

Domain Model Based Design of Business Process Architectures

Fernanda Gonzalez-Lopez ^{1,*} , Guillermo Bustos ², Jorge Munoz-Gama ¹  and Marcos Sepúlveda ¹ 

¹ Department of Computer Science, Pontificia Universidad Católica de Chile, Santiago 7820436, Chile; jmun@uc.cl (J.M.-G.); marcos@ing.puc.cl (M.S.)

² School of Industrial Engineering, Pontificia Universidad Católica de Valparaíso, Valparaíso 2362807, Chile; guillermo.bustos@pucv.cl

* Correspondence: fgonlop@uc.cl

Abstract: A business process architecture (BPA) model depicts business processes in an organization and their relations. An artifact for generating BPA models is proposed as the outcome of a design science research project. The proposed artifact consists of a method (i.e., a set of concepts, a proposed notation, and a detailed procedure), which is termed the domain-based BPA (dBPA) method due to using domain models as a starting point. The dBPA method tackles issues of currently available approaches: lack of structured inputs, limited consideration of process relations types, and restricted use of industry-standard modeling languages. The paper formalizes the dBPA method and illustrates its application in the manufacturing industry. Evaluation of the dBPA method revealed that practitioners perceived it as useful to achieve its goal with the benefits of being objective and clear and allowing to create complete and understandable BPA models that enable the integration of processes and the software that automates them.

Keywords: business process architecture; process landscape; process model; domain model; modeling language; business analysis techniques and technologies



Citation: Gonzalez-Lopez, F.; Bustos, G.; Munoz-Gama, J.; Sepúlveda, M. Domain Model Based Design of Business Process Architectures. *Appl. Sci.* **2022**, *12*, 2563. <https://doi.org/10.3390/app12052563>

Academic Editors: Malgorzata Pankowska and Krzysztof Koszela

Received: 14 January 2022

Accepted: 12 February 2022

Published: 1 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Work carried out in an organization can be conceptualized as a set of business processes [1]. A business process is a set of interrelated and partially ordered decisions and tasks for providing value to an internal or external customer, cf. [2]. Interconnections between processes have become a topic of interest due to their impact on organization performance [3] and service level [4].

The present work focuses on the design of a Business Process Architecture (BPA), namely the fundamental concepts or properties exhibited by a system of business processes. The BPA is also a key component of the Enterprise Architecture [5,6], which includes other related components such as data and technology architecture. A BPA can be graphically represented by a BPA model showing a set of processes and their relations from a high level of abstraction.

A BPA model is a valuable input for managing the BPA of a given organization systemically in terms of process understanding, performance, and control [7]. The BPA model is useful for articulating BPM initiatives in a holistic fashion [8]. In fact, the BPA model should be used as the starting point for the Business Process Management (BPM) cycle [9] and, in this sense, it plays a key role in the strategic alignment cycle [10].

Attending to the relevance of BPAs, a number of BPA design methods have been proposed to guide the structured design of a BPA that is represented by a BPA model [1,11]. Despite the strengths of currently available BPA design methods, these approaches usually lack the rigor found in other BPM tools. Particularly, Gonzalez-Lopez and Bustos [11] identified the following prevalent issues among BPA design methods: (i) lack of structured business knowledge inputs, i.e., BPA design methods usually rely exclusively on unstructured inputs such as expert knowledge and a company's documents; (ii) limited consideration of

business process relations, i.e., most BPA design methods support only one or two process relations types; and (iii) restricted use of industry-standard languages, i.e., dedicated notations are typically used for BPA models instead of standard ones.

For addressing the previously described issues, the present work proposes a new method termed the domain-based Business Process Architecture (dBPA) method. The dBPA method tackles the aforementioned limitations of available BPA design methods by (i) using domain models as structured input for generating the BPA models, (ii) considering a wide range of relation types within the generated BPA models (i.e., composition, specialization, trigger, and resource flow), and (iii) using the well-known ArchiMate industry-standard notation for BPA models.

The research lead to proposing the dBPA method followed the Design Science Research Methodology (DSRM) by Peffers et al. [12]. The proposed method aims to support BPM practitioners, particularly process architects or, more generally, workers of a BPM Center of Excellence that provides the service of process architecture maintenance [13]. The dBPA method was assessed over multiple evaluation activities. An experiment with soon-to-be BPM practitioners provided evidence that the dBPA method was perceived as useful and also as an improvement with respect to an assessed alternative. A later experiment with BPM practitioners allowed gathering perceptions on the suitability of the method to aid companies in their BPA endeavors. Prospective users of the dBPA method found it useful, objective, and clear while allowing to create a complete and understandable BPA models that enable to integrate processes and the software that automates them.

The remainder of the paper is structured as follows. Section 2 discusses the background and related work. Section 3 describes the methodology for developing the dBPA method. Section 4 presents the foundations of the method, which is then formalized and exemplified in Section 5. Section 6 reports on the evaluation of the method. Finally, a discussion is provided in Section 7 and conclusions are presented in Section 8.

2. Background and Related Work

Multiple definitions of the concept of BPA are available in the literature, e.g., [1,7,10]. Altogether, these definitions emphasize that a BPA: (i) offers a general perspective on the business process structure, and (ii) defines business process relations. Thus, this work considers that both processes and process relations are the constituting elements of a BPA. The present work adopts a view of the BPA and related concepts that conform with the ISO/IEC/IEEE 42010 standard for architecture description [14], cf. [15]. As shown in Figure 1, a BPA is expressed by a BPA description, which, in turn, is composed of one or more BPA views; each of which, in turn, is composed of one or more BPA models. A key BPA view is the one that is concerned with the high-level structure of the system of business processes known as process map [15–18], process landscape [2,9,19], process cartography [20,21], or process architecture (model) [10,22–24]. The remainder of this paper refers to it as the *BPA model*.

When the modeling task is complex, having a method—moreover if it is at least partially automated—eases the task, e.g., [25]. A method can be defined as the concepts, notation, and procedure that guide the structured execution of a complex task [26,27]. Accordingly, the present work understands a *BPA design method* as the concepts, notation, and procedure that guide the structured design of a BPA that is represented by a BPA model.

Following historic trends, the development of a BPA was one of the most common BPM projects carried out by companies during 2019 [28]. Having a BPA model eases the task of managing large collections of business processes, which can easily reach several hundreds of processes [1] in an integral manner. In line with this interest, a number of BPA design methods have been proposed over the last decades by both academia and consulting firms. According to Dijkman et al. [1], some of these approaches are based on goals (e.g., hiBPM [29]), actions (e.g., SCOR [30]), objects (e.g., the unfolding procedure [31]), reference models (e.g., PCF [32]), or functions (e.g., BPTrends Associates (BPTA) Methodology [10]).

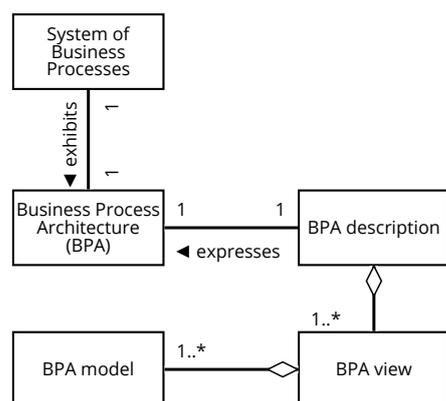


Figure 1. Model for the concepts of business process architecture and other related concepts (instantiation of [14] for a BPA.)

Some of the previously cited BPA design methods are described in detail in the following. This is expected to provide the reader with a set of examples of the available BPA design methods.

The Riva method [8] was developed and improved in time due to academic and consulting efforts. By first understanding the business an organization is in, the Riva method allows building a BPA model that is meant to represent the respective business domain. The domain knowledge is captured in a Unit of Work (UOW) diagram, which is then transformed via a repeatable technique into a BPA model. The inputs of the method are distributed organizational (domain) knowledge and data models; however, the steps of the method lack guidelines on how to use such inputs. The output of Riva is a BPA model referred to as Process Architecture Diagram, which depicts two types of processes (i.e., case processes and case management processes) and their relations (i.e., interaction and instantiation). In terms of its notation, the Riva method uses a non-standard one supported by a dedicated *MS Visio* stencil.

The BPTA methodology [10] is an integral BPM methodology developed by the 2006-founded training and consulting firm BPTrends Associates (BPTA). The input of the method is, again, organizational knowledge, which is found across different documents, databases and know-how distributed among members of the company. The output of the method is a two-level BPA model that is developed based on a number of meetings where the value chains and stakeholders of the company are analyzed. The first-level BPA model shows value streams relating identified processes and stakeholders, while the second-level BPA model shows hierarchical decomposition relations between the identified processes. Both of these models use a dedicated notation as well.

The unfolding procedure [31] is a method for creating a model called the Fractal Enterprise Model (FEM). According to its authors, the unfolding procedure is rooted in systems thinking, fractal models, enterprise modeling frameworks focusing on resources, and consulting experience. The FEM is a variation of a BPA model that shows business processes and their indirect relations by connecting processes to the assets (i.e., enterprise resources) they use and manage based on two types of patterns (i.e., process-assets archetypes and asset-processes archetypes). The inputs of the method are organizational (domain) knowledge and documents, yet no precise guidelines are provided on how to use such inputs. The output of the unfolding procedure is the FEM, which uses a dedicated notation supported by *Insight Maker*.

The previously described BPA methods, while illustrating available and used BPA design methods, also exemplify some common issues of these approaches as identified in [11]: (i) lack of structured business knowledge inputs, (ii) limited consideration of business process relations, and (iii) restricted use of industry-standard languages. As will be discussed in further detail in Section 3, these issues constitute the problem that motivates the design of the BPA method proposed in our work: the dBPA method. In the following,

we outline the position of our proposal with respect to available BPA design methods and their named issues.

First, regarding its inputs and unlike many approaches (e.g., [8,10,31]), the dBPA method does not solely rely on the knowledge of domain experts, but instead it also uses domain models as inputs. In this sense and following [1], the dBPA method can be categorized as an object-based approach. The use of domain models has been previously suggested as an input for BPA design methods in [8], but only in general terms and without precise guidelines of how to use their information. Second and in contrast to the few process relation types found in BPA models generated with current approaches (e.g., [8,10,31]), the dBPA method considers four types of process relations (i.e., composition, specialization, trigger, and resource flow). Third and unlike competing methods (e.g., [8,10,31]), the dBPA method uses the ArchiMate standard language for BPA models. The latter serves not just the purpose of facilitating tool support and standardization, but also allows for integrating the resulting BPA model with other elements within the enterprise architecture.

3. Methodology

The dBPA method is the main outcome of a Design Science Research (DSR) project [33] that followed the Design Science Research Methodology (DSRM) by [12], whose steps are: identify the problem and motivate, define objectives for the solution, design and development, demonstration, evaluation, and communication. The DSRM process for developing the dBPA is outlined in Figure 2 and described in detail in the following. Table 1 summarizes the iterations for developing the dBPA method together with the respective evaluation instances and modifications.

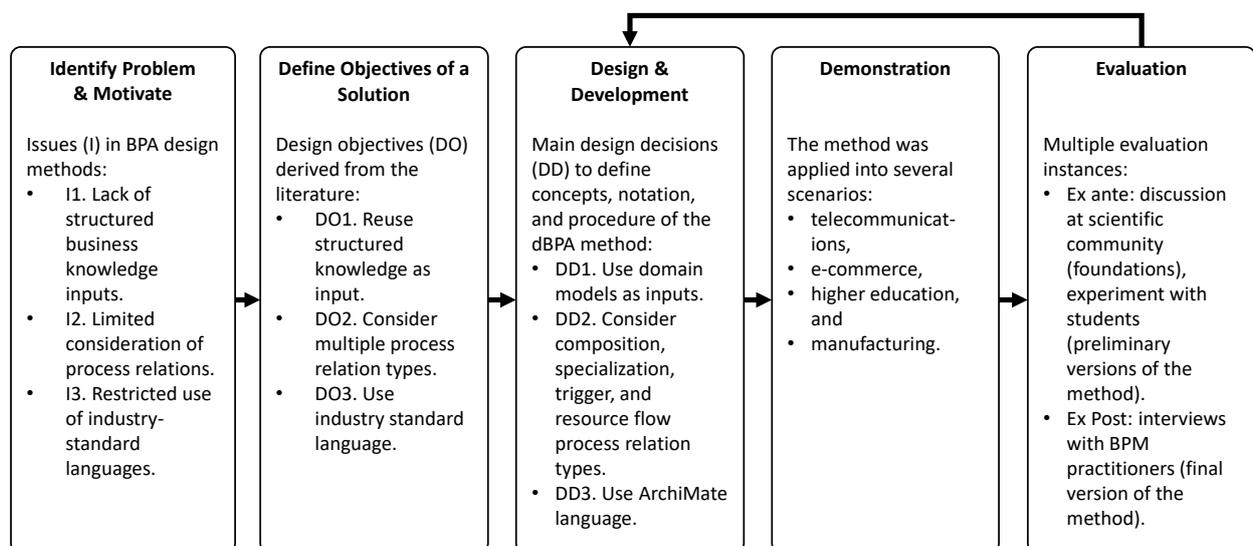


Figure 2. DSRM applied to develop the dBPA method. (The communication step has been omitted for simplicity).

3.1. Identify Problem and Motivate

The problem consists of the issues (I) of available BPA design methods [11] explained and exemplified in the following.

- I1. Lack of structured business knowledge inputs.** BPA design methods usually rely exclusively on unstructured inputs, e.g., expert knowledge and company's documents [8,10,31]. This issue challenges the efficiency and ease of use of methods since their success strongly relies on personal factors of those implementing the method, e.g., process architecture modeling experience, personal bias, and domain knowledge.
- I2. Limited consideration of business process relations** Though relations between business processes are key elements of BPAs [1], most BPA design methods support

only one or two types, e.g., [8,10,31]. It has been established that business processes can be interconnected by at least four relation types, namely composition, specialization, trigger, and resource flow [1]. Not supporting these types jeopardizes the generality of the method, and its resulting BPA models are likely to have completeness and fidelity shortcomings.

- **I3. Restricted use of industry-standard languages.** The use of industry-standard languages is very low among BPA design methods, with some exceptions, e.g., [34]. Instead, most methods use dedicated notations, e.g., [8,10,31]. This may be problematic in terms of tool support and integration with other models.

Table 1. Iterations of the dBPA method.

No.	Method Procedure	Evaluation Instance	Modifications
1	(1) Improve the expressiveness of the domain model, (2) Identify processes, (3) Revisit OLCs and identify their relations, (4) Categorize and relate processes, (5) Create BPA model.	Research community feedback in BPM 2016 Workshop. Design with BPM-trained senior undergraduates (n = 15) in telecommunications and e-commerce scenarios.	(1) Use the concept of existential dependency between classes for identifying trigger relations, (2) Use of the same standard for intermediate models, (3) Discard assumption of availability of OLC. (1) Consider additional elements of a domain model and OLC transitions, (2) Consider additional types of process relation, i.e., resource flow relations, (3) Discard the categorization of processes.
2	(1) Identify business entities and their states, (2) Build preliminary OLC network, (3) Improve expressiveness of domain model, (4) Identify processes, (5) Improve expressiveness of OLC network, (5) Build BPA model.	Design with BPM-trained senior undergraduates (n = 25) in higher education scenario.	(1) Rename activities and provide more precise description, (2) Generate ArchiMate viewpoint.
3	(1) Prepare domain model, (2) Identify business entities, (3) Identify states for each business entity, (4) Build OLC for each business entity, (5) Build BPA model.	Design with BPM practitioners (n = 6) in higher education scenario.	None.

BPA: business process architecture, BPM: business process management, OLC: object lifecycle.

3.2. Define Objectives of a Solution

A set of design objectives (DO) for the new method were defined based on the previously discussed issues.

- **DO1. Reuse structured knowledge as input.** This feature would ease the task of generating a BPA model.
- **DO2. Consider multiple process relation types.** This feature would support generality of the method along with completeness and fidelity of the resulting BPA models.
- **DO3. Use industry-standard language.** This feature would facilitate tool support and integration with other models.

3.3. Design and Development

The development of the dBPA method involved defining its concepts, notation, and procedure [26,27] while fulfilling the design objectives. The design decisions (DD) presented in the following lay the foundations (see Section 4) of the dBPA method as described in Section 5.

- **DD1. Use domain models as inputs for the dBPA method.** A paradigm-based strategy [35] was used for selecting the concepts and designing the procedure of the method. The chosen paradigm was the entity-centric approach for process modeling [36] applied at a business process architecture level. In this context, domain models are used as inputs as in [8].
- **DD2. Consider composition, specialization, trigger, and resource flow process relation types in the dBPA method.** The process type relations to be considered in the dBPA method were the most prominent ones according to the literature [1].
- **DD3. Use the ArchiMate language for the BPA models produced using the dBPA method.** The suitability of industry-standard languages was assessed to decide which one to use for representing BPA models. It was finally decided to use ArchiMate [37].

3.4. Demonstration

The dBPA method was applied to a number of scenarios, i.e., telecommunications, e-commerce, higher education, and manufacturing. The latter scenario is included in the paper as the running example discussed throughout Section 5.

3.5. Evaluation

The evaluations for the dBPA method included presenting the original version of the method at a BPM workshop to submit it to a discussion within the scientific community, conducting experiments and interviews with soon-to-be BPM practitioners and experienced BPM practitioners. The final evaluation activities are reported in Section 6.

4. Foundations

4.1. Elements of a BPA Model

The present work considers the core elements of a BPA model: processes and process relations. On the one hand, processes constitute atomic elements of a BPA model. On the other hand, four types of business process relations are identified [1,16,38], which are defined in the following as considered in the dBPA method.

- **Composition.** One process (referred to as *sub-process*) is part of another process (referred to as *parent* or *high-level process*).
- **Specialization.** One process (referred to as *specialized process*) is a variation with respect to another process (referred to as *general process*).
- **Trigger.** One process (referred to as *trigger source*) causes the start of another process (referred to as *trigger target*).
- **Resource flow.** One process (referred to as *resource source*) provides resources for the execution of another process (referred to as *resource target*).

4.2. Modeling Languages for BPA Model

The Business Process Model and Notation (BPMN) [39] is the de facto standard for detailed business process modeling. However, we discard its use in our method since BPMN has been found not suitable for creating BPA models [40]. Though BPMN has been adapted towards this end, e.g., [19,34,41], no consensus has been reached in this regard.

For the dBPA method, the decision was to use the ArchiMate Enterprise Architecture modeling language [37], as in [1]. ArchiMate is a standard for describing multiple facets of an enterprise architecture targeted to stakeholders such as process and IT architects. ArchiMate is a suitable industry standard for representing BPA models in the dBPA method because its syntax includes all the concepts considered in the BPA definition of the method.

4.3. Entity-Centric Process Modeling

A domain model represents the structure of a domain by showing its main concepts and how they relate to one another as classes and associations, respectively [42], e.g., Figure 3. Complementary to domain models, object lifecycles (OLCs) allow representing possible behavior for instances of domain model classes as states and transitions [43],

e.g., Figure 4. Often, the integration of these perspectives within process models tends to be an afterthought. In fact, data modeling is not a goal of BPMN 2.0 [39].

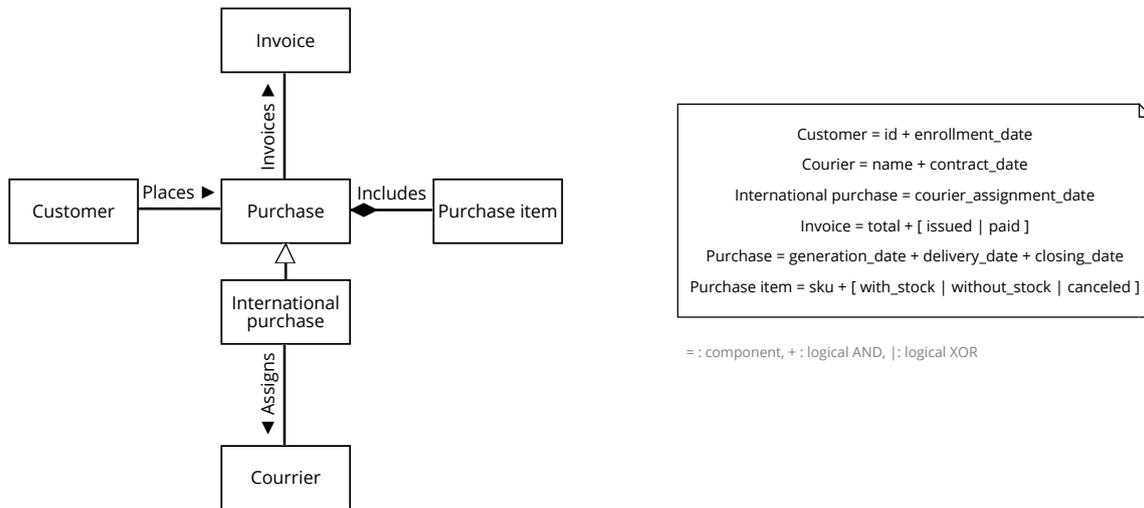


Figure 3. Domain model (left) and data dictionary (right).

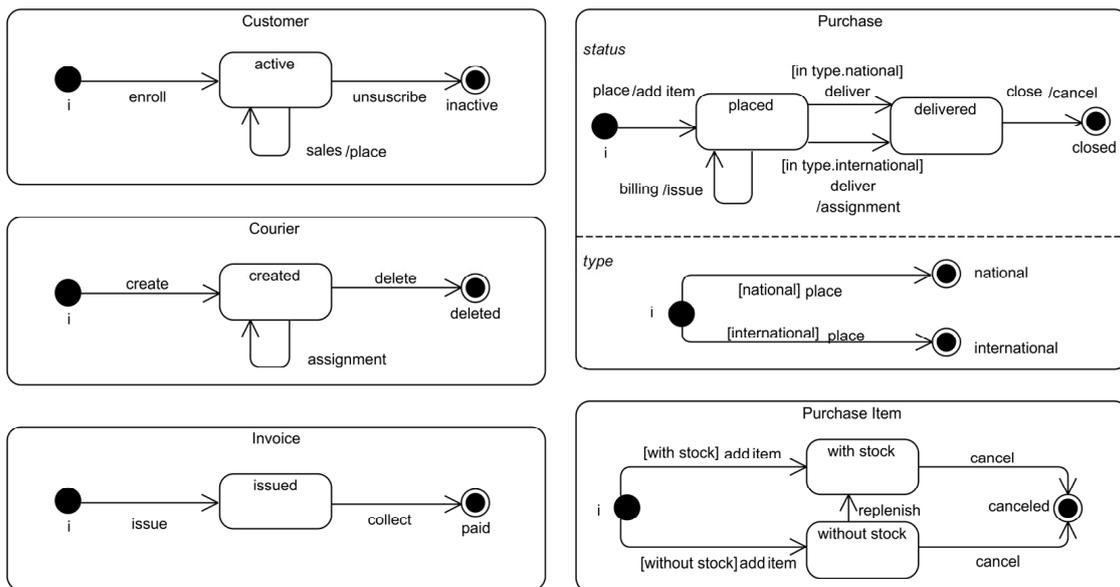


Figure 4. Object lifecycles.

In the traditional activity-centric approach to process modeling, business processes are elicited based on activities, e.g., [2,9,10]. An alternative to this is the elicitation of business processes according to the entity-centric paradigm, e.g., [44–46]. In entity-centric approaches, process modeling is based on the construct of a *business entity*: a relevant real-world concept handled by the organization [36], e.g., a guest check in a restaurant. A business entity can be represented as a domain model class with an associated OLC depicting how the business entity behaves [47].

The dBPA method contends, following [8], that based on the entity-centric paradigm, domain models may be used as inputs for the design of a BPA. This is analogous to the use of domain models as first-class citizens in Model-Driven Engineering (MDE) for software engineering [48]. The applicability of the dBPA method is facilitated by the availability of domain models, either by having them already developed within the organization

(e.g., in the context of an MDE project) or by the availability of such models in the literature (e.g., [49–52]).

Unlike the dBPA method, prior entity-centric BPA design approaches lack systematic guidance for the use of domain models for building BPA models. The position of the dBPA method regarding business entities and integration of domain models and OLCs is presented in the following.

First, regarding the identification of business entities and relations:

- Each domain model class defines, at most, one business entity.
- Each domain model association defines one or more relations between the object lifecycles (OLCs) of the partaking classes.

Second, integration of domain models and OLCs involves specializing the domain model classes with dynamic hierarchies (in a dynamic hierarchy, objects are allowed to migrate between sub-classes during their lifetime in the system [53], which applies when sub-classes are defined as possible states of the super-class) such that:

- A dynamic hierarchy is built by specializing the original class with sub-classes that hold a bi-univocal relation with the OLC states.
- The properties of the hierarchy are defined in a way that they are consistent with the organization of OLC states.

5. The dBPA Method

This section presents the concepts, notation, and procedure of the dBPA method.

5.1. Concepts

In the following, the concepts used in the dBPA method are defined.

Definition 1 (Class). A class c is an abstraction highlighting common properties of the elements of a set. A class is represented by a name and a state attribute j_S .

Definition 2 (Object lifecycle). An object lifecycle (OLC) l depicts the behavior of a class c and is represented by the tuple $(S, s_i, S_F, T, \Sigma, \eta)$, where:

- S is a finite non-empty set of data states,
- $s_i \in S$ is the initial data state,
- $S_F \subseteq S$ is a finite non-empty set of final data states,
- $T \subseteq S \times S$ is a finite non-empty set of time-consuming transitions describing the logical and temporal dependencies between data states,
- Σ is a finite non-empty set of transition labels with the structure $[\text{guard}]proc/\text{signal}$ where *guard* is a logic expression that enables a transition when true, *proc* stands for a behavior executed during the transition, and a *signal* represents an event that occurs within the respective *proc*; *guard* and *signal* can be omitted, and
- $\eta : T \rightarrow \Sigma$ is a function that assigns a label to each state transition.

Definition 3 (Generic object lifecycle). A generic object lifecycle l_g is represented by the tuple $(S_g, s_{i_g}, S_{F_g}, T_g, \Sigma_g, \eta_g)$ where $S_g = \{i, \text{created}, \text{deleted}\}$, $s_{i_g} = i$, $S_{F_g} = \{\text{deleted}\}$, $T_g = \{(i, \text{created}), (\text{created}, \text{deleted})\}$, and $\Sigma_g = \{\text{creation}, \text{elimination}\}$.

Definition 4 (Domain model). A domain model w is an abstraction of the structure of a domain and it is represented by the tuple $(C, A, H, H_s, C_{power}, Z_S, C_Z)$, where:

- C is a finite non-empty set of classes,
- $A \subseteq C \times C$ is a finite non-empty set of binary associations such that each association a is characterized by a name *assoc* and a direction *dir*. For each association, the function type_A returns the type of the association (i.e., aggregation, composition, or standard), and the function exist indicates whether the association involves an existential dependency (i.e., true, or false),

- $H : C \rightarrow \mathcal{P}(C)$ is a finite set of hierarchies such that each hierarchy h defines a relation between a general class called super-class and a set of specialized classes called sub-classes, and $type_H : H \rightarrow \{dynamic, static\} \times \{complete, incomplete\} \times \{disjoint, overlapping\}$ is a function that returns the properties of h ,
- $H_s \subseteq H$ is a finite set of dynamic hierarchies such $type_H[1] = dynamic$ in which sub-classes correspond to the OLC states S of the super-class,
- $C_{power} \subseteq C$ is the finite set of power-types such that each power-type c_{power} is a class whose objects are the sub-classes of other class,
- Z_S is a finite set of state-related enumerations,
- $C_Z \subseteq C$ is a finite set of classes such that for each class, its states S are specified by a state-related enumeration $z_S \in Z_S$ such that $S = z_S$.

Definition 5 (Data dictionary). A data dictionary dd is an organized list defining data in a given domain. A data dictionary dd is represented by the tuple $(C, K, K_s, comp)$ where:

- C is a finite non-empty set of classes,
- K is a finite non-empty set of components,
- $K_s \subseteq K$ is a finite set of state-related components such that for each class $c \in C$ it yields that $K_s \cap S \neq \emptyset$,
- $comp : K \rightarrow (C \cup K)$ is a function that maps each component to the data class or component to which it belongs.

Definition 6 (Business process architecture model). A business process architecture (BPA) model Θ shows the business process architecture of an organization and is represented by the tuple (Φ, R) , where:

- $\Phi = \Phi_H \cup \Phi_L$ is a finite non-empty set of business processes within the organization, such that $\Phi_H \cap \Phi_L = \emptyset$ and the business processes in Φ_H have a higher abstraction level than those in Φ_L ,
- $R = R_{comp} \cup R_{spec} \cup R_{trig} \cup R_{flow}$ is a finite non-empty set of business process relations such that R_{comp} is the set of composition relations r_{comp} , R_{spec} the set of specialization relations r_{spec} , R_{trig} is the set of trigger relations r_{trig} , and R_{flow} is the set of resource flow relations r_{flow} ,
- $r_{comp} : \Phi_L \times \Phi_H \rightarrow \{true, false\}$ is a function that returns true whenever $\phi_{l_s} \in \Phi_L$ referred to as a sub-process is part of $\phi_{h_t} \in \Phi_H$ referred to as a high-level process, such that one high-level process can have multiple sub-processes but one sub-process can only be part of one high-level process,
- $r_{spec} : \Phi_L \times \Phi_L \rightarrow \{true, false\}$ is a function that returns true whenever $\phi_{l_s} \in \Phi_L$, referred to as a specialized process, is a variation of $\phi_{l_t} \in \Phi_L$, referred to as a general process,
- $r_{trig} : \Phi_L \times \Phi_L \rightarrow \{true, false\}$ is a function that returns true whenever $\phi_{l_s} \in \Phi_L$, referred to as a trigger source, causes the start of $\phi_{l_t} \in \Phi_L$, referred to as a trigger target, and
- $r_{flow} : \Phi_L \times \Phi_L \rightarrow \{true, false\}$ is a function that returns true whenever $\phi_{l_s} \in \Phi_L$, referred to as a resource source, enables resources used in the execution of $\phi_{l_t} \in \Phi_L$, referred to as a resource target.

5.2. Notation

The notation used for the dBPA method is based on industry standards. For intermediate models, UML [54] is used: class diagrams for domain models (multiplicity has been omitted for simplicity) and state machine diagrams for object lifecycles. UML was chosen due to its high adoption, but alternative notations might also be used. For final models, i.e., BPA model, ArchiMate [37] is used.

ArchiMate provides a mechanism called viewpoint for defining a dedicated set of elements that focus on a specific aspect of the enterprise architecture [37]. Using this mechanism, a dedicated ArchiMate viewpoint named *BPA viewpoint* was defined for the notation of the dBPA method.

Creating an ArchiMate viewpoint requires selecting a subset of relevant concepts from its meta-model that, in the case of the dBPA method and as previously discussed, are

the concepts of business process and business process relations. However, some of these concepts can be mapped into multiple ArchiMate notation elements and thus violate the semiotic clarity principle by [55], e.g., the concept of business processes can be represented by ArchiMate’s business process symbol as well as by ArchiMate’s business interaction symbol. For addressing these issues, restrictions on the use of one symbol per concept are defined. The notation for BPA models in the dBPA method is the ArchiMate BPA viewpoint shown in Figure 5 and instantiated in the BPA model in Figure 6.

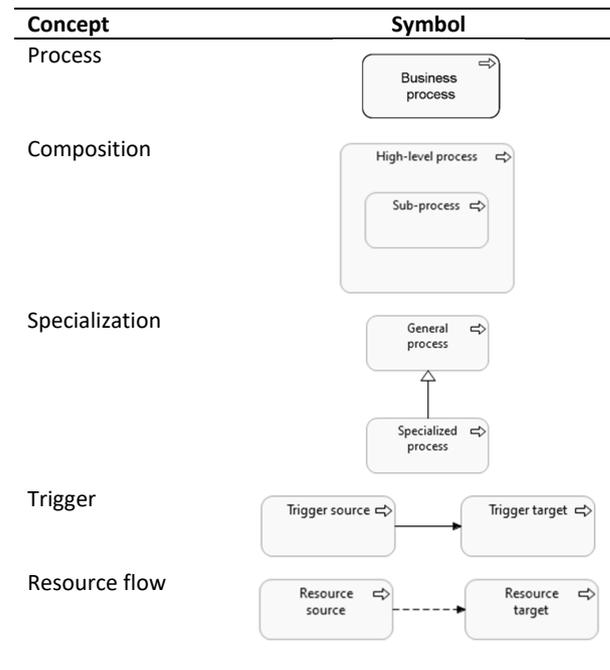


Figure 5. Concepts and symbols for the BPA viewpoint.

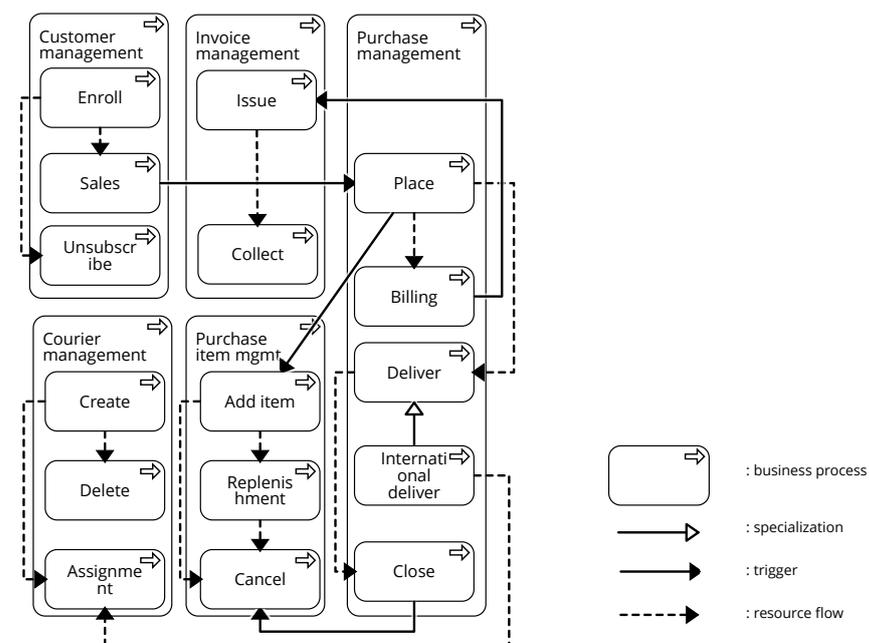


Figure 6. Business Process Architecture model.

5.3. Procedure

The dBPA method has five—generally speaking—sequential steps, as shown in Figure 7. Their descriptions are illustrated via an example of a *Retail Company* that manufactures and sells items to customers that place national or international purchases.

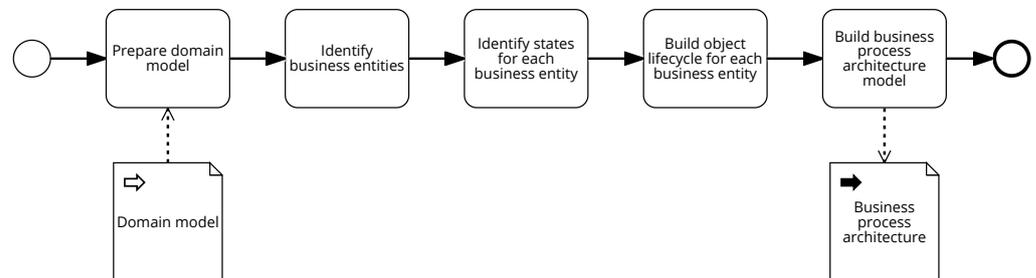


Figure 7. Domain-based Business Process Architecture (dBPA) method.

1. **Prepare domain model.** This step enables applying subsequent steps of the method directly. This is achieved by ensuring that the domain model w complies with some specifications, as stated in the following.
 - For each association $a \in A$ whose direction dir is not shown, show it.
 - For each association $a \in A$ whose name $assoc$ is formulated in a passive way (e.g., *is hired by*), formulate it in an active way (e.g., *hires*), and adjust the direction dir , if necessary.
 - For each power-type $c_{power} \in C_{power}$ transform it into a semantically-equivalent hierarchy h in the following way: considering there exists an association $a \in A$ between c_{power} and $c \in C$, replace a by the hierarchy $h \in H$ such that c is the super-class and c_i are the sub-classes, where i indicates the different objects of c_{power} .

In the example. The domain model in Figure 3 complies with the aforementioned specifications.

2. **Identify business entities.** This step allows identifying the set of business entities B that will be considered in the remainder of the method.
 - Each domain model class $c \in w$ corresponds to a business entity $b \in B$, with the exception of sub-classes partaking in a hierarchy h .

Sub-classes can be mapped into particular cases of business entities and thus are, conceptually, business entities. However, they are discarded from the set of business entities to be analyzed in the remainder of the method.

In the example. Business entities for the retail company are Customer, Courier, Invoice, Purchase, and Purchase item from the domain model in Figure 3. International purchase is discarded from the set of business entities due to being a sub-class.

3. **Identify states of domain model classes.** Business entities may have simple or non-trivial behavior, depending on the number of states it may transit during its lifetime.
 - For each business entity $b \in B$ with non-trivial behavior, its set of states S can be defined in the OLC l for the class c on which the business entity is based. Whenever no OLC is available or it is incomplete, the set of states S can be determined by the following (more than one of the options can be used):
 - The sub-classes partaking in the state-based dynamic hierarchy $h_s \in H_s$ of which $c \in w$ is the super-class (whenever $H_s \neq \emptyset$),
 - The possible values of state attribute j_s of data class $c \in w$ (whenever $Z_s \neq \emptyset$)
 - The possible values of the state attribute j_s of the corresponding data class $c \in dd$ whenever a complementary data dictionary dd is available for w ,
 - Domain-expert knowledge,

- For each business entity $b \in B$ with simple behavior (or no state information available), its set of states S corresponds to the set of states $S_g \in l_g$.

It is also important to ensure that:

- The set of states S of a business entity b include an initial state s_i and a non-empty set of final states S_F .
- After identifying the set of states for each business entity $b \in B$, all involved models are updated as needed for ensuring that they are consistent with one another.

In the example. The set of states of Invoice is $\{i, issued, paid\}$ and of Purchase item is $\{i, with\ stock, without\ stock, canceled\}$. In both cases, they were derived directly from the data dictionary shown in Figure 3. The set of states of Purchase is $\{i, placed, delivered, closed\}$, which was also derived from the data dictionary but in a more indirect way. The set of states of Customer is $\{i, active, inactive\}$ and was derived solely from domain-expert knowledge. The set of states for Courier is $\{i, created, deleted\}$, which was assumed to have a generic behavior.

4. **Build OLC for each business entity.** The OLC l (resp. l_g) for each business entity $b \in B$ needs considering individual behavior as well as interactions between business entities. If an OLC for the business entity b is available, it should be checked against the aspects described in the following and adapted if necessary.

4a. Build preliminary OLCs. Before addressing relations between OLCs of different business entities, the focus is on building a preliminary OLC for each business entity b . Such an OLC l must contain the following:

- **States.** The set of states S is identified in the previous step. Additionally, a concurrent state-independent region is added to the OLC l whenever the class c from which b was derived is the super-class of a state-independent hierarchy in the domain model. The sub-classes of the state-independent hierarchy are mapped into states within the state-independent region.
- **Transitions.** The set of transitions T that allow reachability of all states in S . T is identified using domain-expert knowledge.
- **Labels.** The set of labels Σ of which the label $\sigma_i \in \Sigma$ is assigned to $t_i \in T$. For each label σ_i , its mandatory *proc* element represents is the process that is executed for the corresponding transition to occur. For each label σ_i , its optional *guard* element refers to a condition for enabling the transition. This condition may verify process parameters (e.g., fixed-term contract), or concurrent states (using the form **in** *state.substate*... e.g., **in** *placed*).

In the example. Figure 4 shows the OLCs for the business entities of the retail company, and the OLCs of Invoice and of Purchase are described in the following to illustrate building preliminary OLCs. First, the OLC of Invoice shows the simplest case of an OLC that remains unaltered from its preliminary version. This OLC includes the set of states $\{i, issued, paid\}$ and the transitions with the set of labels $\{issue, collect\}$. The transition *issue* represents the existence of a process called *issue* whose goal is to transform an instance of Invoice into an *issued* one. On the other hand, the *collect* process transforms an *issued* Invoice into a *paid* one. Second, the OLC of Purchase is addressed. Though the preliminary version of this OLC differs from its final version shown in Figure 4, the use of a state-independent region and guards remains unchanged and is described in the following. The OLC of Purchase holds two concurrent regions: *status* representing behavioral aspects and *type* representing static aspects of a Purchase. Notice that static aspects are only part of an OLC whenever the corresponding class is part of a static hierarchy, as is the case of Purchase that differentiates between regular—national—purchases and international ones, as shown in Figure 3. The OLC of Purchase contains *guards*: some of them are related to parameters (e.g., *(national)*) while others correspond to state verifications (e.g., *(in type.national)*). The latter case represents two variations over the *deliver* process.

4b. Enrich OLCs with association-based interactions. Once preliminary OLCs are built, the interactions between them are included to obtain the final OLCs. An interaction between the OLCs of two business entities b' and b'' is given whenever an association $a_k \in A$ in the domain model relates the classes $c', c'' \in C$ on which $b', b'' \in B$ are based, respectively.

- **General case.** Let the relation between c' and c'' be represented by the association a_k , where c' is the source class and c'' is the target class according to the direction $dir(a_k)$. If the execution of process φ''_j results in the state change of the business entity b'' as defined by $t''_j \in T''$ so that φ''_j corresponds to the *proc* element of the respective transition label $\sigma''_j \in \Sigma''$, then there exists a transition $t'_i \in T'$ that relates to t''_j . Consequently, the preliminary OLC of l' needs to be modified so that φ''_j corresponds to the *signal* element of the label $\sigma'_i \in \Sigma'$ of transition $t'_i \in T'$. This modification allows representing that the process φ''_j is signaled from l' but executed in l'' as a consequence of the relation between b' and b'' .
- **Particular case.** Let the relation between c' and c'' be represented by the association a_k such that its type $type_A(a_k)$ is *composition*. In this case, the OLCs of the classes are strongly related. Particularly, the creation and elimination of the business entity based on the component class b^{part} are consequences of the creation and elimination of the business entity based on the composite class b^{whole} . Consequently, the preliminary OLCs of both business entities l^{part}, l^{whole} need to be modified regarding the labels of their initial and final transitions. Particularly, the *signal* element in the initial transition of l^{whole} needs being the same as the *proc* element in the initial transition of l^{part} , and the *signal* element of a subset of the final transitions of l^{whole} needs to be the same as the *proc* element in the final transitions of l^{part} .

In the example. The OLCs based on Customer and Purchase shown in Figure 4 allow illustrating the general case in which OLCs are enriched with association-based interactions. An interaction between the named OLCs is mapped from the association *Places* linking Customer and Purchase in the domain model (see Figure 3). Due to the direction of the *Places* association in the domain model, *place* is the *signal* of a transition in one OLC and the *proc* of a transition in the other OLC. An example of association-based interactions is shown between the OLCs shown in Figure 4 based on the classes Purchase and Purchase Item linked by the *Includes* composition in the domain model (see Figure 3). Due to the direction of the composition, *add item* (resp. *cancel*) is the *signal* of the initial (resp. final) transition in one OLC and the *proc* of the initial (resp. final) transitions in the other OLC.

5. **Build BPA model.** The BPA model Θ is built by mapping information in each OLC l and in the domain model w according to the following.
 - **High-level processes.** Assume the existence of one high-level business process φ_H for each business entity b . Each of these business processes is in charge of managing instances of its corresponding business entity during its OLC-defined lifetime. Accordingly, these processes are designated as the name of their corresponding business entity preceded by the word *management*, e.g., the *Product management* business process is in charge of handling instances of the business entity Product. The name of the business process can be exempt from this convention when an alternative name is more suitable.
 - **Composition relation.** Assume also that each high-level business process φ_H is composed of a set of processes designated sub-processes. Each of these sub-process φ_i is specified in the *proc* part of the labels of the corresponding OLC.
 - **Trigger relation.** Given the business entities b' and b'' derived from the classes c' and c'' , which are related by the association a_k with direction $dir(a_k) = c'c''$ and that involves an existential dependency ($exist(a_k) = true$), and a transition with $(proc, signal) = (\varphi_i, \varphi_j)$ in l' related to a transition with $proc = \varphi_j$ in l'' . Then,

processes φ_i and φ_j are related by a trigger relation in which the former is the trigger source and the latter is the trigger target.

- **Resource flow relation.** Relations between sub-processes of the same high-level processes are usually considered to be of type resource flow. These relations are derived by analyzing the respective OLC l . For each state $s \in l$, the *proc* part of the label of its incoming transition is the source of the resource flow relation, while the *proc* part of the label of its outgoing transition is the target of the resource flow relation. In case of self-transitions, they are considered only as outgoing transitions because they do not influence a state change. The second case considers the business entities b' and b'' derived from the classes c' and c'' , which are related by the association a_k with direction is $dir(a_k) = c'c''$ and that does not involve an existential dependency ($exist(a_k) = false$), and a transition with $(proc, signal) = (\varphi_i, \varphi_j)$ in l' related to a transition with $proc = \varphi_j$ in l'' . Given the aforementioned, then processes φ_i and φ_j are related by a resource flow relation in which the former is the resource source and the latter is the resource target.
- **Specialization relation.** A specialization relation is identified whenever two different transitions within the same OLC are labeled with the same *proc* but have a different *guard*. Domain-expert knowledge is used for identifying which of them corresponds to the general and specialized processes.

In the example. The resulting BPA model for the retail company is shown in Figure 6. The model shows five high-level processes, namely *customer management*, *courier management*, *invoice management*, *purchase management*, and *purchase item management*. Each of these high-level processes is composed of a number of sub-processes, e.g., *invoice management* is composed by *issue* and *collect*. Figure 6 also shows the remaining business process relations: trigger (e.g., *close* triggers *cancel*), resource flow (e.g., resource flow from *place* to *billing*), and specialization (e.g., *deliver* and *international deliver*).

6. Evaluation

The evaluation of the dBPA method involved conducting a set of evaluation activities. These activities had increasing realism in terms of tasks, systems, and users [56]; evolving from more artificial to more naturalistic approaches [57].

This section reports the two last evaluation instances used for developing the dBPA method. Each evaluation followed a rigorous design whose details are summarized in Table 2. The following subsections provide an overview of the experiments and findings of the named evaluation instances.

The subject matter of the evaluation is, on the one hand, the method itself and, on the other hand, the BPA models generated by the method. Both the method and its output are challenging to assess due to their inherent complexity and due to the restricted amount and availability of potential users.

6.1. Comparative Evaluation

A quantitative design with soon-to-be users of the method was conducted for assessing the dBPA method and comparing it with an alternative one, i.e., the BPTrends Associates (BPTA) methodology [10]. This alternative was selected because it is widely used in the industry, well-documented, and significantly different from the dBPA method (i.e., modeling paradigm and notation).

Subjects. The subjects were 25 Industrial Engineering senior undergraduates of the Pontificia Universidad Católica de Valparaíso in Chile. They represent, to some extent, the original context because of: (i) having a background in conceptual modeling due to a previous *System and Business Process modeling* course; (ii) being familiarized with Business Process Management (BPM) in the context of their current enrollment in a *Business Engineering* course.

Table 2. Overview of reported evaluation instances.

Aspect/Evaluation Instance	Comparative	Individual
Goal	Compare proposal with alternative method with soon-to-be users	Assess proposal with experienced prospective users
Date	October 2017	January 2021
Design	Quantitative	Mixed
Scenario	higher education	higher education
Subjects	25 BPM-trained senior undergraduates	6 BPM practitioners
Treatments	BPTA method [10], dBPA method	dBPA method
Variables	method performance: ease of use, usefulness, intention to use	method performance: ease of use, efficiency, generality, operability model quality: completeness, correctness impact of design decisions on method performance and model quality
Steps	(1) Training BPTA (video), (2) Apply BPTA in scenario, (3) Data collection for BPTA, (4) Training dBPA (video), (5) Apply dBPA in scenario, (6) Data collection for dBPA, (7) Data analysis	(1) Training dBPA (live), (2) Apply BPTA in scenario, (3) Data collection, (5) Data analysis
Data collection	Post-task survey	Post-task survey, Small group interviews
Data analysis	Wilcoxon signed rank tests [58]	Descriptive statistic, Affinity diagrams [59]

Data collection. Data were collected using a post-task questionnaire adapted from [60]. The questions (before randomization) are shown in Table 3 and were to be answered according to a 5-point Likert scale from 1 for *strongly disagree* to 5 for *strongly agree*. After rejecting observations of subjects producing incomplete questionnaires, a total of 23 observations were used for the analysis. Scores for questions formulated in a negative form (i.e., PEOU₃ and PEOU₄) were inverted.

The validity of the questionnaire was analyzed via the Spearman coefficient $\rho \in [-1,1]$ for *inter-item correlation*, due to the small sample size and ordinal data. This analysis led to removing PEOU₆ since it held significant negative correlations to other observable variables for Perceived Ease of Use. The reliability of the questionnaire was assessed using Chronbach's alpha coefficient $\alpha \in [0,1]$ due to its suitability for summated scales such as the Likert one used in the questionnaire. After removing PEOU₆, the coefficient for all latent variables, as well as for their combination, lay around this lower limit of adequate reliability (i.e., 0.7).

Data analysis. Six hypotheses were formulated (H1 to H6) whose *p*-values are provided in Table 4. On the one hand, hypotheses H1 to H3 ($H1 : median_{PEOU_B} \leq 3$, $H2 : median_{PU_B} \leq 3$, $H3 : median_{ITU_B} \leq 3$) assess the perceptions of subjects about the dBPA method regarding perceived ease of use, perceived usefulness, and intention to use. Testing each of these hypotheses required checking whether the observed median score of the respective variable significantly differed from the zero point of the 5-point Likert scale ($median_0 = 3$). One-sample Wilcoxon signed rank tests were used [58]: the null hypothesis states that observed median scores are smaller or equal than the zero point and the alternative hypothesis states that observed median scores are greater than the zero point. On the other hand, hypotheses H4 to H6 ($H4 : median_{PEOU_A} \leq median_{PEOU_B}$, $H5 : median_{PU_A} \leq median_{PU_B}$, $H6 : median_{ITU_A} \leq median_{ITU_B}$) compare the dBPA and BPTA methods. This required contrasting the observed median scores for each variable between two different treatments (i.e. A: BPTA and B: dBPA) as reported by the same subjects. The paired Wilcoxon signed rank test [58] was used: the null hypothesis states

that observed median scores of treatment *A* are equal or smaller than those for *B*, and the alternative hypothesis states that observed median scores of treatment *A* are greater than those for *B*.

Table 3. Items of the questionnaire (BPA: business process architecture, BP: business process).

Latent Variable	Observable Variable	Question
Perceived Ease of Use (PEOU)	PEOU ₁	I find learning the method for BPA design is easy.
	PEOU ₂	I find it would be easy for me to become skillful at using the BPA design method.
	PEOU ₃	I find using the method in the BPA design task required a lot of mental effort.
	PEOU ₄	I find using the method in the BPA design task required a lot of time.
	PEOU ₅	Overall, I find the BPA design method easy to use.
	PEOU ₆	I find I am now competent to use this method in practice.
Perceived Usefulness (PU)	PU ₁	I find the method provides an effective solution for designing BPAs.
	PU ₂	Overall, I find the method useful for designing BPAs.
	PU ₃	I find the method useful for identifying BP relations.
	PU ₄	I find the BP relations in the resulting model to be expressive.
	PU ₅	I find the BPA model resulting from using the method to be understandable.
	PU ₆	I find others, provided a quick explanation, would easily understand BP relations in the BPA resulting model.
Intention To Use (ITU)	ITU ₁	If I am given the task of designing a BPA in the future, my intention would be to use this method.
	ITU ₂	If someone asks me to recommend a BPA design method for clearly identifying BP relations, I predict I would recommend this one.

Table 4. Hypotheses.

Hypothesis	<i>p</i>
<i>H1</i>	ns
<i>H2</i>	***
<i>H3</i>	***
<i>H4</i>	ns
<i>H5</i>	ns
<i>H6</i>	ns

ns: $p > 0.05$, ***: $p \leq 0.001$

Results. The results of this evaluation instance are summarized in Figure 8. Statistical analysis of the results supports, on the one hand, that the dBPA method was perceived as useful ($p \leq 0.001$ for *H2*) but not easy to use ($p > 0.05$ for *H1*), and there was intention to use it in practice ($p \leq 0.001$ for *H3*). On the other hand, results suggested that the dBPA method outperforms the BPTA methodology in the aforementioned aspects ($p > 0.05$ for *H4*, *H5*, and *H6*). This constitutes preliminary evidence in favor of the method being an improvement with respect to currently available BPA design methods.

Limitations. The findings of the comparative evaluation need to be considered within a number of limitations. First, the dBPA method is compared to a single method. This was a design decision for avoiding the fatigue of the subjects. Additionally, we argue that proving that our proposal outperforms—in at least one aspect—one alternative method would justify designing the new method. Second, the use of the same scenario for the sequential application of the treatments involves the risk of a learning effect. A design aspect that partially counterbalances this is the significant difference between the two tested methods. Third, the use of soon-to-be practitioners as proxies for the final users of the

method challenges the external validity of the evaluation to some extent. However, though chosen subjects lacked vast practical experience, they have the technical qualifications of such practitioners. Fourth, a convenience sample was used instead of a probabilistic one, which risks statistical bias [61]. This decision was based on the scarce availability of subjects with the needed profile.

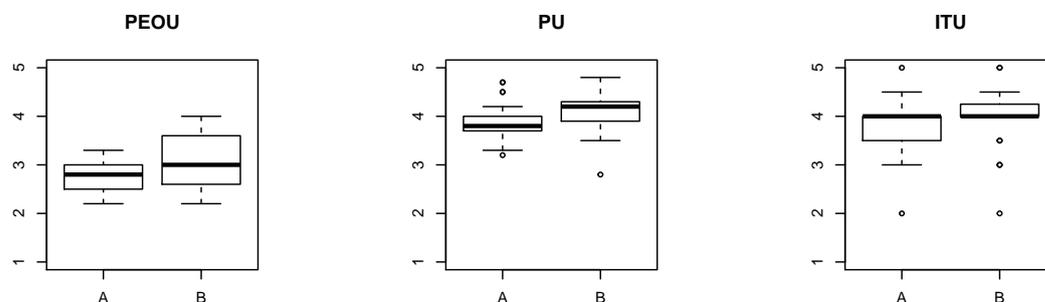


Figure 8. Summary of the data (PEOU: perceived ease of use, PU: perceived usefulness, ITU: intention to use, A: BPTA, and B: dBPA).

6.2. Individual Evaluation

A mixed-method design was conducted for assessing the dBPA method with its prospective users.

Subjects. The population under analysis is BPM practitioners. This population is small and our sample comes from a community of BPM practitioners named *Club CPO*. This 2009-founded community is a meeting point for BPM practitioners from (inter)national organizations operating in Chilean ground: it hosts regular events for sharing process-related experiences and training. We invited 43 members of the most active companies within the cited community to participate in an activity for the evaluation of the dBPA method. A total of six BPM practitioners participated in the activity, i.e., a 14% response rate. An overview of the participants is shown in Table 5.

Table 5. Practitioners involved in the evaluation.

Id	Job Title	Company Type	Employees
P1	Head of IT Operations and Service Management	Manufacturing	>200
P2	Head of Project Management Office	Healthcare	>200
P3	Process Engineer	Financial activities	25–200
P4	Head of Processes and Continuous Improvement	Real state	>200
P5	Process Coordinator	Public administration	>200
P6	Process Engineer	Financial activities	25–200

Data collection. Quantitative data were collected using a post-task questionnaire whose questions (before randomization) are shown in Table 6. Qualitative data were gathered next via an interview designed to assess the same aspects that were probed in the questionnaire. We conducted semi-structured group interviews guided by the open questions shown in Table 7. These interviews were recorded—after informed consent was provided by the participants—leading to 1 h 20 min of recording (40 min per each of the two groups, approximately).

Data analysis. Quantitative data were analyzed using descriptive statistics. Table 8 shows the median values for each of the analyzed variables.

To analyze qualitative data, we used the affinity diagramming technique. This technique allows organizing and making sense of unstructured, far-ranging, and seemingly dissimilar qualitative data into an artifact called the affinity diagram [59]. The affinity diagram consists of a hierarchical organization of verbal data generated during the interviews: taking this into consideration not only answers the questions asked but also other topics that emerged during the interviews.

Table 6. Items of the questionnaire.

Scope	Latent Variable	Observable Variable	Question
Method	Ease of use	U ₁	Overall, the method is easy to use.
		U ₂	After a training period, I would feel competent for using the method in practice.
	Efficiency	E ₁	I find that using the method required reasonable mental effort for the task of modeling a BPA.
		E ₂	I find that the time needed for applying the method to be reasonable for the task of modeling a BPA.
	Generality	G ₁	The method can be applied to an organization, regardless of its size.
		G ₂	I find that the method can be applied to an organization, regardless of the industry to which it belongs.
	Operationality	O ₁	The method is useful for building a BPA model.
		O ₂	The method defines the needed steps for building a BPA model.
Models	Completeness	C ₁	The resulting model shows the relevant process of the modeled BPA.
		C ₂	The resulting model shows the relevant process relations of the modeled BPA.
	Fidelity	F ₁	The resulting model lacks documentation mistakes (syntax).
		F ₂	The resulting model is valid for representing the BPA.

Table 7. Questions of the interview protocol.

Id	Question
1	What is your opinion about the method regarding ease of use, efficiency, generality, and operationality?
2	What is your opinion about the resulting models regarding completeness and fidelity?
3	A distinctive feature of the method is its use of domain models as inputs. In your view, what positive/negative implications does this carry for the method and its resulting models?
4	A distinctive feature of the method is its consideration of a variety of process relations. In your view, what positive/negative implications does this carry for the method and its resulting models?
5	A distinctive feature of the method is its use of ArchiMate. In your view, what positive/negative implications does this carry for the method and its resulting models?
6	Can you identify further (dis)advantages of the method?
7	Do you have any additional comments?

Table 8. Evaluation of the method and its resulting models and how they are influenced by design objectives (Likert scale from 1: Very unfavorable to 5: Very favorable) participants P1 to P5.

	Overall	DO1	DO2	DO3
Ease of use	4.5	4.0	3.0	4.0
Efficiency	4.0	4.0	5.0	3.0
Generality	3.0	4.0	4.0	4.0
Operationality	5.0	5.0	5.0	4.0
Completeness	5.0	5.0	5.0	5.0
Fidelity	4.0	4.0	4.0	4.0

The steps for creating an affinity diagram are creating notes, clustering notes, walking the wall, and documenting [62]. In the *creating notes* step, two or more researchers individually play the recording and write down (into sticky notes) short sentences, descriptions, or citations that capture the essence of the different portions of data generated by each participant. During the *clustering notes* step, researchers bring together their notes and

jointly organize them into preliminary categories (using additional sticky notes) by moving the notes around in a dedicated board, i.e., the wall. The *walking the wall* step has the goal of iterating over the previous step by rearranging notes and categories and building the category hierarchy (using arcs that connect them). The output of this step is the affinity diagram that is, finally, described in the *documenting* step.

In the present evaluation, the affinity team—i.e., researchers taking notes as well as building the affinity diagram—consisted of two of the paper authors. Data were analyzed in the mother tongue of researchers and participants, namely Spanish. Due to the sanitary context, their work was conducted in a digital fashion by using a dedicated template of the Miro software. Each of the generated notes had an identifier of the participant from whom it was generated (P1, P2, . . . , P6), as well as from the researcher that generated it. A total of 156 notes were generated, from which 15% were discarded due to reporting contextual information (cf. [62]). The English version of the resulting affinity diagram is shown in Figure 9.

Results. According to quantitative results summarized in Table 8, the highest-ranked aspect of the dBPA method was operationality ($median_{operationality} = 5$) followed by ease of use ($median_{easeofuse} = 4.5$), efficiency ($median_{efficiency} = 4.0$), and generality ($median_{generality} = 3.0$). Whereas the resulting models of the method were best ranked in terms of completeness ($median_{completeness} = 5.0$) followed by fidelity ($median_{fidelity} = 4.0$). Results also showed that the reuse of structured knowledge as an input (DO1) had a favorable impact, which was highest for operationality and completeness ($median_{DO1,operationality} = median_{DO1,completeness} = 5.0$). The consideration of multiple process relation types (DO2) had a favorable impact, which was highest for efficiency, operationality, and completeness ($median_{DO2,efficiency} = median_{DO2,operationality} = median_{DO2,completeness} = 5.0$). Finally, the use of an industry-standard language (DO3) had a favorable impact as well, which was highest for completeness ($median_{DO3,operationality} = 5.0$).

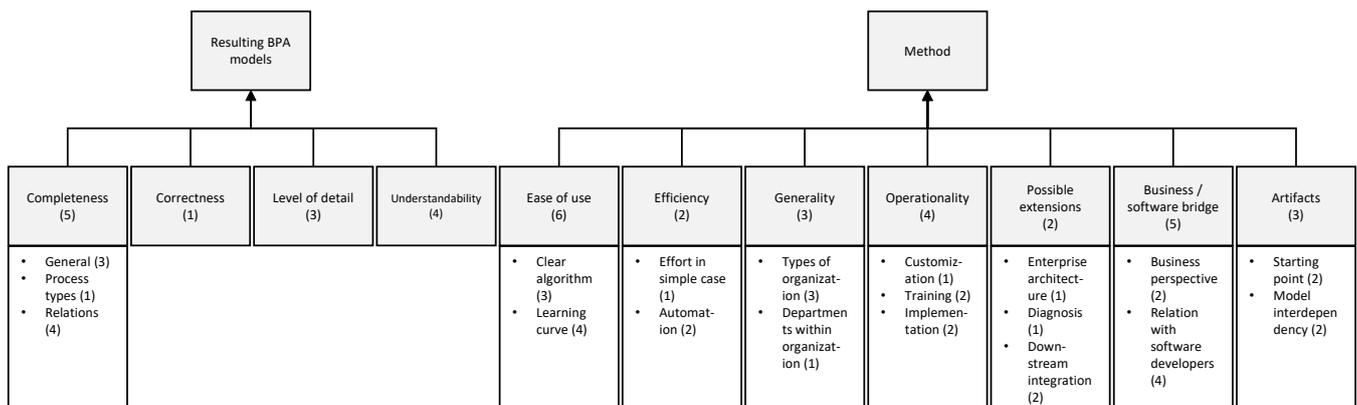


Figure 9. Affinity diagram: the numbers indicate how many participants referred to the (sub)category.

For qualitative data analysis, recordings of the interviews were analyzed by creating the affinity diagram in Figure 9, and described in detail in Tables 9 and 10. To illustrate the results, we provide quotes from participants, translated to English. According to this analysis, practitioners agreed that the method was useful to achieve its goal of building a BPA model. The ease of use of the method was positively influenced by its clear procedure, to the point that the potential for automation was discussed. Additionally, a learning curve was identified by the fact that not all interviewees were experts on domain models. However, there was a consensus that having an interdisciplinary team with IT experts would facilitate this issue. Moreover, practitioners identified a salient value of the dBPA method for integrating the work of process experts and IT professionals automating those processes. In terms of efficiency and generality, the dBPA method seems to be more adequate for core processes of mid-size companies with in-house IT capabilities and a

small portion of manual processes. The resulting BPA models generated by using the dBPA method were perceived as complete and correct. The latter was seen as a by-product of the clear procedure where consistency was continuously verified. Regarding the impact of design decisions in the dBPA method and its resulting models, the following was observed. Overall, though the use of domain models as inputs presented some challenges (i.e., learning curve and maintainability), it had the great benefit of making the method more reality-based and objective. The consideration of multiple process relation types was seen as a key factor for the completeness of the resulting BPA models; however, this also might lead to cluttered models, for which the use of alternative/dynamic views might be advisable. Finally, it was perceived that the use of the ArchiMate language was not a problematic issue among participants, who mostly found it easy to understand.

Limitations. The individual evaluation of the method needs to be understood within a number of limitations. First, the small sample size can be seen as a threat to external validity. However, we argue that the population of users of the method is quite small as well. Moreover, a positive aspect of the sample is its inter-subject variability in terms of levels of experience and the organization in which the subjects work. Second, the qualitative data analysis technique (i.e., affinity diagrams) cannot be fully reproduced because of the subjectivity of the members of the affinity team. However, the use of a team—instead of a single person—in the analysis provides a counterbalance for individual subjectivity. In the same line, a mixed-method technique for data gathering was used [63] to counteract this and other internal validity issues.

Table 9. Affinity diagram results for category ‘Resulting BPA models’.

Sub-Category	Details
Completeness	Half of the participants manifested that the BPA models obtained with the method provide enhanced completeness, e.g., “in contrast to the [BPA] models we have now, the method would allow obtaining more complete [BPA] models” (P4). Most participants also highlighted that representing process relations plays a key role in this regard, e.g., “it helps to understand processes and how they relate to each other” (P1). One participant raised the issue that the method, though suitable for core processes, might not be adequate for other types of processes, e.g., support processes.
Correctness	One participant commented on how the method ensured correctness of the resulting models: “in terms of correctness, I see it good: all the steps of the method facilitates identifying all possible mistakes” (P3).
Level of detail	Half of the participants referred to the level of detail of the resulting models. The level of detail was perceived as “an intermediate point between detailed process models and a strategic overview of the processes architecture” (P5). The level of detail was found “favorable for maintainability of the model since implementation details of processes are abstracted” (P1).
Understandability	Half of the participants perceived the models to be easy to follow—despite the fact that most were unfamiliar with ArchiMate—, e.g., “the resulting models are understandable, easy to grasp, visualize, and comprehend” (P1). A participant had doubts if the resulting models were “non-expert friendly” (P3), and another found the model conveyed too much information, i.e., “I would rather see two versions of the model: one with the arrows, and one without them” (P4).

Table 10. Affinity diagram results for category ‘Method’.

Sub-Category	Details
Ease of use	Half the participants claimed that the method was objective, complete, and clear, e.g., “it is well-defined algorithm” (P2). In terms of the learning curve, the background of the participant seemed to be relevant, e.g., “I believe that it [the method] calls for [UML] notation knowledge, which is more usual in IT professionals” (P5).
Efficiency	In terms of efficiency, a third of the participants perceived the method as adequate for automation, e.g., “All of this can be automated” (P2). A participant claimed a considerable effort might be needed to apply the method to a simple case (P2).
Generality	A participant perceived the method to be general in terms of “being applicable to any company, to different departments within a company, as well as to the company as a whole” (P1). Other participants raised concerns about the generality of the method related to company size, the proportion of process types, as well as to the IT capabilities within the company. Overall, they believe the method to be adequate for middle-size companies with in-house IT capabilities, and a small proportion of manual processes.
Operationality	A participant pointed out that the method was prone to customization: “it could be adapted to resources of the company, e.g., using alternative notations” (P6). A third of the participants claimed that training would be needed to apply the method. A third of the participants made suggestions to improve operationality, i.e., use of templates (P5) and clear definition of roles for method execution (P6).
Possible extensions	A participant with a background in ArchiMate pointed out the potential to integrate other elements of the language (i.e., capabilities) lead to a broader architectural view (P2). Another participant discussed the potential of using the model for a diagnosis of the process architecture (P5). Both participants also pointed out how the information of the resulting models could be used to be integrated to define parameters of detailed process models and business rules.
Business/software bridge	A third of the participants perceive that the method is suitable to provide a business-level perspective. Most participants perceive the value of the method in bridging the process level with the implementation level, e.g., “this might close the gap between process analysts and software developers [for developing applications that automate the named processes]” (P2).
Artifacts	A third of the participants perceived that using the domain model as a starting point to be beneficial. This allowed building the model “from a reality-based artifact” (P4). A third of the participants also found the interdependence between the domain and the architecture model to be a challenge for maintainability.

7. Discussion

A DSR project was conducted for designing a novel method for BPA design. The use of such an approach was a suitable strategy for generating and validating an artifact that tackles some issues of currently available alternatives. The dBPA method was designed for addressing three design objectives derived from the literature [11]: reuse of structured business knowledge as input (DO1); consideration of multiple process relation types (DO2); and use of an industry-standard language (DO3).

First, the reuse of structured business knowledge was implemented by using domain models as inputs for the BPA design method (DD1). This constitutes an entity-centric approach and, as such, it relies on the assumption that the structure and behavior of things that are important for a given organization (i.e., business entities), which set the ground for the process structure to be found in the organization. However, unlike other entity-centric BPA approaches—e.g., [8,31]—this proposal provides precise guidelines on how to use domain models to convey information for building the BPA model. In broad terms, the dBPA method uses domain model classes for the identification of business entities, and domain model associations together with object lifecycles (OLCs) for the identification of business process relations.

Second, the resulting BPA models of the dBPA method consider composition, specialization, trigger, and resource flow process relations (DD2). Though discussed in the literature of business process relations, see [1,16,38], these four types of relations have been rarely simultaneously incorporated in BPA design approaches so far. The integration of the variety of possible process relations allows providing a more realistic picture of the structure of business processes: in terms of model quality, this contributes to improving the model completeness. Additionally, as posited by [64], the extent to which a business process is interconnected to other business processes in a BPA can be used for prioritizing process improvement initiatives in an organization.

Third, a dedicated ArchiMate viewpoint—designated the BPA viewpoint—was proposed for the BPA models of the dBPA method (DD3). Though some recent works relate BPA models with ArchiMate, e.g., [65,66], no BPA design method has, so far, has used ArchiMate for the specification of BPA models. It is usually the case that BPA design methods use ad hoc languages. An advantage of using standard notations is the availability of tool support. In this work, for instance, *Signavio Process Manager 14.3* was used for building domain and BPA models (Figures 3 and 6), and *Visual Paradigm 16.1* for building OLCs (Figure 4), yet there are other tools for these purposes in the market. The paper shows the use of UML class diagrams for domain models, state machine diagrams for OLCs, and an ArchiMate viewpoint for BPA models. However, the method is not language-dependent and, for example, entity-relationship diagrams, Petri nets, and other model types could be used alternatively. The notations presented in the paper are, however, widely used and thus recommended for the target users of the dBPA method.

The comparative evaluation reported in this paper is a step towards justifying that the dBPA method might be an improvement in comparison to other available BPA design methods. However, this conjecture might be better backed up by further research. The evaluation of the dBPA method with BPM practitioners revealed they perceived it as useful to achieve its goal with the benefits of being objective and clear and allowing to create complete and understandable BPA models that enable the integration of processes and the software that automates them.

Regarding future empirical work, it would be interesting to complement user perceptions with more objective measures. For instance, the Method Evaluation Model (MEM) [67] is an adaptation of [60] that measures the actual usefulness and ease of use of methods. In a similar vein, the BPA models resulting from applying the dBPA method might be analyzed further. For example, it would be interesting to assess the cognitive effectiveness of BPA models to complement the information about perceptions. Furthermore, a model quality approach such as the SEMiotic QUALity framework (SEQUAL) [68] might be used to compare BPA models generated using alternative BPA design methods.

In addition to further empirical work, two main future research topics are envisioned. On the one hand, a new version of the dBPA method could explore further aspects of domain models, e.g., association multiplicity (cf. [69]), and their impact on the resulting BPA model. In line with this, it would be interesting to perform a sensitivity study for assessing the impact of the variations of different parameters of domain models in the resulting BPA models. On the other hand, the dBPA method adopts some simplifications. First, it considers a one-to-one relation between a business process and an OLC transition. However, in some cases there might be a one-to-many relation between a business process and OLC transitions. For the sake of flexibility, future versions of the dBPA method should address this issue. Business process decomposition under the entity-centric paradigm has also been discussed in approaches such as [70]. Second, the dBPA method does not differentiate between the most general composition and the most specific aggregation process relations. The latter, which would allow representing processes that can be reused, is not included in the dBPA method in its current version but might be considered in the future.

8. Conclusions

The present work focuses on process architectures, which is a growing research field with a number of open challenges. For instance, pivotal conditions for widespread adoption of BPA design methods are notation standardization, integration, and adequate tool support. The focal point and contribution of our study was in the direction of designing an artifact that could overcome some limitations of current approaches for BPA design. Given the nature of the task, the work was carried out as a Design Science Research project aiming at building and evaluating the new method, which was termed the domain-based BPA (dBPA) method, due to using domain models as the main input for eliciting business processes and their relations. By using domain model information, the dBPA method allows obtaining prescriptive BPA models that can be used as a validation tool against which to assess a current BPA model. Evaluation of the dBPA method provided evidence in favor of being a tool to aid the challenging task of building a BPA model with a number of benefits regarding integration with other aspects of the organization.

Author Contributions: Conceptualization, F.G.-L. and G.B.; methodology, F.G.-L.; validation, F.G.-L., J.M.-G. and M.S.; writing—original draft preparation, F.G.-L.; writing—review and editing, J.M.-G. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Agency for Research and Development (ANID) under ANID FONDECYT 3210147, ANID FONDECYT 1200206, and ANID FONDECYT 1220202. The work was also supported by Pontificia Universidad Católica de Chile under Beca Postdoctorado Escuela de Ingeniería and by CORFO Engineering 2030 14ENI2-26862.

Institutional Review Board Statement: Ethical review was considered within the approval of the doctoral research project of the first author at the School of Industrial Engineering at Pontificia Universidad Católica de Valparaíso.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dijkman, R.; Vanderfeesten, I.; Reijers, H. Business Process Architectures: Overview, Comparison and Framework. *Enterp. Inf. Syst.* **2016**, *10*, 129–158. [CrossRef]
2. Weske, M. *Business Process Management: Concepts, Languages, Architectures*; Springer: Berlin, Germany, 2019.
3. Kremser, W.; Pentland, B.T.; Brunswicker, S. Interdependence Within and Between Routines: A Performative Perspective. *Res. Sociol. Organ.* **2019**, *16*, 79–98.
4. Reijers, H.A. Business Process Management: The evolution of a discipline. *Comput. Ind.* **2021**, *126*, 103404. [CrossRef]
5. Lankhorst, M.M. Enterprise architecture modelling—The issue of integration. *Adv. Eng. Inform.* **2004**, *18*, 205–216. [CrossRef]
6. Ross, J.W.; Weill, P.; Robertson, D. *Enterprise Architecture as Strategy: Creating a Foundation for Business Execution*; Harvard Business Press: Boston, MA, USA, 2006.
7. Malinova, M.; Leopold, H.; Mendling, J. An Empirical Investigation on the Design of Process Architectures. *Wirtschaftsinformatik* **2013**, *75*, 1197–1211.
8. Green, S.; Ould, M.A. The Primacy of Process Architecture. In Proceedings of the CAiSE Workshops, Riga, Latvia, 7–11 June 2004; pp. 154–159.
9. Dumas, M.; Rosa, M.L.; Mendling, J.; Reijers, H.A. *Fundamentals of Business Process Management*; Springer: Berlin, Germany, 2013.
10. Harmon, P. *Business Process Change: A Business Process Management Guide for Managers and Process Professionals*; Elsevier: Cambridge, MA, USA, 2014.
11. Gonzalez-Lopez, F.; Bustos, G. Business process architecture design methodologies—A literature review. *Bus. Process Manag. J.* **2019**, *25*, 1317–1334. [CrossRef]
12. Peffers, K.; Tuunanen, T.; Rothenberger, M.; Chatterjee, S. A design science research methodology for information systems research. *J. Manag. Inf. Syst.* **2007**, *24*, 45–77. [CrossRef]
13. Jesus, L.; Macieira, A.; Karrer, D.; Rosemann, M. A Framework for a BPM Center of Excellence. 2009. Available online: <http://www.bptrends.com> (accessed on 13 January 2022).
14. ISO. ISO/IEC/IEEE 42010:2011, Systems and Software Engineering—Architecture Description. 2011. Available online: <http://www.iso-architecture.org/42010/> (accessed on 13 January 2022).

15. Poels, G.; Garcia, F.; Ruiz, F.; Piattini, M. Architecting Business Process Maps. *Comput. Sci. Inf. Syst.* **2020**, *17*, 117–139. [CrossRef]
16. Malinova, M.; Leopold, H.; Mendling, J. *A Meta-Model for Process Map Design*; CAiSE (Forum/Doctoral Consortium). 2014. pp. 25–32. Available online: <http://ceur-ws.org/Vol-1164> (accessed on 13 January 2022).
17. Chourabi, H.; Mellouli, S.; Bouslama, F. Modeling e-government Business Processes: New Approaches to Transparent and Efficient Performance. *Inf. Polity* **2009**, *14*, 91–109. [CrossRef]
18. Heinrich, B.; Henneberger, M.; Leist, S.; Zellner, G. The process map as an instrument to standardize processes: Design and application at a financial service provider. *Inf. Syst.-Bus. Manag.* **2009**, *7*, 81–102. [CrossRef]
19. Polancic, G. BPMN-L: A BPMN extension for modeling of process landscapes. *Comput. Ind.* **2020**, *121*, 103276. [CrossRef]
20. Mu, W.; Bénaben, F.; Pingaud, H. A Methodology Proposal for Collaborative Business Process Elaboration Using a Model-driven Approach. *Enterp. Inf. Syst.* **2015**, *9*, 349–383. [CrossRef]
21. Mu, W.; Bénaben, F.; Pingaud, H. Collaborative process cartography deduction based on collaborative ontology and model transformation. *Inf. Sci.* **2016**, *334–335*, 83–102. [CrossRef]
22. Barros, O. A process architecture pattern and its application to designing health services: Emergency case. *Bus. Process. Manag. J.* **2019**, *26*, 513–527. [CrossRef]
23. Babar, Z.; Yu, E.; Carbajales, S.; Chan, A. Managing and Simplifying Cognitive Business Operations Using Process Architecture Models. In *Advanced Information Systems Engineering, CAiSE 2019 (LNCS 11483)*; Springer: Berlin, Germany, 2019.
24. Ould, M. *Business Process Management: A Rigorous Approach*; Meghan-Kiffer Press: Tampa, FL, USA, 2005. Available online: <https://www.bcs.org/books/bpm> (accessed on 13 January 2022).
25. Hewelt, M.; Pufahl, L.; Mandal, S.; Wolff, F.; Weske, M. Toward a methodology for case modeling. *Softw. Syst. Model.* **2019**, *19*, 1–27. [CrossRef]
26. Sandkuhl, K.; Seigerroth, U. Method engineering in information systems analysis and design: A balanced scorecard approach for method improvement. *Softw. Syst. Model.* **2019**, *18*, 1833–1857. [CrossRef]
27. Goldkuhl, G.; Lind, M.; Seigerroth, U. Method integration: The need for a learning perspective. *IEEE Proc. Softw.* **1998**, *145*, 113–118. [CrossRef]
28. Harmon, P. The State of Business Process Management 2020—A BPTrends Report. 2020. Available online: <https://www.bptrends.com/bptrends-state-of-business-process-management-2020-report/> (accessed on 13 January 2022).
29. Lapouchnian, A.; Yu, E.; Sturm, A. Re-designing process architectures towards a framework of design dimensions. In Proceedings of the 2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS), Athens, Greece, 13–15 May 2015; pp. 205–210.
30. Stewart, G. Supply-chain operations reference model (SCOR): The first cross-industry framework for integrated supply-chain management. *Logist. Inf. Manag.* **1997**, *10*, 62–67. [CrossRef]
31. Bider, I.; Perjons, E.; Elias, M.; Johannesson, P. A Fractal Enterprise Model and its Application for Business Development. *Softw. Syst. Model.* **2017**, *16*, 663–689. [CrossRef]
32. APQC. Process Classification Framework (PCF) Version 7.2. 2011. Available online: <https://www.apqc.org/process-frameworks> (accessed on 13 January 2022).
33. Hevner, A.R.; March, S.T.; Park, J.; Ram, S. Design Science in Information Systems. *MIS Q.* **2004**, *28*, 75–105. [CrossRef]
34. Barros, O.; Julio, C. Enterprise and process architecture patterns. *Bus Process Manage J.* **2011**, *17*, 598–618. [CrossRef]
35. Ralyté, J.; Deneckère, R.; Rolland, C. Towards a Generic Model for Situational Method Engineering. In Proceedings of the CAiSE 2003, LNCS Vol. 2681, Velden, Austria, 16–20 June 2003; Springer: Berlin, Germany, 2003.
36. Nigam, A.; Caswell, N.S. Business artifacts: An approach to operational specification. *IBM Syst. J.* **2003**, *42*, 428–445. [CrossRef]
37. The Open Group. ArchiMate® Version 3.0.1 Specification. 2017. Available online: <https://www.opengroup.org/archimate-forum> (accessed on 13 January 2022).
38. Kurniawan, T.A.; Ghose, A.K.; Le, L.S.; Dam, H.K. On formalizing interprocess relationships. In *Business Process Management Workshops; BPM 2011 (LNBIP 100)*; Daniel, F., Barkaoui, K., Dustdar, S., Eds.; Springer: Berlin, Germany, 2012; pp. 75–86.
39. OMG. Business Process Model and Notation (BPMN) Version 2.0. 2011. Available online: <https://www.omg.org/spec/BPMN/2.0/> (accessed on 13 January 2022).
40. Malinova, M.; Mendling, J. Why is BPMN not appropriate for process maps? In Proceedings of the ICIS 2015 Proceedings, Fort Worth, TX, USA, 13–16 December 2015.
41. Eid-Sabbagh, R.H.; Dijkman, R.; Weske, M. Business Process Architecture: Use and Correctness. In Proceedings of the Business Process Management, BPM 2012 (LNCS 7481), Tallinn, Estonia, 3–6 September 2012; Springer: Berlin, Germany, 2012; pp. 65–81.
42. Le, D.M.; Dang, D.H.; Nguyen, V.H. On domain driven design using annotation-based domain specific language. *Comput. Lang. Syst. Struct.* **2018**, *54*, 199–235. [CrossRef]
43. Eshuis, R.; Van Gorp, P. Synthesizing object life cycles from business process models. In Proceedings of the International Conference on Conceptual Modeling, Florence, Italy, 15–18 October 2012; Springer: Berlin, Germany, 2012; pp. 307–320.
44. Hull, R.; Damaggio, E.; De Masellis, R.; Fournier, F.; Gupta, M.; Heath, F.T., III; Hobson, S.; Linehan, M.; Maradugu, S.; Nigam, A.; et al. Business artifacts with guard-stage-milestone lifecycles: Managing artifact interactions with conditions and events. In Proceedings of the DEBS 2011, New York, NY, USA, 11–15 July 2011; ACM: New York, NY, USA, 2011; pp. 51–62.
45. Popova, V.; Fahland, D.; Dumas, M. Artifact Lifecycle Discovery. *Int. J. Coop. Inf. Syst.* **2015**, *24*, 1550001. [CrossRef]

46. Eck, M.L.V.; Sidorova, N.; van der Aalst, W. Guided Interaction Exploration and Performance Analysis in Artifact-Centric Process Models. *Bus. Inf. Syst. Eng.* **2019**, *61*, 649–663.
47. Ryndina, K.; Küster, J.M.; Gall, H. Consistency of business process models and object life cycles. In Proceedings of the International Conference on Model Driven Engineering Languages and Systems, MODELS 2006 (LNCS 4364), Genoa, Italy, 2 October 2006; Springer: Berlin, Germany, 2006; pp. 80–90.
48. Rodrigues da Silva, A. Model-driven engineering: A survey supported by the unified conceptual model. *Comput. Lang. Syst. Struct.* **2015**, *43*, 139–155. [[CrossRef](#)]
49. Sapien. MIT Enterprise Architecture Guide. 2004. Available online: <https://barsand.files.wordpress.com/2014/09/mit-enterprise-architecture-guide.pdf> (accessed on 13 January 2022).
50. Derave, T. A Reference Architecture for Customizable Marketplaces. In Proceedings of the International Conference on Conceptual Modeling, Salvador, Brazil, 4–7 November 2019; Springer: Cham, Switzerland, 2019; pp. 222–229.
51. Mohammadi, N.G.; Borchert, A.; Pampus, J.; Heisel, M. A generic conceptual data model of social media services. In Proceedings of the 24th European Conference on Pattern Languages of Programs, Irsee, Germany, 3–7 July 2019; pp. 1–12.
52. Silverston, L. *The Data Model Resource Book, Vol. 1: A Library of Universal Data Models for All Enterprises*; John Wiley & Sons, Inc.: New York, NY, USA, 2001.
53. Fowler, M. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*; Addison-Wesley Professional: Westford, MA, USA, 2004.
54. OMG. Unified Modeling Language (UML) Version 2.5.1. 2017. Available online: <https://www.omg.org/spec/UML/2.5.1/About-UML/> (accessed on 13 January 2022).
55. Moody, D.L. The physics of notations: Towards a scientific basis for constructing visual notations in software engineering. *IEEE Trans. Softw. Eng.* **2009**, *35*, 756–779. [[CrossRef](#)]
56. Sonnenberg, C.; Vom Brocke, J. Evaluations in the science of the artificial—reconsidering the build-evaluate pattern in design science research. In Proceedings of the International Conference on Design Science Research in Information Systems, Las Vegas, NV, USA, 14–15 May 2012; Springer: Berlin, Germany, 2012; pp. 381–397.
57. Venable, J.; Pries-Heje, J.; Baskerville, R. FEDS: A Framework for Evaluation in Design Science Research. *Eur. J. Inf. Syst.* **2016**, *25*, 77–89. [[CrossRef](#)]
58. Hollander, M.; Wolfe, D.; Chicken, E. *Nonparametric Statistical Methods*; John Wiley & Sons, Inc.: New York, NY, USA, 2014.
59. Hartson, R.; Pyla, P. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*; Morgan Kaufmann: Amsterdam, The Netherlands, 2012.
60. Davis, F. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Q.* **1989**, *13*, 319–340. [[CrossRef](#)]
61. Shull, F.; Singer, J.; Sjöberg, D. *Guide to Advanced Empirical Software Engineering*; Springer: London, UK, 2008.
62. Lucero, A. Using Affinity Diagrams to Evaluate Interactive Prototypes. In Proceedings of the INTERACT 2015, Bamberg, Germany, 14–18 September 2015; Springer: Cham, Switzerland, 2015; pp. 231–248.
63. Creswell, J.W. *Research Design: Qualitative, Quantitative, and Mixed Method Approaches*, 4th ed.; Sage Publications: Newbury Park, CA, USA, 2014.
64. Lehnert, M.; Röglinger, M.; Seyfried, J. Prioritization of Interconnected Processes—A PageRank-based Approach. *Bus. Inf. Syst. Eng.* **2018**, *60*, 95–114. [[CrossRef](#)]
65. Orlovskiy, D.; Kopp, A. Enterprise architecture modeling support based on data extraction from business process models. In Proceedings of the CMIS-2020 (CEUR 2608), Zaporizhzhia, Ukraine, 27 April–1 May 2020.
66. Poels, G.; Ruiz, F.; García, F. An Experience in Modelling Business Process Architecture. In Proceedings of the QUATIC 2019 (CCIS 1010), Ciudad Real, Spain, 11–13 September 2019; Springer: Cham, Switzerland, 2019; pp. 119–126.
67. Moody, D.L. The Method Evaluation Model: A Theoretical Model for Validating Information Systems Design Methods. In Proceedings of the ECIS 2003, Naples, Italy, 16–21 June 2003; pp. 1327–1336.
68. Lindland, O.I.; Sindre, G.; Solvberg, A. Understanding quality in conceptual modeling. *IEEE Softw.* **1994**, *11*, 42–49. [[CrossRef](#)]
69. van der Aalst, W.M.P.; Barthelmeß, P.; Ellis, C.A.; Wainer, J. Workflow Modeling using Procllets. In *Cooperative Information Systems. CoopIS 2000 (LNCS 1901)*; Scheuermann, P., Etzion, O., Eds.; Springer: Berlin, Germany, 2000.
70. Künzle, V.; Reichert, M. PHILharmonicFlows: Towards a framework for object-aware process management. *J. Softw. Maint. Evol. Res. Pract.* **2011**, *23*, 205–244. [[CrossRef](#)]