AA$_{11}$: *Storing the donation data* | 0.000 | 0.000 |
| AA$_{12}$: *Notification of the delivery data* | 0.000 | 0.000 |
| AA$_{13}$: *Delivery of the goods to beneficiary* | 0.000 | 0.000 |
| AA$_{14}$: *Acceptance of the donation* | 0.000 | 0.000 |
| AA$_{15}$: *Cancel the Request* | 0.000 | 0.000 |
| AA$_{16}$: *Donation approved* | 1.000 | 1.000 |
| AA$_{17}$: *Repeat the selection of the other goods* | 1.000 | 1.000 |
| AA$_{18}$: *Delete donation* | 0.000 | 0.000 |
| Sum | 6.000 | 10.667 |
| Goal Adequacy: GoalAdequacy | | |
| GoalA$_1$: *Treating request* | 0.000 | 0.778 |
| GoalA$_2$: *Treating donation* | 0.286 | 0.428 |
| GoalA$_3$: *User Management* | 0.670 | 0.670 |
| Sum | 0.956 | 1.876 |

The actor technological adequacy, *ActorAdequacy*, reached the lowest value, highlighting a bad support provided by the software system to the actors of the activities. In addition, the actor technological coverage, *ActorCoverage*, confirms this result. Actually, just one actor of the three ones involved in the business process execution was automatically supported.

**Table 8**
TA and TC values obtained for SantaClaus.

| Metric name | Value before changes | Value after changes |
|---|---|---|
| *Actor Coverage (ActorCoverage)* | 0.334 | 1.000 |
| *Artifacts Coverage (ArtifactCoverage)* | 0.750 | 1.000 |
| *Activity Coverage (ActivityCoverage)* | 0.334 | 0.714 |
| *Goal Coverage(GoalCoverage)* | 0.334 | 0.667 |
| *Path Coverage (PathCoverage)* | 0.285 | 0.714 |
| Technological Coverage | 0.407 | 0.819 |
| *Actor Adequacy (ActorAdequacy)* | 0.208 | 0.667 |
| *Artifacts Adequacy (ArrtefactAdequacy)* | 0.535 | 0.754 |
| *Activity Adequacy (ActivityAdequeacy)* | 0.334 | 0.619 |
| *Goal Adequacy (GoalAdequacy)* | 0.318 | 0.625 |
| Technological Adequacy | 0.349 | 0.667 |

Obviously, the final evaluation regarding the alignment degree between the considered business process and supporting software system depends on the threshold, *Thre_TC* and *Thre_TA*, codifying the expected alignment level in the specific context of the executing organization.

### *5.2.3. Identification and execution of evolution actions*

The observed misalignment between the business process and supporting software system required the identification of evolution actions to be performed for increasing the alignment level. In particular, the values in the second column of Table 7 and the models of Figs. 7, 8 and 9, suggested the following evolution changes:

1. Introduction of an automatic support to the *Beneficiary*'s activities. This need emerged from the low value of the Actor Adequacy, shown in the second column of Table 8, and the null value of *ActorA$_2$* shown in the second column of the Table 7, concerning the *Beneficiary* actor. The required change was the automation of activity *Make a donation request*, for receiving the donation requests from the beneficiary user. This implied: the implementation of the new *Request* class including methods *ProvideRequestedGoods* and *ProvideBeneficiaryData*; and the

complete automation of activity *Registration* of the donation request.

2. Introduction of an automatic support to the activities executed by the *Administrator* actor. In particular, the second column of Table 7 indicates the null value of the activity adequacy of the following activities: *Evaluation of the donation request, Admission of the donation request* and *Rejection of the donation request*. The execution of this change required the implementation of new methods in the *Request* class, such as methods *GetRequest, GetUserHistory, AdmitRequest* and *RejectRequest*.

Before executing the above evolution changes they were discussed with the stakeholders for understanding their impact on the association and the possibility to perform them.

#### 5.2.4. Evaluation of the alignment after the evolution action execution

After the execution of the actions above the business process was analyzed again and the alignment level re-evaluated.

The new values reached by the Technological Coverage and Technological Adequacy, indicated in the third column of Table 8, show that the evolution brought to an increasing of the alignment level. In particular, a good improvement can be observed not only in the technological coverage and adequacy values, 0.819 and 0.667, respectively, but also in each coverage and adequacy value included in the third column of Table 7, showing the analytical evaluations obtained after the change execution. The table shows that all the three business actors involved in the process are supported by the evolved software system, implying the increasing of the technological coverage and adequacy with reference to the actors. Moreover, the implementation of a support for the activities concerning the Administrator and Beneficiary actors, brought to the increasing of the characteristics with reference to the activities. In particular, the evolved software system supports 11 of the 18 activities, against 6 of 18 activities supported by the previous version. Moreover, the number of artifacts implemented in the evolved software system increased, improving the values of the artifact technological coverage and adequacy. The change execution also permitted to increase the path transition coverage, as shown in the last column of Table 6, and this implied also the increasing of the Goal Coverage.

Finally, the proposed evaluation was useful to understand if the system architecture was well-designed. Actually, the analysis permitted to highlight that some methods were implemented in wrong classes. For example methods *saveDonation, acceptDonation, deleteDonation*, were implemented in class *Article* while they concerned the *Donation* artifact. This suggested a redesign of the software system for including the right methods in the right classes.

In conclusion, the application of the proposed approach to the discussed case study showed that just with few changes the alignment level considerably increased.

## 6. Conclusions

This paper deals with the management of the alignment level existing between a business process and the supporting software systems. This issue was widely recognized as relevant for the business process performance. In the paper, the attention has been focused on the functional alignment between business processes and software systems. Thus, it proposes an approach including modeling, evaluation and evolution activities helping in detecting misalignment, if it exists, and supporting the identification of the software changes to be performed for increasing the alignment level.

The modeling activity uses a proposed formalism, based on the UML constructs. The evaluation activities are supported by a set of metrics, helping to assess the alignment degree, through the evaluation of Technological Coverage and Technological Adequacy of the considered business process. This entails a semantic analysis to be performed for mapping business process assets and supporting software components. The approach application involves the analysis of all the goals, activities, artifacts, actors and operations of the business process, and of the software components. The dynamic behavior with reference to the business goal is also considered.

The results of the evaluation of the metrics provide a measure of the alignment degree and help to identify possible misalignments. If this happens, the alignment measures, together with the comparison of the business and software models and related links between business and software entities, can be used for indicating which software activities, classes or methods are missing for achieving the target alignment. Thus, evolution activities can be easily planned and executed for improving the alignment level and guaranteeing the most efficient execution of the business process.

A case study was presented for discussing the applicability of the approach. The people involved confirmed that the changes introduced in the business process execution and software system were appropriate, because they provided a driver for the execution of the evolution activities and facilitated the production of the necessary documents.

Lessons learned from the performed case study are as follows.

- The approach contributes in all the phases of the process and software system evolution. In any case, the approach can be improved for identifying the impact of the changes.
- The active role of the process stakeholders is very important for identifying the most relevant changes and improving the business process and its alignment degree with the software system.
- The proposed approach facilitated the understanding of business processes, software systems and related models. This favors the interaction of the software and business analysts, as it is possible to better formulate the conducted interviews with reference to the objectives and, thus, the collection of data.
- The desired level of alignment, codified by the threshold *Thre_TA* and *Thre_TC*, is essential for improving and simplifying the management of changes to be introduced. In the future, it could be useful the management of a change plan directly connected to the gap existing between the actual alignment degree and the desired value.
- The support of a software environment, such as the one described in Section 4, permits to manage a big quantity of data involved in the evolution process. This allows reducing the number of meetings and contributes to achieving a positive feeling with the process owners.

Obviously, the evaluation of the adequacy of the way a software system supports a business process could depend also on the way the actors execute it. Therefore, the adequacy measurement should also consider the actors' opinion.

The future work to be performed will regard the completion of the experimental activities aiming at understanding the approach applicability and refining the set of the chosen metrics and mechanisms for their computation. In addition, further study will also regard the development of an automatic detection tool helping to evaluate the impact of an evolution strategy. The aim is to offer a decision support when a business or software concept needs to be evolved in an operative context. Finally, an investigation will be performed for analyzing the integrability of the proposed approach with other strategies discussed in literature.

## References

[1] L. Aversano, T. Bodhuin, M. Tortorella, Assessment and impact analysis for aligning business processes and software systems, in: *Proc. of the ACM Symposium on Applied Computing* (SAC 2005), ACM Press, 2005, pp. 1338–1343.

[2] L. Aversano, C. Grasso, M. Tortorella, Measuring the alignment between business processes and software systems: a case study, in: *Proc. of the ACM Symposium on Applied Computing* (SAC 2010), ACM Press, 2010, pp. 2330–2336.

[3] L. Aversano, C. Grasso, M. Tortorella, A characterization framework for evaluating business/IT alignment strategies, in: Proc. of the 14th International Conference on Enterprise Information Systems, Institute for Systems and Technologies of Information, Control and Communication, Setubal, 2012, pp. 155–164.

[4] L. Aversano, C. Grasso, M. Tortorella, A literature review of business/IT alignment strategies, Procedia Technology, 5, Elsevier Ltd, 2012, pp. 462–474.

[5] L. Aversano, F. Marulli, M. Tortorella, Recovering traceability links between business activities and software components, in: *Proc. of Conference on ENTERprise Information Systems* (CENTERIS 2010), *Advances in Software Engineering, Communications in Computer and Information Science*, 109, Springer-Verlag, 2010, pp. 385–394.

[6] L. Aversano, M. Tortorella, Business process-aware maintenance task: a preliminary empirical study, in: *Proc. of 13th European Conference on Software Maintenance and Reengineering* (CSMR 2009), IEEE Computer Society, 2009, pp. 233–236.

[7] D. Avison, J. Jonesb, P. Powellc, D. Wilson, Using and validating the strategic alignment model, J. Strateg. Inf. Syst. 13 (2004) 223–246.

[8] Y. Baghdadi, R. Pérez-Castillo, Extracting services from legacy for service-oriented business processes: challenges and issues, in: *Proc. of 1st International Workshop in Software Evolution and Modernization* (SEM 2013), ScitePress Digital Library, 2013, pp. 3–11.

[9] Y. Baghdadi, N. Kraiem, Y. Jamoussi, R. Pérez-Castillo, Realization of agile enterprises: a meet-in-the-middle MDA approach for service-oriented business processes, in: *Proc. of 1st International Workshop in Software Evolution and Modernization* (SEM 2013), ScitePress Digital Library, 2013, pp. 103–110.

[10] V. Basili, M. Lindvall, M. Regardie, C. Seaman, J. Heidrich, J. Münch, D. Rombach, A. Trendowicz, Bridging the gap between business strategy and software development information systems strategy and governance, in: Proc. of *28th International Conference on Information Systems*, 2007.

[11] V. Basili, A. Trendowicz, M. Kowalczyk, J. Heidrich, C. Seaman, J. Münch, D. Rombach, *Aligning Organizations Through Measurement: The GQM+Strategies Approach* (The Fraunhofer IESE Series on Software and Systems Engineering), Springer, 2014.

[12] S.J. Bleistein, K. Cox, J. Verner, Validating strategic alignment of organizational IT requirements using goal modeling and problem diagrams, J. Syst. Softw. 79 (2006) 362–378.

[13] S.J. Bleistein, K. Cox, J. Verner, K.T. Phalp, B-SCP: A requirements analysis framework for validating strategic alignment of organizational IT based on strategy, context, and process, J. Inf. Softw. Technol. 48 (9) (2006) 846–868.

[14] G. Bruno, A Notation for the Task-Oriented Modeling of Business Processes, Int. J. Hum. Cap. Inf. Technol. Professionals 3 (3) (2012) 42–53.

[15] T.A. Byrd, B.R. Lewis, R.W. Bryan, The leveraging influence of strategic alignment on IT investment: an empirical examination, J. Inf. Manage. 43 (3) (2006) 308–321.

[16] R. Carvalho and P. Sousa, Business and Information Systems MisAlignment Model (BISMAM): an holistic model leveraged on misalignment and medical sciences approaches, in *Proc. of International Workshop on Business/IT ALignment and Interoperability (BUSITAL 2008)*.

[17] Y.E. Chan, B.H. Reich, IT alignment: what have we learned? J. Inf. Technol. 22 (2008) 297–315.

[18] Y.E. Chan, S.L. Huff, D.W. Barclay, D.G. Copeland, Business strategic orientation, information systems strategic orientation, and strategic alignment, Inf. Syst. Res. 8 (2) (1997) 125–150.

[19] H.M. Chen, Towards service engineering: service orientation and business-IT alignment, in: *Proc. of the 41st Hawaii International Conference on System Sciences* (HICSS 2008), IEEE Computer Society, 2008, pp. 114–123.

[20] R. Chena, C. Sunb, M.M. Helmsc, W. Jihd, Aligning information technology and business strategy with a dynamic capabilities perspective: a longitudinal study of a Taiwanese Semiconductor Company, J. Inf. Manage. 28 (2008) 366–378.

[21] E.J. Chikofsky, J.H. Cross, Reverse engineering and design recovery: a taxonomy, IEEE Software 7 (1) (1990) 13–17.

[22] C.U. Ciborra, De Profundis? Deconstructing the concept of strategic alignment, Scand. J. Inf. Syst. 9 (1) (1997) 57–82.

[23] J. Coughlan, M. Lycett, R.D. Macredie, Understanding the business–IT relationship, J. Inf. Manage. 25 (2005) 303–319.

[24] V. De Castro, E. Marcos, J.M. Vara, Applying CIM-to-PIM model transformations for the service-oriented development of information systems, J. Inf. Softw. Technol. 53 (2011) 87–105.

[25] J. De Leede, J.C. Looise, B. Alders, Innovation, improvement and operations: an exploration of the management of alignment, Int. J. Technol. Manage. 23 (4) (2002) 353–368.

[26] K. Elhari, B. Bounabat, Platform for assessing strategic alignment using enterprise architecture: application to e-government process assessment, *J. Comput. Sci. Issues*. ISSN (Online): 8 (1) (2011) 1694-0814. www.IJCSI.org.

[27] B. Elvesæter, D. Panfilenko, S. Jacobi and C. Hahn, aligning business and IT models in service-oriented architectures using BPMN and SoaML, in *Proc. of the 1st International Workshop on Model-Driven Interoperability* (MDI 2010), pp. 61–68.

[28] A. Etien, C. Rolland, Measuring the fitness relationship, J. Req. Eng. 10 (3) (2005) 184–197.

[29] D. Harman, Ranking algorithms, Information Retrieval: Data Structures and Algorithms, Prentice-hall, Englewood Cliffs, NJ, 1992, pp. 363–392.

[30] J.C. Henderson, N. Venkatraman, Strategic alignment: leveraging information technology for transforming organizations, IBM Syst. J. 38 (2/3) (1999) 472–484.

[31] V.A. Hooper, S.L. Huff, P.C. Thirkell, The impact of IS-marketing alignment on marketing performance and business performance, The DATA Base for Advances in Information Systems, 41, ACM Press, 2010, pp. 36–55.

[32] D. Karagiannis, F. Ronaghi, H.G. Fill, Business-oriented IT management: developing e-business applications with E-BPMS, in: *Proc. of 9th International Conference on Electronic Commerce* (ICEC'07), ACM Press, 2007, pp. 97–100.

[33] G.S. Kearns, A.L. Lederer, A resource-based view of strategic IT alignment: how knowledge sharing creates competitive advantage, Decis. Sci. 34 (1) (2003) 1–29.

[34] B. Korherr, B. List, Aligning business processes and software connecting the UML 2 profile for event driven process chains with use cases and components, in: *Proc. of 18th International Conference on Advanced Information Systems Engineering* (CAiSE 2006), Springer, 2006, pp. 19–22.

[35] S. Lee-Klenz, P. Sampaio, A. Trevor, Wood-Harper, Requirements elicitation framework and tool for sourcing business-IT aligned e-services, in: *Proc. of the ACM Symposium on Applied Computing* (SAC 2010), ACM Press, 2010, pp. 111–117.

[36] V. Mandic, V. Basili, L. Harjimaa, M. Oivo, J. Markkula, Utilizing GQM+strategies for business value analysis: an approach for evaluating business goals, in: Proc. of ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM 2010), ACM, New York, USA, 2010.

[37] J. Li, J. Tang, Y. Li, Q. Luo, RiMOM: a dynamic multistrategy ontology alignment framework, IEEE Trans. Knowl. Data Eng. 21 (8) (2009) 1218–1232.

[38] J. Lonchamp, Open source software development process modeling, The Kluwer International Series in Software Engineering, 10, 2005.

[39] J. Luftman, R.M. Kempaiah, An update on business-IT alignment: “a line” has been drawn, MIS Q. Exec. 6 (3) (2007).

[40] E. Maij, P.J. Toussaint, M. Kalshoven, M. Poerschke, J.H.M. Zwetsloot-Schonk, Use cases and DEMO: aligning functional features of ICT-infrastructure to business processes, Int. J. Med. Inf. 65 (2002) 179–191.

[41] J.D. Mckeen, H. Smith, Making IT Happen: Critical issues in IT management, Wiley, Chichester, Hoboken, NJ, 2003.

[42] C. Monsalve, A. Abran, A. April, Measuring software functional size from business process models, Int. J. Softw. Eng. Knowl. Eng. 21 (3) (2011) 311–338.

[43] Object Management Group (OMG). Business Process Modeling Notation (BPMN), 2009. www.bpmn.org.

[44] R. Papp, Introduction to strategic alignment, Strategic Information Technology: Opportunities for Competitive Advantage, Idea Group, Hershey, PA, 2001, pp. 1–24.

[45] D. Peak, C.S. Guynes, V. Kroon, Information technology alignment planning – a case study, J. Inf. Manage. 42 (5) (2005) 635–649.

[46] C. Pereira, P. Sousa, Getting into the misalignment between Business and Information Systems, in: Proc. of 10th European Conference On Information Technology Evaluation, 2003.

[47] V. Raja, K.J. Fernandes, Reverse Engineering – An Industrial Perspective, Springer, 2008.

[48] S. Ramel, E. Grandry, E. Dubois, Towards a design method supporting the alignment between business and software services, in: *Proc. of 33rd Annual IEEE International Computer Software and Applications Conference* (Compsac 2009), IEEE Computer Society press, 2009, pp. 349–354.

[49] J. Recker, J. Mendling, On the translation between BPMN and BPEL: conceptual mismatch between process modeling languages, in: Proc. 18th International Conference on Advanced Information Systems Engineering, 2006, pp. 521–532.

[50] B.H. Reich, Investigating the linkage between business and information technology objectives: a multiple case study in the insurance industry (Unpublished Doctor of Philosophy Thesis), University of British Columbia, 1993.

[51] B. Reich, I. Benbasat, Factors that influence the social dimension of alignment between business and information technology objectives, MIS Q. 24 (1) (2000) 81–113.

[52] C. Rolland, Fitting system functionality to business needs: alignment issues and challenges, in *Proc. of Conference on New Trends in Software Methodologies, Tools and Techniques* (SoMeT 2010), pp.137–147.

[53] T. Smaczny, Is an alignment between business and information technology the appropriate paradigm to manage IT in today's organisations? Manage. Decis. 39 (10) (2001) 797–802.

[54] Society For Information Management, IT Management Concerns Survey, What keeps CIO awake at night? (2006).

[55] Spem, *Software Process Engineering Metamodel Specification*, version 1.0, OMG Document formal/02-11-14, 2002.

[56] L. Thevenet, C. Salinesi, Aligning is to organization's strategy: The Instal Method, in: *Proc. of 19th Conference on Advanced Information System Engineering* (CAiSE 2007), Springer, 2007, pp. 279–293.

[57] L. Thevenet, C. Salinesi, A. Etien, I. Gam, M. Lasoued, Experimenting a modeling approach for designing organization's strategies in the context of strategic alignment, in: Proc. of Australian Workshop on Requirements Engineering, 2006.

[58] A. Ullah, R. Lai, A systematic review of business and information technology alignment, ACM Trans. Manage. Inf. Syst. 4 (1) (2013) 4.

[59] A. Ullah, R. Lai, Requirements engineering and business/IT alignment: lessons learned, J. Softw. 8 (1) (2013) 1–10.

[60] O. Velcu, Strategic alignment of ERP implementation stages: an empirical investigation, J. Inf. Manage. 44 (2010) 158–166.

[61] A. Wegmann, P. Balabko, G. Regev, I. Rychkova, A method and tool for business-IT alignment in enterprise architecture, in: Proc. of Conference on Advanced Information System Engineering, Springer LNCS, 2005, pp. 113–118.
[62] A. Wegmann, R. Regev, B. Loison, Business and IT Alignment with SEAM for Enterprise Architecture, in: Proc. of 11th IEEE International Enterprise Distributed Object Computing Conference, IEEE Computer Society, 2007, pp. 111–121.
[63] D. Weiß, J. Leukel, S.T. Kirn, *Proceedings of the PRIMIUM Subconference at the Multikonferenz Wirtschaftsinformatik* (MKWI 2008), CEUR Workshop Proceedings, 2008.
[64] R.J. Wieringa, H.M. Blanken, M.M. Fokkinga, P.W.P.J. Grefen, Aligning application architecture to the business context, in: Proc. of Conference on Advanced Information System Engineering, 2681, Springer. LNCS, 2003, pp. 209–225.
[65] P. Wohed, W.M.P. Van Der Aalst, M. Dumas, A.H.M. Ter Hofstede, Pattern-based analysis of BPMN – an extensive evaluation of the control-flow, the data and the resource perspectives, BPM Center Report (2005) www.BPMcenter.org.
[66] Workflow Management Coalition (WFMC), *Process Definition Interface – XML Process Definition Language* – Version 2.0., 2005 www.wfmc.org.
[67] C.N. Zarvi´, R. Wieringa, P. Van Eck, Checking the alignment of value based business models and IT functionality, in: *Proc. of the ACM Symposium on Applied Computing* (SAC 2008), ACM Press, 2008, pp. 607–613.
[68] Y. Zou, J.C. Foo, M. Hung, Recovering business processes from business applications, J. Softw. Maint. Evol. Res. Pract. 21 (2009) 315–348.