



HAL
open science

A Review of Core Operational Business-IT Alignment

Pascal Andre, Dalila Tamzalit, Ali Benjilany, Hugo Bruneliere

► **To cite this version:**

Pascal Andre, Dalila Tamzalit, Ali Benjilany, Hugo Bruneliere. A Review of Core Operational Business-IT Alignment. 31st International Conference on Information Systems Development (ISD 2023), Aug 2023, Lisbon, Portugal. 10.62036/ISD.2023.6 . hal-04140806

HAL Id: hal-04140806

<https://hal.science/hal-04140806v1>

Submitted on 26 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Review of Core Operational Business-IT Alignment

Pascal André

Nantes University - CNRS / LS2N
F-44000 Nantes, France

pascal.andre@ls2n.fr

Dalila Tamzalit

Nantes University - CNRS / LS2N
F-44000 Nantes, France

dalila.tamzalit@ls2n.fr

Ali Benjilany

Nantes University - CNRS / LS2N
F-44000 Nantes, France

ali.benjilany@etu.univ-nantes.fr

Hugo Bruneliere

IMT Atlantique - CNRS / LS2N
F-44000 Nantes, France

hugo.bruneliere@imt-atlantique.fr

Abstract

In terms of competitiveness, Business-IT Alignment (BITA) is still a crucial challenge for business leaders and CIOs, especially in the context of Digital Transformation and time-to-market challenges. *Core Operational BITA* can be seen as a projection of BITA to the Enterprise Information System perimeter, i.e., the operational alignment between the business processes and supporting IT. It is a major source of issues (e.g., strong couplings, maintenance costs, technical debt, slow adaptation). These cause a misalignment and thus contribute to the well-known Business-IT Gap. In this paper, we review the current state of this operational alignment in the context of Enterprise Architecture (EA) *i.e.* between the Business Process layer and the Application layer. Our analysis focuses on the models used at the Business Process and Application layers, the existing or potential links between these layers, and the use of these links to carry out a core operational alignment and facilitate the detection of potential divergence points. As a result, we notably outline some current limitations, such as modelling disparities, misuse of links between the two layers and an under-coverage of real alignment processes. We also discuss some lessons learned and future challenges, mainly around modelling needs and consistency management between the two considered layers.

Keywords: Information Systems - Enterprise Architecture - Core Operational Business-IT Alignment - Business Process modelling - Software Application modelling.

1. Introduction

Business-IT Alignment (BITA) [17] ensures that business orientations/capabilities and IT-based systems are consistent altogether, which is a major concern in Enterprise Information Systems (EIS). Investing on the alignment of the business and the IT is a necessity since IT is becoming the strongest asset of companies for improving responsiveness and efficiency [8, 21, 32]. However, most companies have been considering IT as a separate activity, with dedicated departments or even divisions that operate in parallel to the business teams. A major side effect is that 48% of the CIOs spend most of their time trying to align their IT strategies with the overall organisational objectives [19]. In addition, this task is even more complex for legacy systems. As a consequence, IT departments are continually chasing the business, with an average of

72% of the budget spent in software maintenance¹. BITA has a wide scope that ranges from the enterprise's business activities (strategy, organisation, social and cultural) to its operational IT activities. In the past, seminal works already laid the foundation for a general alignment model [17]. Recently, it has been stated that a correct alignment resides in a right balance of dependencies between business and IT concerns [15]. While the alignment at the strategic level has been widely addressed [14, 18], operational BITA has been less studied so far [30, 16]. Operational BITA is actually a challenging research area since the IT solutions must provide adequate answers to numerous and constantly evolving business needs [24, 1]. *Enterprise Architecture (EA)* provides a convenient frame to reason about EIS and BITA [26] by reducing the scope. Using different models and abstraction layers, EA captures the essentials of the business, IT and its evolution [22]. In this context, alignment is about connecting the layers via the concepts of their underlying models. To avoid confusion with other definitions, we define *Core operational BITA (COBITA)* as the operational integration of the alignment between the Business Process and Application layers of EA. Unlike [16], we do not consider goal or business requirement models in the business layer but we do consider information data (e.g., for security or privacy concerns) instead.

The work presented in this paper focuses on COBITA. We believe it deserves much more attention for two main reasons: i) When software applications are not or no longer in line with the business organisation, dysfunctions appear in the system and generate technical debt as well as unforeseen costs, ii) COBITA is a complex process tied to the intrinsic differences between the business and the IT. This frequently leads to misunderstanding between business and technical teams [15], leading to a Business-IT gap. To the best of our knowledge, COBITA has partially been studied in a broader context of literature reviews [1, 16, 33]. They all highlight its importance but also its complexity due to the disparity of the existing approaches in terms of objectives, terminology, models, alignment, evaluation and evolution. Because of that, the provided contributions are difficult to assess and compare regarding the various topics they cover and the different concerns they address (e.g., mapping, architecture, evolution). Moreover, none of them proposes a core and common approach to consider COBITA issues. This paper intends to be a step in this direction by proposing an overview of the current state of the core operational alignment between the Business Processes and Application layers.

The remainder of the paper is structured as follows. Section 2 presents the methodology we used for selecting, preparing and comparing the works from the literature. This methodology has been designed to tackle three main Research Questions (RQs) we identified. The three following sections address each one of these RQs, respectively. Section 3 reviews the modelling paradigms and languages used in the selected references, as well as their predominance. Section 4 analyses the categories of links considered between the business and application layers. Section 5 explores how explicit links between these two layers can be exploited to compute and conduct alignment-related activities. Then, Section 6 describes general lessons learned from our study, discusses potential threats to validity related to our work, and opens on future research perspectives. Finally, Section 7 concludes the paper.

2. Study methodology

In order to conduct a uniform analysis of research works dedicated to COBITA, we need to consider the concerned Business Process and Application layers as well as how they are modelled in the context of EA. Then, we also consider how these two layers are interrelated and how these relations are exploited in the alignment context. The research questions are:

RQ1 *How the business process and application layers are represented?* This relates to the

¹<https://www.computerworld.com/article/2486278/it-management/how-to-balance-maintenance-and-it-innovation.html>

modelling paradigms and languages used in both layers (i.e., the "B" and the "IT" of BITA). It was only superficially studied in [16]. RQ1 is addressed in Section 3.

RQ2 *How the relations between the business process and application layers are represented and computed?* This is the essence of any alignment (i.e., the "A" of BITA) but was not studied in [16]. RQ2 is addressed in Section 4.

RQ3 *How can we exploit these relations to perform alignment-related activities and what can we do with the alignment?* This is quite complex because this directly relates to the motivation of BITA (Why aligning? Which concerns to align? Etc.). It was not studied at all in [16]. RQ3 is addressed in Section 5.

This section presents the methodology we followed to select, prepare and compare scientific research contributions on COBITA². To clarify the terminology we use all along the paper, we introduce the following definitions: a **reference** is a publication (e.g., in a conference or a journal), while a **work** is a contribution of a group of authors that can refer to several **references** around the same approach. A reference's citation appears as a number within the paper while a work's citation appears with the letter G(roup) followed by a number. Our methodology followed three main stages, as follows.

Selection stage. We started from three search entries and built three data sets: (1) *Previous works and related references.* Considering around 700 entries from a previous bibliography [23] and narrative reviews coming from search engines centred on business-IT alignment and enterprise architecture, we selected only the references actually dedicated to COBITA, e.g. [31, 12, 11, 7][G02,G41,G12]. (2) *Surveys and systematic studies.* This secondary literature data set includes 56 references and comparison of BITA approaches, e.g. [1, 29, 33, 16, 13]. The closest contribution to our context is the one of Habba et al [16] that covers Operational BITA at large; Thus, it is an interesting starting point to our survey. However, it provides few details on COBITA. (3) *A systematic study of recent researches.* To complement the above data sets, we conducted a systematic mapping study on the period going from 2016 to 2022. The goal was to focus on recent advances not covered by our two previous data sets [16] [23]. We collected references by searching on IT databases (ACM, Elsevier, IEEE, Scopus, Springer) with variants (according to the database search engine rules) of the main keyword string ("*Business-IT Alignment*" OR "*Alignment*") AND "*Business Process*" AND ("*software architecture*" OR "*Software System*" OR "*IT application*") AND PUBYEAR > 2014". We also used additional keywords "Modeling" "Alignment Metrics" "Refinement" "Traceability" "Link" "Mapping". We collected 362 new references in total, and we included 17 of them at the end, because tied to COBITA.

Preparation stage. The working data set, obtained by the union of the three previously mentioned data sets, has then been refined to get a final list of works. The refinement process we applied was the following: (1) We cleaned up the references: duplicated entries were removed or renamed. (2) We performed "forward snowballing" to identify missed references and "pair snowballing" to collect other references for the same work (i.e., same group of authors around a given research approach). (3) Once purged from redundant or wrong entries (e.g. surveys, abstract visions, problem statements, etc.), we obtained 127 published references. (4) Then, we checked again the selection criteria (modelling language, inter-layer links, alignment goals, tooling, case studies, evolution) to further filter the references. This led to 88 references grouped into 44 works. (5) Finally, we selected one *representative reference* for each one of these 44 works in order to build our consolidated list of works/references. The reader will find all the details on this process, as well as the final list of works/references, in an *appendix* we made

²We do not consider industrial EA tools. See the Gartner peer review and rating at <https://www.gartner.com/reviews/market/enterprise-architecture-tools>

available online³.

Comparison stage To compare the selected works, we considered three axes: i) the models used at the two concerned layers, ii) the existing or potential links between these layers, and iii) how these links are exploited. Thus, we organise the comparison around the corresponding research questions.

3. Business Process layer and Application Layer Modelling

To answer *RQ1*, we analyse the current usages of different modeling languages to express models of the Business Process and Application layers. A summary table of this analysis is available in Section C1 of the provided appendix³. In what follows, we discuss languages to model the two layers. It can be domain specific languages *e.g.* Tartarus [G05], system modelling with standards *e.g.* UML, or general purpose languages (*e.g.* ontology-based models [G16,G31,G34,G46]).

Business Process (BP) Layer. Recall that we focus on business processes (no goal or requirements are considered), the use of BPMN is predominant *i.e.* 16 works out of 44, while surprisingly UML is referred in 4 works only. Even if BPMN is a standard language, it is used differently in the 16 identified works. Sometimes, BPMN is coupled with another language to provide additional information or to address a specific need. For example, [G12] exploit the service-oriented development method (SOD-M) to model business and information views. This work also uses the e3value modelling method to include economic data. Another common way to model the BP layer is to use Domain Specific Languages (DSLs): 9 works out of 44. These DSLs are often based on different customised metamodels. For example, [6] leans on the Tartarus Metamodel inspired from Model Driven Architecture (MDA), while [27] leans on a component-based representation to realise Software Derivation from Business Components (SDBC). The other works (19 out of 44) exploit various modelling languages such as Map Ontology [G16], BSCG [G18], SMC [G22], BPEL [G29] or BPOSA [G45].

To summarize, BPMN appears to be a good choice to model the Business Process layer because of its wide adoption and its status as a standard. However, it is often not sufficient to cover all the modelling needs of this layer. Thus, it can be required to connect BPMN models to other approaches for specific concerns (requirements, value creation, etc.) and to DSLs for more specialised models (data, security, privacy).

Application Layer. In this layer, also called System model in [17], the disparity is more remarkable than for the BP layer: 12 standard languages are used (ArchiMate, BPEL, BPMN, BPOSA, ISO/IEC 42010, OWL, SCA, SMC, SOA, SoaML, UML, WSDL). UML is the most frequently encountered one (14 works out of 44) [G02,G03,G09,G10,G12,G16,G19,G26,G27,G28,G32,G33,G40,G47]. Class and component diagrams are the most used structural diagrams. Use case and activity diagrams are the most used behavioural diagrams, even if they are not only intended to the application layer. DSLs are also used in 7 works to address specific needs at the application layer [G05,G23,G30,G34,G36,G42,G46]. For example, the SDBC model enables to specify the different application elements from previously identified (generic) business components [G36]. Moreover, several architectural styles can be considered [3]: Layered, Service-Oriented Architecture, Model-View-Controller, Client-Server patterns, publish-subscribe (ERPs).

To summarize, the used modelling languages and practices are quite heterogeneous at the Application layer. We identified 12 different ones, without counting various architectural styles and in-house models of legacy systems. This is probably a main reason explaining the lack of reusable COBITA approaches. Indeed, dealing with the alignment between the business and the IT first implies to be able to correctly abstract the application architecture (*e.g.*, in terms of components and connectors, Service-Oriented Architectures).

³<https://tinyurl.com/4eh3wdkc>

4. Linking Business Process and Application Layers

To answer *RQ2*, we studied how the selected works specify the relations between the Business Process and Application layers. A relation is a set of *vertical links*⁴ that connect element(s) of one layer to element(s) of the other layer. For the sake of simplicity, we will use the term **link** in the remainder of the paper. We consider three features for links: their nature (implicit/explicit), their representation in case they are explicit (kind of relation, multiplicity, orientation, structure, etc.) and their semantics/meaning (refinement, mapping, traceability, etc.). In addition, we also consider the way the links are discovered (manual mapping, transformation, inference, etc.). A summary table of this analysis is available in Section C2 of the provided appendix³.

Nature of links. *Implicit* links are relations that exist between the layers but have no concrete representation. For example, all the works that transform (or generate) application elements from business elements without storing the traceability links belong to the implicit category of links. In our study, 25 works out of 44 define implicit links, and are mainly oriented towards a development perspective. They deal with the implementation of the business processes in the IT thanks to model transformation and/or code generation. For instance, business processes (Event-driven Process Chains - EPC) can be generated from use cases without specified links [G29]. *Explicit* links are specified within models in only 15 works out of 44. Only explicit links enable alignment metrics, inference or computations. For instance, explicit links can be defined between concepts of business processes and web services based on a global alignment metamodel [G34].

Representation of explicit links. We observed two kinds of representation for explicit links (i) 10 works out of 44 [G02,G06,G16,G18,G30,G31,G33,G34,G38,G43] consider links as instances of simple relations between concepts such as associations, dependencies, generalisation/specialisation or custom (e.g., MapReduce [G16]). Using associations enables to fix roles (for model navigation or querying), directions (unidirectional or bidirectional), and cardinalities. For instance, [G05,G18] uses UML associations and [G37,G43] use Archimate's ones. Associations are a powerful enough representation to evaluate the consistency of the considered alignment. Sometimes, an entity from a model is linked to many entities in the other model [G06] e.g. one business activity can be implemented by many application services, and one application service can realise many business activities (*one-to-many* links). Such links are used when refining business activities or entities into application ones. For example, [G04] proposes a typology of 5 different patterns of associations, namely "map-split-merge-remove-insert". (ii) 5 works out of 44 consider links as first-class entities whose types are specified in metamodels [G04,G05,G22,G37,G40]. The benefits of having such a metamodel are numerous: provide a richer semantic and richer set of possible queries, disconnect the links from the layers representation, enable the aggregation of data information for alignment, etc. The ISO 42010 standard for Systems and software engineering is a fair entry like in [G18]. In some cases, the link metamodel is merged with the Business Process and Application layers metamodels [G22]. In other cases, it is independent e.g. as a separate "Link Model" [G40].

Links Semantics. We identify several kinds of semantics for (explicit) links in the different works: (i) **correspondence** - This binary information simply indicates that a connection exists between concepts. For example, the Tartarus metamodel establishes unidirectional correspondences [G05]. (ii) **Traceability or refinement** - This indicates that the application layer concept(s) actually come from business concept(s). Traceability is mentioned in 3 works out of 44 [G02,G13,G18]. For instance, a traceability matrix can be used to store the links between business activities and application components [G02]. (iii) **Single versus multi-dimensions** - The dimensions target the different stakeholder's concerns, e.g. functions, data, security, privacy,

⁴*Horizontal links* connect elements of the same layer.

urbanisation areas, people. Most of the works mentioned in this survey focus on a single dimension. For example, 13 works [G02,G04,G06,G09,G17,G19,G22,G30,G32,G37,G38,G41,G44] focus on aligning functions, 9 works [G01,G28,G33,G34,G37,G39,G42,G43,G45] focus on services and 9 works focus on data [G03,G05,G17,G18,G30,G37,G38,G40,G47]. Business actors [G17,G19,G38,G43] and business events [G19,G29] have also been considered. However, dealing with multiple dimensions is important in the context of COBITA. It can be achieved in an orthogonal vision by separating the analyses, for example to work on both the structural and behavioural aspects [G26,G30,G37]. To the best of our knowledge, and despite some existing works on views proposing so-called "tracks" [G18,G23,G47], no existing works propose a concrete integration of different dimensions.

Explicit Link computation. In practice, architects maintain models and (sometimes) links for small architectures that quickly become out-of-date. A real challenge in COBITA is to discover the links (whatever representation or semantics they have) and to instantiate the links models for alignment-related purposes. However, the architects need assistance and dedicated tooling in the context of large and complex systems. (i) In generative approaches following a top-down process, **forward engineering** consists in parsing the business model to generate application models. 14 works currently use such an approach [G02,G09,G11,G12,G13,G19,G25,G28,G32,G33,G34,G42,G46,G47]. A typical scenario is the transformation of business activities into SOA services [G11,G12,G25,G28,G42,G46]. But, once again, the (traceability) links are surprisingly neither stored nor exploited for alignment purposes in these works. (ii) **reverse engineering** intends to abstract application concepts from business ones by following a bottom-up approach and 4 works [G01,G02,G34,G39] mentioned this technique. [G01] uses reverse engineering to produce a BPMN-to-BPEL transformation by synchronising BPMN with BPEL updates. [G02] analyses the source code and performs reverse engineering to discover a UML model, but this is a software model with a low-level of abstraction (classes and operations) while we need a high-level architectural model of the applications (usually expressed using components and services). In [G30], abstracting from source code to architecture models remains challenging. [G39] proposes service identification based on generated documentation of both functional and non-functional requirements, but this work focuses more on aggregating data than abstracting the software architecture in terms of services. (iii) **Mapping** consists in establishing a correspondence between concepts. We found 11 out of 44 works using different forms of mappings [G02,G06,G22,G23,G24,G28,G30,G37,G40,G43,G44]. Some (generic) mappings rely on types rather than instances, and are represented in tables or as model transformation rules. A comparison of such model mapping techniques is provided in [23]. This approach also proposes a user-driven mapping by drag-and-drop, but heuristics are necessary as soon as the layer models grow in size. (iv) **Matching** goes one step further: it looks for candidate mappings [G02,G05,G16,G31,G34,G46]. The matching technique is usually implemented in matching engines, and lean on algorithms relying on features related to the concepts which are candidates for the alignment. Similarity of names is one key feature that enables matching [G05,G34], and ontology matching is a bit more sophisticated solution [G36]. However, full automation without verification may lead to inconsistent matching results (cf. Section 5).

Note that forward and reverse engineering are interesting in our context because their goal is not only to generate models for the target layers, but also to store the used inter-layer links for further processing. As linking the Business Process and Application layers is complex, we would advocate for applying divide-and-conquer strategies (e.g., over several dimensions) and step-by-step transformations bringing the two layers progressively closer.

5. Exploring Alignment

To answer *RQ3*, we considered only the case of explicit links between the Business Process and Application layers. Indeed, all the works that define implicit links (25 out of 44 as mentioned

in Section 4) assume an alignment by construction that cannot be studied. Only 15 works do explicitly deal with alignment, the others focus only on links representation or inference. A summary table of this analysis is available in Section C3 of the provided appendix³. In what follows, we consider the alignment from four different aspects.

Consistency and completeness checking are about Verification and Validation (V&V). *Consistency* means that the semantics of the links is correct according to defined properties. *Completeness* means that each concept from one layer is linked to at least one concept from the other layer, except manual BP activities (lost) and new features (from IT technical architecture). Checking both properties may be complex especially when several viewpoints are considered such as data, functions, domains, performance or security. It is easier to detect inconsistencies and incompleteness than to prove the properties (!). In any case, the *horizontal links*⁴ may be part of the verification rules. In [G04], a mapping model describes the combination of business activities with IT services via transformation operators. There is an internal consistency check to ensure that each activity is transformed once and that the transformed services are those of the IT. In [G05], business and IT concepts populate ontologies, and ontology matching establishes a mapping that can be explored by KALKAS queries to detect data misalignments. In [G30], the mapping is fed by visual weaving, and consistency is provided by customisable OCL queries over data and alignment functions. In [G06], the used Shared Process Model is a n-ary mapping of correspondences to synchronise views of stakeholders, e.g. business and IT views. In [G34], alignment mismatch goes beyond the binary checking of presence or not. Incompatibilities can be detected based on both service signatures and semantics (e.g., synonym names matching). However, it implies a very detailed definition of business processes where atomic activities end up to be close to application functions.

Metrics and rating aim to produce an alignment rate for COBITA, e.g. *Business Processes and IT are aligned at 75%*. Measuring the alignment requires suitable metrics for characterising its quality level. This is a systemic vision of consistency checking that aggregates different hierarchical individual metrics. Indeed, consistency checking focus on individual elements while measuring alignment is an aggregation calculus that gives different weights to different kind of sub-alignments. Alignment metrics' computation and checking have been addressed in different ways [2, 9, 10, 20, 25, 34]. The framework of Aversano et al. [G02] considers two attributes (*Technological Coverage* and *Technological Adequacy*) over activity, actors, artefacts and transitions. The aggregation is performed by summing the resulting metrics. In [G47], the business activities are decomposed into atomic tasks and data, and a modularity metric enables to cluster the tasks in software packages. Two approaches use matrices for both compute rating and visual representation [G23,G27,G30]. In [G23], several alignment viewpoints are considered (B2B, B2IT,IT2IT) and each one of them supports both allocation and alignment. In particular, the BP2IT alignment model consists of IT service and I/O alignment matrices. In this case, the alignment is measured by matrix comparison. In [G30], clustering enables to group related concepts of one layer, e.g. components of one application. Sometimes, the metrics are used for other purposes than to compute the alignment quality e.g. to define similarities to align ontologies [G31] or to evaluate the impact cost [G11]. In [G16], generic metrics quantify the coherence between the business and the system which supports it. Defined on general ontologies, these metrics are independent from specific languages and are interesting as a reference. In [28], the goal is to measure the alignment of one viewpoint model according to a *reference* model. A global alignment is estimated via a weighted average of alignment estimations based on quantitative metrics (such as counting the number of links). Overall, this particular metrics paradigm is not directly applicable to COBITA and would require adaptations. Nevertheless, it is interesting because it provides relevant evaluation means.

Change impact for maintenance and evolution consists in aligning the current *as-is* models with possible *to-be* situations. From the management point of view, change impact is the most

interesting alignment objective because it contributes to evaluate the cost of strategic scenarios. Several solutions already intend to address it [G06,G11,G14,G16,G23,G34,G38,G44], mainly through change primitives. In [G06], in addition to consistency, the Shared Process Model can evolve with releases and perform change propagation. Changes are based on primitive change operations and corresponding structured patterns. In [G11], a change is composed of a sequence of evolution operations (CRUD), and metrics are computed by assigning costs to these operations. In [G16], the Alignment Correction and Evolution Method (ACEM) performs the change operations on a pivot model that reflects alignment refinements from intention to implementation. In [G38] business process re-engineering is addressed using *i** but the IT part remains implicit. Evaluating change impact by relying on code analysis techniques has also been proposed [G44]. However, this approach relies on the existence of sufficient links from class methods to business activities.

Dimension coverage exhibits the fractal nature of BITA. As mentioned in [17], alignment is a *multi-dimension* paradigm depending on the stakeholders or the alignment objectives *e.g. align business processes with people*. Despite reducing the scope, COBITA remains fractal. For example, we identify primary dimensions related to business concepts (functions, data, actors, causality, etc.) and secondary dimensions often related to quality and non-functional requirements (availability, security, privacy, etc.). None of the works in our study addresses the secondary dimensions, and most of the works address only one primary dimension. These latest works are mainly targeting functions to align activities to IT services [G01,G02,G04,G06,G09,G13,G28,G32,G33,G34,G39,G41,G42,G44,G45] or data [G03,G05,G19,G40]. Other approaches work on both data and functions [G17,G23,G30,G37,G38,G47]. The works that use implicit relations (cf. Section 4) provide a "refinement/traceability" one-dimension semantics. Of course, the more the models are semantically close, the more the alignment can be inferred (at least partially). For example, [G09] transforms business processes to UML activities. Moreover, aligning on events and ordering is only possible if there is some orchestration at the IT-level, *e.g.* web services [G01,G04,G06,G13,G34]. However, the order is not checked except when it is implicitly generated [G13]. The principle of service discovery has been introduced as a way to establish alignment with capabilities [G18,G21], while service discovery uses signature and name matching [G34]. Actor alignment is also possible [G38,G43], as well as goal modelling which is outside the scope of our study. Collaborative work provides another perspective on COBITA but is rather a viewpoint than another alignment dimension. For example, stakeholders can have different viewpoints according to their responsibilities [G06,G23]. In all cases, the coverage of multiple dimensions in COBITA remains a fundamental challenge.

6. Discussion and Roadmap

Sections 3 to 5 reported on our analysis concerning the three RQs introduced in Section 2. In what follows, we discuss more general lessons learned and some threats to validity regarding our work. Finally, we open on future research perspectives and next steps from our side.

6.1. Lessons Learned

Existing solutions are heterogeneous and not easy to deploy in practice. Our study revealed that the existing approaches (*i.e.* 44 studied works corresponding to 88 references) and their underlying techniques are hardly generalisable and reusable in different contexts. Many of the existing solutions have very different backgrounds: they are based on specific assumptions validated on one specific case study, and address various kinds of BITA-related problems. As a consequence, it appears to be difficult to reuse (parts of) these works in order to elaborate on a generic COBITA approach to be deployed in practice. Moreover, a large majority of the

studied works relies on tools that are not publicly available, if existing at all⁵. Added to the fact that, to the best of our knowledge, there are no benchmark case studies to compare the approaches existing in the literature, this makes deployability and replicability serious issues as far as COBITA is concerned.

Most of the solutions are partial and not really applied nor applicable. Another key finding of our study is that BITA, in its current research state, does not seem to be mature enough to be efficiently reused in companies. Indeed, only one of the studied approaches has been actually deployed on real legacy applications [G30]. Many approaches relied on (partial) application models that have been designed manually for the sake of a publication, or obtained automatically by transformation from business models. Moreover, the considered relation between business process and application models is often an approximated mapping between concepts that do not precisely match in terms of semantics. Thus, the problem of sufficiently feeding the application models appears to be globally under-investigated in the studied literature. While partial answers already exist in terms of (model driven) reverse engineering [5], the abstraction process from the source code to application architectures still requires human expertise (*cf.* the corresponding lessons learned hereafter).

A high level of human expertise is required and specific to a given company. Within our study, we also observed that many existing works rely on an implicit high-level expertise on all the concerned layers (*i.e.* the Business Process and Application layers in this paper) and on related company-specific model(s). Such an advanced expertise should be provided by experienced architects with a very good knowledge of the particular company's business and IT. However, this kind of human resources is rare in companies, and more particularly in SMEs that often delegate to third-party companies the development and management of their information systems. Even in larger companies having in-house IT services, architects are frequently specialised in a certain number of aspects (*e.g.* enterprise, business, application, infrastructure). Moreover, these architects are not always well-assisted in their tasks nor strongly supported by the company. This situation can be considered as an important factor limiting the wider dissemination and adoption of COBITA solutions in general. Conversely, companies consider this subject as strategic information and don't make their (best) practices publicly available.

Few existing solutions have a user-centred approach. From our study, we also found out that most solutions are not considering sufficiently the human during the alignment process. While a certain degree of automation is highly desirable (*e.g.* for efficiency or completeness reasons), it appears to be also relevant to better integrate the human in the loop. For instance, human intervention can be beneficial for verification and validation purposes. This is more particularly true for the decision-makers inside companies, *e.g.* management executive, business leaders or technical directors, that in-fine should also be key actors of the alignment decisions. This situation can be seen as a direct consequence of the lack of human expertise usually available in order to properly establish, evaluate and then exploit the alignment (*cf.* the previous lesson learned). Once again, this can be considered as an important limitation to the larger adoption of COBITA solutions in companies.

6.2. Threats to Validity

In terms of *internal validity*, our search of relevant research publications in the literature has been performed from different publication sources. Indeed, we considered a combination of the most well-known and trusted publication databases in our community (in addition to the snowballing step performed later in the selection process). To further reduce the possibility of missing relevant publications, we systematically used a complete keyword string that we com-

⁵We remind that purely commercial tools were not in the scope of this paper.

plemented with additional keywords and adapted according to the database search engines (*i.e.* using different query formats). Another internal threat concerns the possibility of finding different publications from the same work or group of authors. In such a case, we carefully studied the related publications to identify one key publication to be selected as the reference one (quite often the most complete and recent one). Moreover, to avoid misunderstanding or misclassification, each publication has been reviewed by at least two different persons. Finally, concerning reliability, the step 3 of the selection stage can be replayed if needed and its preparation stage is available with the Rayyan tool. To this end, the comparison criteria are shortly described in page 3 of the appendix³. However, further work is naturally needed in order to understand (and replay) the detailed classification.

In terms of *external validity*, we do not claim any result outside of the precise scope of our survey. For instance, we cannot consider publications that may have presented interesting work but used a very different terminology. We cannot consider neither publications that focus on other problems or challenges without explicitly referring to BITA, and more particularly to concepts related to COBITA, in their core contributions.

All these elements make us globally confident regarding the actual relevance, if not completeness, of the final list of selected works and references.

6.3. Research Perspectives and Next Steps

From our analysis and the lessons learned described in this paper, we identified a couple of key research lines we plan to tackle in the future.

Discuss the choice of concepts/languages and its impact on alignment. We reported on the way existing COBITA solutions model both the Business Process and Application layers and the alignment between these two layers. However, we believe that the choice of modelling language made at both layers can have a significant impact on the way the alignment can then be realised, automated, exploited, maintained, etc. To the best of our knowledge, such a correlation (or even possible causality) has not been studied yet and is still an interesting open area for new research.

Organize multiple dimensions across the different layers and their alignment. We studied the layers and alignment modelling in general, *i.e.* without focusing on the different possible dimensions that can be considered for each one of them. For instance, at different layers, we can imagine modelling both the structural and behavioural dimensions. In that case, such multiple dimensions will have to be coherently considered during the alignment as well. While some techniques already exists to support viewpoints on models for example [4], the relation with the performed alignment and the management of its coherence are still not supported efficiently (e.g. how to coherently align the functions, data or actors across multiples layers and dimensions). This is also an interesting open area for new research, possibly at the intersection with similar concerns in Enterprise Architecture for example.

To be able to go further, and in direct continuation with the work presented in this paper, our very next step is first to extend our survey as follows.

Consider additional axis for COBITA. We already provided a quite complete vision over the state-of-the-art regarding the modelling of the concerned layers, the representation of the inter-layer links, and the ways to exploit these links for alignment-related purposes. However, this work can still be completed by proposing other complementary axis. Concretely, with the previous two research lines in mind, we also plan to work on studying and characterising both the available case studies and existing tooling for COBITA.

Compare/evaluate the approaches for COBITA. In addition to the axis analysis, and to be able to proceed with the previous two research lines, we also plan to perform a more in-depth classification, comparison and then evaluation of the selected works and references. This will be realised based on criteria such as the type and correctness of the involved models, the relev-

ance of the established inter-layer links, the level of automation of the proposed techniques, the support for impact analysis and/or for evolution, etc.

7. Conclusion

In this paper, we provided a vision over the state-of-the-art of Core Operational BITA in the context of Enterprise Information Systems and from an Enterprise Architecture perspective. We notably studied the modelling of the concerned layers (*i.e.* the Business Process and Application layers), the representation of the inter-layer links as cornerstones of the alignment, and the ways to exploit these links for alignment-related purposes. While standard modelling languages (*e.g.* BPMN, UML, Archimate) are frequently used in different ways, they cannot cover all the modelling needs. Thus, complementary modelling means are also considered (*e.g.* DSLs). This is particularly true for Application layer modelling where different languages have been observed, supporting various architectural styles. As a result, there is currently a lack of uniformity when addressing COBITA. Another main finding is that the modelling of the links between the two layers has been largely uncovered so far. Indeed, 56% of the identified works do not explicitly materialise these links, 22% consider them as instances of simple relations, and only 11% consider them as first-class entities that conform to dedicated metamodels. Moreover, the semantics of these links is rather general (*e.g.* refinement links, correspondence links) while interesting benefits could be obtained by semantics more tied to the needs. Our overall objective was to expose these findings to the interested academics and practitioners from the domain. By doing this, we aim at stimulating the work on more efficient and sustainable ways to achieve and maintain COBITA [33].

References

1. Aversano, L., Grasso, C., Tortorella, M.: A literature review of business/it alignment strategies. *Procedia Technology* 5, 462–474 (2012)
2. Aversano, L., Grasso, C., Tortorella, M.: Managing the alignment between business processes and software systems. *Inf. Softw. Technol.* 72, 171–188 (2016)
3. Bass, L., Clements, P., Kazman, R.: *Software architecture in practice*. Addison-Wesley Professional (2003)
4. Bruneliere, H., Burger, E., Cabot, J., Wimmer, M.: A Feature-based Survey of Model View Approaches. *Software and Systems Modeling* 18(3), 1931–1952 (2019)
5. Bruneliere, H., Cabot, J., Dupé, G., Madiot, F.: MoDisco: a Model Driven Reverse Engineering Framework. *Inf. Softw. Technol.* 56(8), 1012–1032 (2014)
6. Castellanos, C., Correal, D.: A framework for alignment of data and processes architectures applied in a government institution. *J. Data Semant.* 2(2-3), 61–74 (2013)
7. Clark, T., Barn, B.S., Oussena, S.: A method for enterprise architecture alignment. In: *Proceedings of PRET*, vol. 120, pp. 48–76. Springer (2012)
8. Cusumano, M.A., Gawer, A., Yoffie, D.B.: *The business of platforms: Strategy in the age of digital competition, innovation, and power*. Harper Business (2019)
9. Dahman, K., Charoy, F., Godart, C.: Alignment and change propagation between business processes and service-oriented architectures. In: *IEEE SCC 2013*, Santa Clara, CA, USA. pp. 168–175. IEEE Computer Society (2013)
10. Etien, A., Rolland, C.: Measuring the fitness relationship. *Requir. Eng.* 10(3), 184–197 (2005)
11. Fabra, J., de Castro, V., Álvarez, P., Marcos, E.: Automatic execution of business process models: Exploiting the benefits of model-driven engineering approaches. *J. Syst. Softw.* 85(3), 607–625 (2012)
12. Fritscher, B., Pigneur, Y.: Business IT alignment from business model to enterprise

- architecture. In: CAiSE 2011 Workshops, London, UK. pp. 4–15 (2011)
13. Gellweiler, C.: IT architects and it-business alignment: a theoretical review. In: CENTERIS 2021 - ProjMAN 2021 - HCist 2021, Braga, Portugal. vol. 196, pp. 13–20. Elsevier (2021)
 14. Gerow, J.E., Grover, V., Thatcher, J., Roth, P.L.: Looking toward the future of it-business strategic alignment through the past. *MIS quarterly* 38(4), 1159–1186 (2014)
 15. Gouigoux, J.P., Tamzalit, D.: Business-it alignment anti-patterns: a thought from an empirical point of view. In: ISD 2021, Valencia, Spain (2021)
 16. Habba, M., Fredj, M., Benabdellah Chaouni, S.: Alignment between Business Requirement, Business Process, and Software System: A Systematic Literature Review. *Journal of Engineering* 2019, 6918105 (2019)
 17. Henderson, J.C., Venkatraman, H.: Strategic alignment: Leveraging information technology for transforming organizations. *IBM Systems* 38(2.3), 472–484 (1999)
 18. Hess, T., Matt, C., Benlian, A., Wiesböck, F.: Options for formulating a digital transformation strategy. In: *Strategic Info. Manag.*, pp. 151–173. Routledge (2020)
 19. Kahre, C., Hoffmann, D., Ahlemann, F.: Beyond business-it alignment-digital business strategies as a paradigmatic shift: a review and research agenda. In: HICSS 2017 (2017)
 20. Kassahun, A., Tekinerdogan, B.: Collaboration viewpoint for modeling cross-organizational business concerns. In: ICST 2016. pp. 3–21. Springer (2016)
 21. Kostic, Z.: Innovation and digital transformation as a competition catalyst. *Ekonomika* 64(1350-2019-2780), 13–24 (2018)
 22. Lankhorst, M.M.: *Enterprise Architecture at Work - Modelling, Communication and Analysis* (3. ed.). The Enterprise Engineering Series, Springer (2013)
 23. Pepin, J., André, P., Attiogbé, J.C., Breton, E.: A method for business-it alignment of legacy systems. In: Hammoudi, S., Maciaszek, L.A., Teniente, E. (eds.) ICEIS 2015, Barcelona, Spain. pp. 229–237. SciTePress (2015)
 24. Pereira, C.M., Sousa, P.: Enterprise architecture: business and it alignment. In: SAC 2005. pp. 1344–1345 (2005)
 25. Ralha, C.G., Gostinski, R.: A methodological framework for business-it alignment. In: Bartolini, C., et al. (eds.) BDIM 2008, Salvador, Brazil. pp. 1–10. IEEE (2008)
 26. Romero, D., Vernadat, F.B.: Enterprise information systems state of the art: Past, present and future trends. *Comput. Ind.* 79, 3–13 (2016)
 27. Shishkov, B.: *Designing Enterprise Information Systems - Merging Enterprise Modeling and Software Specification*. Springer (2020)
 28. Simonin, J., Nurcan, S., Gourmelen, J.: Weighted alignment measures of enterprise architecture viewpoints. In: CAiSE 2012 International Workshops, Gdańsk, Poland. vol. 112, pp. 592–599. Springer (2012)
 29. Ullah, A., Lai, R.: A systematic review of business and information technology alignment. *ACM Trans. Manage. Inf. Syst.* 4(1), 4:1–4:30 (2013)
 30. Wagner, H., Weitzel, T.: Operational IT business alignment as the missing link from IT strategy to firm success. In: AMCIS 2006, Acapulco, Mexico. p. 74. AIS (2006)
 31. Wieringa, R., Blanken, H., Fokkinga, M., Grefen, P.: Aligning application architecture to the business context. In: CAiSE'03, Klagenfurt, Austria. pp. 209–225. Springer-Verlag (2003)
 32. Yeow, A., Soh, C., Hansen, R.: Aligning with new digital strategy: A dynamic capabilities approach. *The Journal of Strategic Inf. Sys.* 27(1), 43–58 (2018)
 33. Zhang, M., Chen, H., Luo, A.: A systematic review of business-it alignment research with enterprise architecture. *IEEE Access* 6, 18933–18944 (2018)
 34. Zhao, X., Zou, Y.: A business process-driven approach for generating software modules. *Softw. Pract. Exp.* 41(10), 1049–1071 (2011)