



HAL
open science

Core Operational Business-IT Alignment: Current Trends and Open Challenges

Pascal Andre, Hugo Bruneliere, Dalila Tamzalit, Ali Benjilany

► To cite this version:

Pascal Andre, Hugo Bruneliere, Dalila Tamzalit, Ali Benjilany. Core Operational Business-IT Alignment: Current Trends and Open Challenges. Alberto Rodrigues da Silva, Miguel Mira da Silva, Jacinto Estima, Chris Barry, Michael Lang, Henry Linger, Christoph Schneider. Advances in Information Systems Development: Information Systemes Development, Organizational Aspects, and Societal Trends, Springer, 2024, Lecture Notes In Information Systems and Organisation. hal-04557812

HAL Id: hal-04557812

<https://hal.science/hal-04557812>

Submitted on 24 Apr 2024

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.



Distributed under a Creative Commons Attribution 4.0 International License

Core Operational Business-IT Alignment: Current Trends and Open Challenges

Pascal André

Nantes University - CNRS / LS2N
F-44300 Nantes, France

pascal.andre@ls2n.fr

Hugo Bruneliere

IMT Atlantique - CNRS / LS2N
F-44300 Nantes, France

hugo.bruneliere@imt-atlantique.fr

Dalila Tamzalit

Nantes University - CNRS / LS2N
F-44300 Nantes, France

dalila.tamzalit@ls2n.fr

Ali Benjilany

Nantes University - CNRS / LS2N
F-44300 Nantes, France

ali.benjilany@etu.univ-nantes.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* is 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 causes 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 five facets: modelling notations used at the Business Process and Application layers, existence of explicit and implicit links between these layers, potential of these links in terms of alignment or misalignment in the information system, tooling and support. 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 for core operational BITA.

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

This chapter is a revised and extended version of the ISD 2023 paper untitled "A Review of Core Operational Business-IT Alignment" [1].

1. Introduction

Business-IT Alignment (BITA) [19] ensures that business orientations/capabilities and IT-based systems are consistent, as 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' responsiveness and efficiency [9, 23, 36]. However, most companies still consider IT as a separate activity, with dedicated departments or divisions operating in parallel to the business teams. A side effect is that 48% of the CIOs spend most of their time trying to align their IT strategies with the organisational objectives [21], and this task is even more complex for legacy systems. As a consequence, IT departments are continually chasing the business, with more than 70% of the budget spent in software maintenance¹. BITA has a wide scope ranging from business activities (strategy, organisation, social and cultural) to operational IT activities. In the past, seminal works already laid the foundation for a general alignment model [19]. More recently, it has been stated that a correct alignment resides in a right balance of dependencies between business and IT concerns [17]. While the strategic level alignment has been widely addressed [15, 20], operational BITA has been less studied so far [34, 18]. Operational BITA is a challenging research area since IT solutions must provide adequate answers to numerous and constantly evolving business needs [27, 2]. *Enterprise Architecture (EA)* provides a convenient frame to reason about EIS and BITA [30]. Using different models and abstraction layers, EA captures the essentials of the business, IT and their evolution [24]. In this context, alignment is about connecting layers via concepts of their underlying models. In this paper, we define *Core operational BITA (COBITA)* as the operational integration of the alignment between the Business Process and Application layers of EA. The BP layer is the part of the *business layer* that focuses on processes and information data (e.g., for security or privacy concerns).

This paper focuses on COBITA as deserving 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, frequently leading to misunderstanding between business and technical teams [17] and thus to a Business-IT gap. To the best of our knowledge, COBITA has partially been studied in the broader context of literature reviews [2, 18, 37]. They all highlight its importance but also its complexity due to the disparity of the existing approaches in terms of objectives, terminology, models, alignment and evaluation. Because of that, the provided contributions are difficult to assess and compare against 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. Thus, this paper provides 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 introduces the methodology we used to perform our study. We tackled five main Research Questions (RQs) which are treated in the five following sections. Section 3 reviews the modelling languages used at the business process and application layers. Section 4 analyses the categories of links considered between these two layers. Section 5 explores how explicit links can be exploited to perform alignment-related tasks. Section 6 presents the available tooling support for assisting users. Section 7 describes the provided case studies and benchmarks for evaluation. Then, Section 8 discusses general lessons learned from our study and potential threats to validity, while Section 9 opens on future research perspectives. Finally, Section 10 concludes the paper.

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

2. Methodology of the Study

To perform a uniform analysis of research works dedicated to COBITA, we consider the concerned Business Process (BP) and Application layers and notably 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 an alignment context. Finally, we consider how the proposed solutions are implemented in practice and how they are evaluated accordingly. Thus, the research questions we intend to answer in this paper are:

- RQ1 *How the business process and application layers are represented?* This relates to the modelling paradigms and languages used in both layers (i.e., the "B" and the "IT" of BITA). RQ1 is treated 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-related activity (i.e., the "A" of BITA). RQ2 is treated 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 and directly relates to the motivation of BITA (Why aligning? Which concerns to align? Etc.). RQ3 is treated in Section 5.
- RQ4 *Is there available tooling support for COBITA?* Going from ideas to actually reusable tools is costly. Implementing POCs, techniques, plugins or integrated solutions requires a significant engineering effort. RQ4 is treated in Section 6.
- RQ5 *Are there open source case studies and benchmarks?* Any proposed approach should be evaluated by reproducible experiments, ideally on publicly shared case studies or benchmarks. RQ5 is treated in Section 7.

In what follows, we present the methodology we used to select, prepare, analyse and compare research contributions on COBITA². We first introduce definitions we will use all along the paper: a **reference** is a publication (e.g., in a journal or conference), a **work** is a contribution from a group of authors that can refer to several references around a same solution. A reference's citation is a regular numbered citation while a work's citation starts with the letter G (for Group) instead. As shown in Figure 1, our methodology has three main consecutive stages.

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 [25] as well as reviews on both business-IT alignment and EA coming from search engines, the first data set contains the references actually dedicated to COBITA, e.g. [35, 13, 12, 8][G02,G41,G12]. (2) *Surveys and systematic studies.* This second data set includes 56 references and comparison of BITA approaches, e.g. [2, 33, 37, 18, 14]. As an interesting starting point, Habba et al [18] covered Operational BITA at large even though providing few details on COBITA. (3) *A systematic study of recent researches.* To build the third data set, we conducted a systematic study on the period from 2016 to 2023. The goal was to identify recent COBITA advances not covered by the two previous data sets. We searched on publication databases (ACM, Elsevier, IEEE, Scopus, Springer) with variants of the main keyword string adapted to the database search engine: *"("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 "Modelling" "Alignment Metrics" "Refinement" "Traceability" "Link" "Mapping". We collected 362 new references and included 17 of them at the end.

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

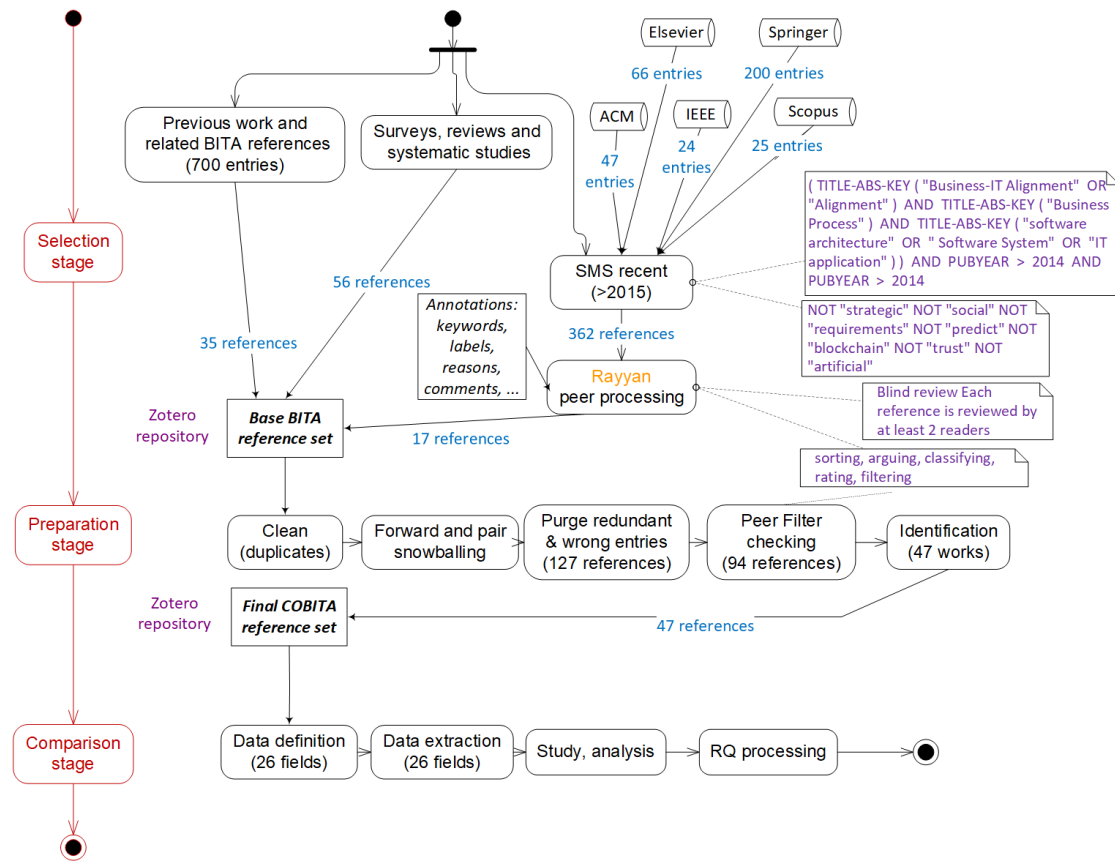


Fig. 1. Overview of the methodology used to perform our study.

Preparation stage. The working data set is the union of the three previously mentioned data sets. To obtain a final list of works from it, we applied the following refinement process: (1) We cleaned up the references by removing duplicated entries. (2) We performed "forward snowballing" to identify missed references and "pair snowballing" to collect other references for a same work (i.e., same group of authors around a same solution). (3) Once purged from redundant or inappropriate entries (e.g. surveys, vision papers, problem statements, etc.), we obtained 127 references. (4) Then, we checked again the selection criteria (modelling language, inter-layer links, alignment goals, tooling, case studies) 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. All the detailed information, including the final list of works/references, can be found in an online *appendix*³.

Comparison stage To compare the selected works, we considered five axes corresponding to our five research questions. The next sections of this paper detail the results of the performed analysis and comparison regarding these five axes/research questions.

3. Modelling the Business Process Layer and Application Layer

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 *appendix*³. In what follows, we discuss the languages currently used to model the two layers, as shown in Figure 2.

³<https://tinyurl.com/4eh3wdkc>

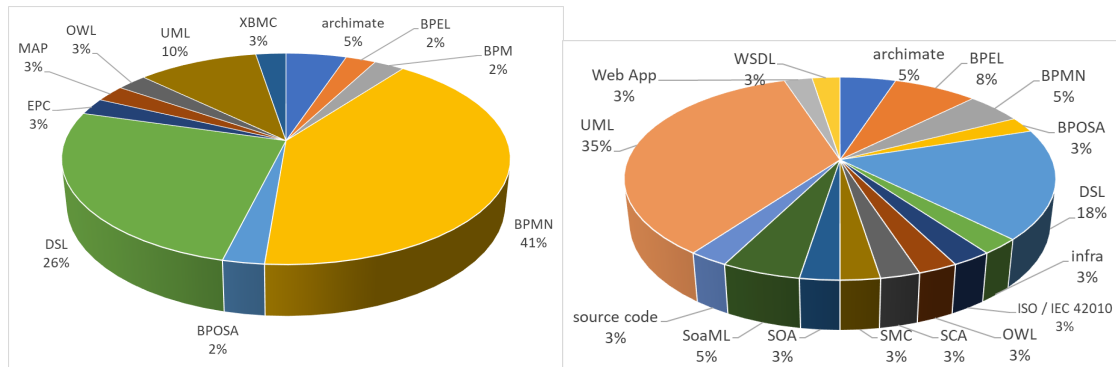


Fig. 2. Used modelling languages: Business Process layer (left) and Application layer (right).

Business Process (BP) Layer. In this layer focusing only on business processes, BPMN is widespread *i.e.* 16 works out of 44 while surprisingly, UML is referred in 4 works only. Even if BPMN is a standard modelling language, it is used differently in the 16 identified works [G01, G04, G06, G09, G10, G11, G12, G13, G19, G23, G24, G27, G28, G33, G34, G46]. 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, [7] leans on the Tartarus Metamodel inspired from Model Driven Architecture (MDA), while [31] 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 summarise, 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 [19], the disparity is more remarkable than for the BP layer: 12 standard modelling 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 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 application elements from previously identified (generic) business components [G36]. Moreover, several architectural styles can be considered [4]: Layered, Service-Oriented Architecture, Model-View-Controller, Client-Server patterns, publish-subscribe (ERPs).

To summarise, 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).

4. Linking the Business Process layer and the Application Layer

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 are considered implicit. As visible in Figure 3, we found out that a majority of the links are implicitly defined (in 29 works out of 44), 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]. On the contrary, *Explicit* links are stored information. They are specified within models in only 15 works out of 44. For instance, explicit links can be defined between concepts of business processes and web services based on a global alignment metamodel [G34].

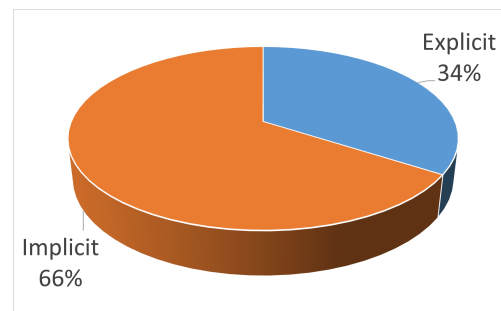


Fig. 3. Nature of the links.

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 powerful enough representations 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: it provides a richer semantic and richer set of possible queries, disconnects the links from the layers representation, enables the aggregation of data information for alignment, etc. The ISO 42010 standard for Systems and software engineering appears to be a relevant entry point [G18]. In some cases, the link metamodel is merged with the Business Process and Application layers metamodels [G22]. In others, it is independent and exists as a separate "Link Model" [G40].

Links Semantics. We identified several kinds of semantics for the links: (i) **Correspondence** - This binary relation simply indicates that a connection exists between concepts. For example, the Tartarus metamodel establishes unidirectional correspondences [G05]. (ii) **Traceability**

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

or refinement - This indicates that 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 related application components [G02]. (iii) **Single versus multi-dimensions** - Dimensions target the different stakeholder's concerns, e.g. functions, data, security, privacy, urbanisation areas, people. As visible in the left of Figure 4, most of the found works focus on a single dimension. 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 aligning services and 9 works focus on aligning data [G03,G05,G17,G18,G30,G37,G38,G40,G47]. In some works, business actors [G17,G19,G38,G43] and business events [G19,G29] have additionally 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 corresponding 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 work proposes a concrete integration of different dimensions.

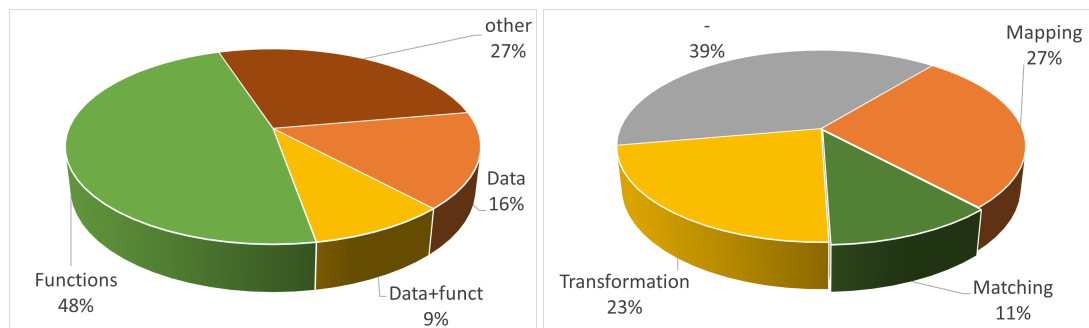


Fig. 4. Targeted dimensions (left) and Link computation techniques (right).

Explicit Link computation. In practice, architects maintain models and, sometimes, related links in the context of relatively small architectures that quickly become out-of-date. Thus, a real challenge in COBITA is to discover the links and to explicitly represent them in dedicated links models for alignment-related purposes. In the context of large and complex systems, the architects really need assistance and dedicated techniques in order to perform these tasks. However, 39% of the works do not deal with that point as showed in the right of Figure 4, for the others we have (i) **Transformations** (23%) are generative approaches. *Forward engineering* consists in using the business model to generate corresponding 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. *Reverse engineering* intends to abstract business concepts from application ones by following a bottom-up approach. 4 works [G01,G02,G34,G39] mentioned such an approach. [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. However, this is a software model with a low-level of abstraction (classes and operations) while a high-level architectural model of the applications (usually expressed using components and services) would be preferable. As discussed in [G30], abstracting from source code to architecture models remains challenging. [G39] proposes service identification based on generated documentation describing both functional and non-functional requirements. However, this work focuses more on aggregating data than abstracting the software architecture in terms of services. (ii) **Mapping** (27%) consists in establishing a correspondence between concepts. We found 11 works out of

44 that use different forms of mappings [G02,G06,G22,G23,G24,G28,G30,G37,G40,G43,G44]. Some mappings rely on types rather than instances, and are represented in tables or expressed as model transformation rules. A comparison of such model mapping techniques also proposes a user-driven mapping by drag-and-drop [25]. However, heuristics are necessary as soon as the layer models grow in size. (iii) **Matching** (11%) goes one step further as 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] while ontology matching is a bit more sophisticated solution [G36]. However, full automation without verification may lead to inconsistent results in the alignment exploration (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.

5. Exploring the Alignment between the Business Process Layer and the Application Layer

To answer *RQ3*, we considered only the case of explicit links between the Business Process and Application layers (15 works out of 44 as mentioned in Section 4) because those links are required to infer alignment information. A summary table of this analysis is available in Section C3 of the provided appendix³. The question is: what can we do with the links in terms of alignment? In what follows, we consider four alignment-related activities.

Consistency and completeness checking are Verification and Validation (V&V) tasks. *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. Note that there may be exceptions for manual BP activities where no IT is required, or for IT-specific features. Checking both consistency and completeness is complex, especially when several viewpoints are considered such as data, functions, domains, performance or security. In [G04], a mapping model describes the combination of business activities with IT services via transformation operators. There is also an internal consistency check to ensure that each activity is transformed once and that the obtained services are the IT ones. In [G05], business and IT concepts populate various ontologies, and ontology matching is used to establish a mapping that can be explored by KALKAS queries to detect data misalignments. In [G30], the provided 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], the detection of alignment mismatches goes beyond the binary checking of presence or not. Indeed, 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 are actually 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%*. To this end, suitable metrics are required for characterising the quality level of the evaluated alignment. Such an approach follows a systemic vision of consistency checking by aggregating different hierarchical individual metrics. Indeed, consistency checking focus on individual elements while measuring alignment is an aggregation calculus that gives different weights to different kinds of sub-alignments. In the literature, alignment metrics' computation and checking have been addressed in different ways [3, 10, 11, 22, 29, 38]. The framework of Aversano et al. [G02] considers two attributes (*Technological Coverage* and *Technological Adequacy*) over activities, actors, artefacts and transitions. The aggregation is performed by summing the values of the computed metrics. In [G47], the business activities are decom-

posed into atomic tasks and data, and a modularity metric enables to cluster the tasks in software packages. Other approaches use matrices for both computing alignment rates and providing visual representations [G23,G27,G30]. In [G23], several alignment viewpoints are considered (BP2BP, BP2IT, 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 techniques enable to group related concepts of one layer, *e.g.* application components in the Application layer. Sometimes, 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 costs [G11]. In [G16], generic metrics are used to 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 [32], the goal is to measure the alignment of one viewpoint model according to a *reference* model. A global alignment is estimated via the weighted average of alignment estimations based on quantitative metrics (such as counting the number of links). Overall, these metrics-based approaches are not directly applicable to COBITA and would require adaptations. Nevertheless, they are interesting because they provide relevant practical means for alignment evaluation.

Change impact for maintenance and evolution consists in aligning the current *as-is* models with possible *to-be* situations. We noted 14 works out of 44 that mention this topic, but only 8 of them actually discuss change propagation, impact or realignment from a management point of view. Change impact is the most targeted alignment aspect 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 related 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. [G24] presents an evolution methodology for services. This approach is not directly related to the Business Process layer, but can help at the Application layer.

Dimension coverage exhibits the fractal nature of BITA. Indeed, alignment is a *multi-dimension* paradigm depending on the stakeholders or alignment objectives [19], *e.g.* *align business processes with people*. Despite reducing the scope, COBITA remains fractal. It involves 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 identified works 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 one-dimension semantics for "refinement/traceability". Of course, the closer the models are semantically, the more the alignment can be inferred automatically (at least partially). For example, [G09] systematically 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 align-

ment with capabilities [G18,G21], while service discovery uses signature and name matching [G34]. In addition, actor alignment is also possible [G38,G43], as well as goal modelling which is outside the scope of our study. Collaborative work provides a complementary perspective on COBITA but is rather a viewpoint than another alignment dimension. For example, various stakeholders can have different viewpoints according to their levels of responsibility [G06,G23]. In all cases, the coverage of multiple dimensions in COBITA remains a fundamental and open challenge.

6. Assisting Users with Tooling Support for Alignment Purposes

To answer *RQ4*, we focused on implementations and tools providing support for alignment (and not only for modelling the two concerned layers). As far as we know, industrial EA tools⁵ have facilities to model links but much less to exploit these links for alignment purposes. Thus, in what follows, we consider only the tooling mentioned in the research works we identified. A summary table of this analysis is available in Section C3 of the provided appendix³.

Tooling support for alignment is mentioned in 21 out of 44 works but only a few actual tools are publicly available. One URL is provided for an available (online) tool [G03], another for a broader tool [G13], some URLs are no more available [G06], are provided as future work [G40] or mention external tools [G02,G05,G12,G46]⁶. Three works propose *proof-of-concepts* (POCs) [G04, G06,G11,G29]. Different works propose techniques or rules, but with no implementation to validate the approaches. For example, [G37] propose an extension of the Archimate metamodel but there is no related implementation, despite Archi being open-source. Three other works define transformations with ATL [G01,G03,G09,G32] or QVT [G13]. They all provide mapping for functions, except [G03] that focuses on data. Available implementations are usually based on Eclipse [G02,G05,G12,G15,G30], IBM WebSphere and Aris BA [G04], ontologies [G05,G31,G34,G46] and graph techniques [G11]. As an exception, [G13] uses its own framework Minerva. Toscana [G42] offers a technical bridge between process modelling and service modelling. However, linking and alignment are loosely detailed in a limited POC. In [G29], the prototype consists of an XSLT stylesheet associated to a GUI front-end in Java. As far as we have seen, there is no tooling support for change impact and evolution (*as-is/to-be*) despite the fact that 8 out of 44 works address this topic (cf. Section 5). For example, [G11] proposes a synchronisation framework and refers to an MDE approach [16] for potential implementation. Evolution is mentioned in the process of [G34], but no related tool is provided.

Feeding the tools for alignment is an important required feature. However, few available solutions include reverse engineering capabilities to feed the tools with (application) models. Service discovery is proposed in [G34,G39,G47] but (partially) implemented in [G34,G47] only. [G01] mentioned the reverse engineering of BPEL models from source code. However, there are no details nor tooling support provided. An abstraction process is proposed in [G30], but inferring high-level application concepts remains difficult to implement in practice. Some tools are mapping solutions that could be made generic in order to support different modelling languages [G02,G30]. To this end, import/export add-ons could help by allowing tools to accept different inputs (i.e. models) coming from both the BP and Application layers.

Reusing and/or combining tools is also important, whenever they are independent or part of a common ecosystem (e.g. Eclipse Modelling). [G02] provides an interesting tool that separates modelling from alignment and covers both metrics and semantic matching. However, it is not easily reusable as an Eclipse plugin. On the contrary, the authors of [G30] propose an Eclipse Facet Plugin to Support EA Alignment [26]. [G34] proposes to use semantic matching between

⁵Gartner Magic Quadrant for EA Tools <https://tinyurl.com/gartnerMQ-EAT>

⁶We also tried to access to these tools, without success.

business process and activities with web services, while the WordNet Semantic DataBase is used for performing such a semantic analysis embedded in the Alignment Evaluation Engine [G02]. Another example is data alignment via ATL transformation [G03] that may be associated to the functional alignment of services [G03,G09,G12]. Similarly, ontology-based solutions for data [G05] and functions [G34,G46] may be complementary.

Associated methodologies and processes are mentioned in 15 out of 44 works. For example, a general three-step process (modelling, alignment evaluation and evolution execution) is given in [G34] and could be applied to different use cases. [G12] proposes SOD-M, a service-oriented development method that enables implicit alignment by model transformation (i.e. functional alignment). Similarly, [G25] provides guidelines to implement business processes by BPEL/WSDL web-services while [G32] defines transformation rules from UML activity diagrams to UML state and class diagrams. However, these may not be completely generalisable in the context of many use cases. Based on ontology matching, the framework of [G31] comes with 7 steps from identifying business and IT processes to domain ontology's alignment by metrics. But the obtained results have been quite limited at this stage. In a similar way, the Software Derived from Business Components (SDBC) approach [G36] proposes an enterprise modelling cycle interacting with the software specification cycle. However, only general guidelines are provided without any tool support. Finally, the MigraSOA approach [G34] defines a quite complete modernisation process from legacy web applications to SOA. This process notably allows to discover the web-services to be aligned with BP models. It is important to note that, as introduced earlier, many works do not come with dedicated processes nor methodological guidelines [G10,G15,G22,G24,G29,G38,G40,G45].

Overall, we have observed that the more the models are specific, the more the available tools gain in relevance. For example, aligning business activities with services is easier than connecting a business process with software components. As a consequence, providing a generic and reusable tooling support for alignment purposes is still a largely open challenge.

7. Providing Case Studies and Benchmarks to Evaluate Alignment Solutions

To answer *RQ5*, we looked for available experimental data or benchmarks, as well as (even better) for accessible case studies of alignment. The objective is to identify material that can be reused for new experiments in other alignment contexts. A summary table of this analysis is available in Section C3 of the provided appendix³.

Reusable experimental material is unfortunately very rare. As far as we have seen in our study, and also searched further on the web, there are no completely available experimentation data, benchmarks, and case studies. In particular, we did not find any full case study description coming with corresponding BP models and Application models that could be directly reused. However, some case studies relying on open source applications provide source code that can be used to build application models, manually or (semi-)automatically. For example, we can mention IBM WebSphere Commerce (WSC) and Opentaps ERP [G47], Open For Business “Purchase Order” (BPM) adopted from (OMG, 2006b) [G33], or Project (OFBiz) [G44].

Validation with case studies is found in 32 out of 44 works, despite the lack of really available material as observed before. These case studies come from several application domains including business, ERP and e-commerce [G12,G22,G32,G33,G36,G39,G43,G44,G45,G47], industry, production and monitoring [G04,G09,G17,G23,G27,G38], construction [G15], finance and insurance [G06,G16,G24,G25,G26,G30,G43], healthcare [G40,G46] (cf. [28] for a comparison of BPM in healthcare), associations [G02], education [G03,G05,G19,G29,G34], and others [G39]. In addition, 13 works rely on toy or customised examples for evaluating their solutions. They consider systems for book loaning [G03], change process (in the automotive industry) [G04], travel agency [G12], payment management [G24], fictional education [G19,G29], “Purchase Or-

der” (adapted from the OMG’s BPMN specification) [G33], e-commerce [G32, G45], insurance claims [G25,G26], distributed Web Application for SOA Evaluation [G39]. Several case studies are also presented in chapter 7 of the book of Shishkov [G36] and discuss systems for financial products, vote, or border security.

Real case studies enable a stronger validation process that challenges the proposal with various cases and exceptions. Such cases are found in 14 works: Switcher SA, a small manufacturer and distributor of garments [G17], a Web application implementing a Conference Review System (CRS) [G34], the SantaClaus software from the voluntary charity association Beneslan [G02], the Colombian Institute for the Evaluation of Education (ICFES) [G05], the Jordan’s Cancer Care and Registration (CCR) [G46], the Penawar Healthcare Information System (HCareSys) [G40], the order management in an automobile company [G38], Global Travel International (2003) and X Company (2021) [G43], a Web System for Financial Services for Enterprises from a real big financial organisation in Brazil [G43], a food supply chain transparency system [G23], a real business case in the Architecture, Engineering and Construction (AEC) sector [G15], a system in the mobile music delivery business [G22], the control monitoring of a convention center [G27], a metrics system at Renault/DIAC for managing credits and leases [G16].

Industrial applications imply large-scale application where user-driven interactions and decision are hardly scalable: automatisation and tooling are necessary. They are also found in 5 works: three insurance companies were studied in [G30], an empirical study of a large company in the banking sector has been conducted in [G06], [G09] experimented with various real-world models from complex domains in a THALES scenario from the the MODELPLEX European project, [G47] evaluated their approach on two large-scale business systems: IBM WebSphere Commerce (WSC) server and Opentap. [G44] applied their approach on the Open For Business Project (OFBiz) open source system. Finally, 6 works reuse case studies from others [cited in their respective references], still none of them is publicly available yet: [G46] reused a case from the Jordan’s CCR, [G25] reused IBM Insurance Application Architecture⁷, [G27] reused the HVAC process of the convention center from Tranquillini et al., [G09] reused the THALES scenario from the MODELPLEX European project⁸, [G47] reused IBM WSC server and Opentap [G44] reused OFBiz, [G33] reused the OMG’s BPMN specification.

8. Discussion

Sections 3 to 7 reported on our detailed analysis concerning the five research questions introduced in Section 2. In what follows, we now discuss more general lessons learned as well as some threats to validity regarding our study.

A) Lessons Learned

Coverage of the existing solutions is limited in terms of COBITA aspects. A large majority of the identified works do not cover the 5 research questions we studied. For example, some works focus more on models for the considered layers (RQ1) or identified links (RQ2), while some others focus more on dedicated metrics and different consistency checks (RQ3). A few other works are about proposing POCs (RQ4) or experimental scenarios (RQ5). However, among the 44 works, we identified 3 works [G02,G23,G30] covering at least partially the five research questions regarding multiple dimensions. Moreover, [G05] covers the 5 research questions for the Data dimension and [G11,G12,G33] cover the 5 research questions for the Functions dimension. In addition, even if the Application layer is not deeply covered, [G06] starts from an empirical study to propose the original Shared Process Model that provides interesting insights in the five research questions as well.

⁷<http://www.ibm.com/industries/financialservices/iaa>

⁸<https://cordis.europa.eu/project/id/034081>

Existing solutions are heterogeneous and not easy to deploy in practice. Our study revealed that the currently existing solutions (*i.e.* 44 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 COBITA-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 all⁹. 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 COBITA, in its current research state, does not seem to be mature enough to be efficiently reused in companies. Indeed, only one of the studied solutions has been actually deployed on real legacy applications [G30]. [G02] presents the most complete approach from modelling to evolution execution including metrics, but the supporting tool is not available. An open-source version of it would be of great interest. 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 [6], 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, at least the automated part. 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 do not 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

⁹We remind that purely commercial tools were not in the scope of this paper.

learned). Once again, this can be considered as an important limitation to the larger adoption of COBITA solutions in companies.

B) 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 complemented with additional keywords and we 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 third data set of the selection stage can be recomputed if needed and the 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 rebuild the detailed classification and analysis regarding the five research questions.

In terms of *external validity*, we do not claim any result outside of the precise scope of our survey (*i.e.* COBITA). For instance, we cannot consider publications that may have presented interesting solutions but used a completely different terminology. We cannot consider neither publications that focus on different problems or challenges without explicitly referring to concepts directly 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 identified works and references.

9. Research Perspectives and Next Steps

From the detailed analysis of the literature, we performed regarding our five research questions, and from the general lessons learned we presented before, we identified several open challenges and corresponding research lines to be possibly explored by the community 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. We also studied how explicit links between these two layers are currently categorised and modelled. Based on this, 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. It can also have a direct impact on the way the inter-layer links are represented and further reused in the context of different alignment-related activities. 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.

Organise multiple levels and dimensions across the different layers and their alignment. As linking the Business Process and Application layers is complex, we would advocate for applying divide-and-conquer strategies step-by-step transformations bringing the two layers progressively closer through intermediate layers (small steps process). In the COBITA context, 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 [5], 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.

Improve significantly the corresponding tooling support. We observed that the number and coverage of available tools for supporting COBITA are still too limited. This concerns both the tooling support for dealing with the alignment itself and related solutions for properly feeding it with appropriate input data (*e.g.* business process and application models, traceability information). There is also no real support for the study of the alignment impact in terms of system evolution. A few tools have already been proposed but are not sufficiently implemented nor validated at this stage. As a consequence, more research is required in order to overcome this, notably by 1) reusing and combining various existing solutions or techniques and 2) better specifying the associated methodologies and processes. A longer-term objective is to make available a generic and extensible tooling support dedicated to COBITA purposes.

Provide dedicated experimental material for evaluation. When performing our study of COBITA, we noted the particularly low number of actually reusable case studies or benchmarks for evaluation purposes. Many existing works do mention case studies, customised system examples (including toy ones) or other kinds of industrial applications. However, they are generally not provided along with really available experimental material. Thus, there is room for more research on specifying and developing full case study descriptions coming with related benchmarking data (*e.g.* complete business process and application models). Such material could be produced in the context of dedicated workshops, ideally involving industrial partners from different application domains. A final goal is to make it available on public repositories or via open services for dissemination purposes.

10. Conclusion

In this paper, we proposed a detailed vision over the state-of-the-art of Core Operational BITA (or COBITA) in the context of Enterprise Information Systems and from an Enterprise Architecture (EA) 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. We also studied the corresponding tooling support as well as the provided evaluation material (case studies, benchmarks) currently made available.

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. Furthermore, there is a real lack of tools dedicated to COBITA. Only 48% of the identified works mention tooling support, and only 34% of the identified works provide associated methodologies and processes to use this tooling in practice. Among these, only a few of these works actually made a tool publicly available. The missing capabilities notably include reverse engineering solutions allowing to appropriately feed the alignment tools in terms of models, traceability data, etc. Related to that, the available evaluation material dedicated to COBITA is currently insufficient. Thus, more reusable case studies, benchmarks and realistic

alignment scenarios are strongly needed.

Our objective in this paper is to expose all these findings to the interested academics and practitioners from the domain. To give a fully complete picture of the situation, our study of the scientific literature could be compared in the future with a more detailed study of the commercial EA solutions for example². Overall, we aim at stimulating the work on more efficient and sustainable ways to achieve, maintain and evolve COBITA in the context of various and varied Enterprise Information Systems.

References

1. André, P., Tamzalit, D., Benjilany, A., and Bruneliere, H. (2023). A review of core operational business-it alignment. In *ISD 2023 Proceedings*, pages 1–12, Lisbon, Portugal. AIS eLibrary.
2. Aversano, L., Grasso, C., and Tortorella, M. (2012). A literature review of business/it alignment strategies. *Procedia Technology*, 5:462–474.
3. Aversano, L., Grasso, C., and Tortorella, M. (2016). Managing the alignment between business processes and software systems. *Information and Software Technology*, 72:171–188.
4. Bass, L., Clements, P., and Kazman, R. (2003). *Software architecture in practice*. Addison-Wesley Professional.
5. Bruneliere, H., Burger, E., Cabot, J., and Wimmer, M. (2019). A Feature-based Survey of Model View Approaches. *Software and Systems Modeling*, 18(3):1931–1952.
6. Bruneliere, H., Cabot, J., Dupé, G., and Madiot, F. (2014). MoDisco: a Model Driven Reverse Engineering Framework. *Information and Software Technology*, 56(8):1012–1032.
7. Castellanos, C. and Correal, D. (2013). A framework for alignment of data and processes architectures applied in a government institution. *Journal on Data Semantics*, 2(2-3):61–74.
8. Clark, T., Barn, B. S., and Oussena, S. (2012). A method for enterprise architecture alignment. In *Proceedings of PRET*, volume 120, pages 48–76. Springer.
9. Cusumano, M. A., Gawer, A., and Yoffie, D. B. (2019). *The business of platforms: Strategy in the age of digital competition, innovation, and power*. Harper Business.
10. Dahman, K., Charoy, F., and Godart, C. (2013). Alignment and change propagation between business processes and service-oriented architectures. In *IEEE SCC 2013, Santa Clara, CA, USA*, pages 168–175. IEEE Computer Society.
11. Etien, A. and Rolland, C. (2005). Measuring the fitness relationship. *Requirements Engineering*, 10(3):184–197.
12. Fabra, J., de Castro, V., Álvarez, P., and Marcos, E. (2012). Automatic execution of business process models: Exploiting the benefits of model-driven engineering approaches. *Journal of Systems and Software*, 85(3):607–625.
13. Fritscher, B. and Pigneur, Y. (2011). Business IT alignment from business model to enterprise architecture. In *CAiSE 2011 Workshops, London, UK*, pages 4–15.
14. Gellweiler, C. (2021). IT architects and it-business alignment: a theoretical review. In *CENTERIS 2021 - ProjMAN 2021 - HCist 2021, Braga, Portugal*, volume 196, pages 13–20. Elsevier.
15. Gerow, J. E., Grover, V., Thatcher, J., and Roth, P. L. (2014). Looking toward the future of it–business strategic alignment through the past. *MIS quarterly*, 38(4):1159–1186.
16. Giese, H. and Wagner, R. (2008). From model transformation to incremental bidirectional model synchronization. *Software & Systems Modeling*, 8(1):21–43.
17. Gouigoux, J.-P. and Tamzalit, D. (2021). Business-it alignment anti-patterns: a thought from an empirical point of view. In *ISD 2021, Valencia, Spain*.
18. Habba, M., Fredj, M., and Benabdellah Chaouni, S. (2019). Alignment between Busi-

- ness Requirement, Business Process, and Software System: A Systematic Literature Review. *Journal of Engineering*, 2019:6918105.
19. Henderson, J. C. and Venkatraman, H. (1999). Strategic alignment: Leveraging information technology for transforming organizations. *IBM Systems*, 38(2.3):472–484.
 20. Hess, T., Matt, C., Benlian, A., and Wiesböck, F. (2020). Options for formulating a digital transformation strategy. In *Strategic Information Management*, pages 151–173. Routledge.
 21. Kahre, C., Hoffmann, D., and Ahlemann, F. (2017). Beyond business-it alignment-digital business strategies as a paradigmatic shift: a review and research agenda. In *HICSS 2017*.
 22. Kassahun, A. and Tekinerdogan, B. (2016). Collaboration viewpoint for modeling cross-organizational business concerns. In *ICST 2016*, pages 3–21. Springer.
 23. Kostic, Z. (2018). Innovation and digital transformation as a competition catalyst. *Ekonomika*, 64(1350-2019-2780):13–24.
 24. Lankhorst, M. M. (2013). *Enterprise Architecture at Work - Modelling, Communication and Analysis (3. ed.)*. The Enterprise Engineering Series. Springer.
 25. Pepin, J., André, P., Attiogbé, J. C., and Breton, E. (2015). A method for business-it alignment of legacy systems. In *ICEIS 2015, Barcelona, Spain*, pages 229–237. SciTePress.
 26. Pepin, J., André, P., Attiogbé, J. C., and Breton, E. (2016). An improved model facet method to support EA alignment. *Complex Systems Informatics and Modeling Quarterly*, 9:1–27.
 27. Pereira, C. M. and Sousa, P. (2005). Enterprise architecture: business and it alignment. In *SAC 2005*, pages 1344–1345.
 28. Pufahl, L., Zerbato, F., Weber, B., and Weber, I. (2022). Bpmn in healthcare: Challenges and best practices. *Information Systems*, 107:102013.
 29. Ralha, C. G. and Gostinski, R. (2008). A methodological framework for business-it alignment. In Bartolini, C. and et al., editors, *BDIM 2008, Salvador, Brazil*, pages 1–10. IEEE.
 30. Romero, D. and Vernadat, F. B. (2016). Enterprise information systems state of the art: Past, present and future trends. *Comput. Ind.*, 79:3–13.
 31. Shishkov, B. (2020). *Designing Enterprise Information Systems - Merging Enterprise Modeling and Software Specification*. Springer.
 32. Simonin, J., Nurcan, S., and Gourmelen, J. (2012). Weighted alignment measures of enterprise architecture viewpoints. In *CAiSE 2012 International Workshops, Gdańsk, Poland*, volume 112, pages 592–599. Springer.
 33. Ullah, A. and Lai, R. (2013). A systematic review of business and information technology alignment. *ACM Transactions on Management Information Systems*, 4(1):4:1–4:30.
 34. Wagner, H. and Weitzel, T. (2006). Operational IT business alignment as the missing link from IT strategy to firm success. In *AMCIS 2006, Acapulco, Mexico*, page 74. AIS.
 35. Wieringa, R., Blanken, H., Fokkinga, M., and Grefen, P. (2003). Aligning application architecture to the business context. In *CAiSE'03, Klagenfurt, Austria*, pages 209–225. Springer-Verlag.
 36. Yeow, A., Soh, C., and Hansen, R. (2018). Aligning with new digital strategy: A dynamic capabilities approach. *The Journal of Strategic Inf. Sys.*, 27(1):43–58.
 37. Zhang, M., Chen, H., and Luo, A. (2018). A systematic review of business-it alignment research with enterprise architecture. *IEEE Access*, 6:18933–18944.
 38. Zhao, X. and Zou, Y. (2011). A business process-driven approach for generating software modules. *Software: Practice and Experience*, 41(10):1049–1071.