Noname manuscript No.
(will be inserted by the editor)

A Data-Driven, Goal-Oriented Framework for Process-Focused Enterprise Re-engineering

Thai-Minh Truong · Lam-Son Lê ·
Elda Paja · Paolo Giorgini

Received: date / Accepted: date

Abstract Along with enterprise transformation, enterprise re-engineering is essential for maintaining the competitiveness of an enterprise. Enterprise re-engineering addresses (emergent) changes, re-organizing, outsourcing and re-aligning alike. Re-engineering itself has drawn traction in both academia and business. Most scholarly work in this area is confined to model-driven analysis, holistic frameworks for analyzing as-is/to-be enterprise models, and a few other conceptualization techniques. The practice of process redesign understandably takes the stage in re-engineering. Yet algorithmic techniques that insightfully point out how a process might be improved for proactively re-engineering process-intensive enterprise architecture are missing. Data science and business intelligence have brought a refreshingly new analysis to this mainstream problem by studying the operational history of a business process to facilitate most plausible changes. In this article, we investigate enterprise process redesign taking into account enterprise's high-level strategy and data warehouse. More specifically, we propose an approach to reasoning about an enterprise's strategy together with data mining rules extracted from the data warehouse of the enterprise. Our redesign algorithms suggest design-time changes to be made to its business processes, primarily by eliminating redundant tasks and re-ordering inefficiently-located tasks. We analyze the effectiveness of candidate to-be business processes with regard to business intelligence indicators. We report our work on the enterprise architecture developed for a retailer of low-cost domestic flights.

T-M. Truong, L-S. Lê (✉)
Faculty of Computer Science & Engineering, Ho Chi Minh City University of Technology (VNU-HCM), Vietnam
E-mail: lam-son.le@alumni.epfl.ch

E. Paja
Department of Computer Science, IT University of Copenhagen, Denmark

P. Giorgini
Department of Information Engineering and Computer Science, University of Trento, Italy

Keywords Enterprise Re-engineering · Goal-oriented Modeling · Business Process Management · Data Mining · Business Intelligence · Digital Enterprise

1 Introduction and research questions

The literature of information systems has enjoyed a growing interest in enterprise re-engineering for the sake of making enterprises stay competitive and dynamic in today's ever changing world (Laudon and Laudon, 2019). On the one hand, a number of definitions for this term have come out (Tribolet, 2014; Alter, 2014) most of which see enterprise re-engineering as the fundamental rethinking and radical change of business processes to gain certain improvements (Ştefănescu et al., 2008). On the other hand, enterprise re-engineering addresses changes that are constantly made to enterprises (Tribolet, 2014) some of which are emergent (Alter, 2014). Industry 4.0 has emerged as a new phase in the industrial revolution that adopts what was obtained in the third revolution and enhances it with smart and autonomous systems based on changing radically the way the entire business operates and grows. Given today's on-going industrial revolution, business processes are tightly linked to other enterprise layers and as such play a central role in enterprise engineering (Xu et al., 2018; Qu et al., 2019) and in business engineering (Winter, 2001; Österle, 2013).

Mainstream research in the realm of enterprise re-engineering is dedicated to the alignment between the enterprise's operational level (e.g., business processes, people) and its high-level management, i.e. business strategy (Iacob et al., 2012; Hinkelmann and Pasquini, 2014; Umoh and Sampaio, 2014; Salgado et al., 2014). The lower levels of an enterprise, i.e., its operational level and its computing and technology level combined, often handle a large volume of data that have increased significantly these days for the following two reasons. First, today's e-business models through which enterprises may collaborate, yield a large number of transactions. Second, computing capability and storage capacity in use today keeps expanding. To this end, enterprise-wise data increasingly play a key role in today's enterprise architecture. Data analytics would give an insight into how the enterprise in question has been operationalized of late.

In case the operationalization and the management of the enterprise diverge, the enterprise's internals need to be re-aligned to regain its competitiveness. Alterations to the existing business model of the enterprise, including process redesign, contribute to this re-alignment. In a broader scope, the evolution and transformation of enterprise architecture together have given rise to a new line of research. Enterprise transformation concerns non-trivial fundamental changes to significantly alter an organization's relationships with key constituencies (Rouse, 2005). Enterprise transformation might be enabled by the Internet of Things (Zimmermann et al., 2015), supported by a helpful management of enterprise architecture (Labusch et al., 2013, 2014), and enabled by an enhanced enterprise architecture framework (Donaldson et al., 2015). The ongoing endeavor of redesigning and restructuring today's enterprise information systems necessitates pipeline evolution (Hacks et al., 2019) and continuous re-alignment (Hinkelmann et al., 2016). A request for changes to an enterprise architecture will likely trigger a reparation process that eventually propagates changes to restore the well-formedness of the said architecture (Dam et al., 2016).

When narrowing the subject down into process re-engineering, managing changes to business processes is key to the success of a digital enterprise (Deokar and Sarnikar, 2016). The improvement of business processes could be guided using a set of pre-identified change patterns (Yousfi et al., 2016). In business management, scholarly work has confirmed that process re-engineering is evolutionary by nature (Jarvenpaa and Stoddard, 1998) and is driven by enterprise strategy (Teng et al., 1996; Kallio et al., 1999).

1.1 Research questions and design science

Despite having been investigated in many aspects, automation in enterprise re-engineering leaves a lot to be desired. Existing approaches require intensive handcraft and human expertise in the form of pattern-, product-, heuristic-based methods (Reijers, 2003; Dumas et al., 2018). To the best of our knowledge, a crisp, algorithmic suggestion to re-engineer a process-focused business is currently missing in the literature. Though business processes primarily matter in the operational level our view of process redesign is that it would be impossible without considering the big picture of the organization of an enterprise – which is heterogeneous by nature as Dietz emphasized in his work on enterprise ontology (Dietz, 2006). In the following, we elaborate our research questions. We also point out how our work fits into design science in information systems engineering.

*1.1.1 **RQ1**: How can we turn the rule-based knowledge mined thanks to enterprise data warehouse into an actionable redesign of enterprise's business processes in order to stay tuned to changing requirements?*

Business processes generate transactions as they are on the go. Enterprise data warehouses play the role of a driver for the sake of process redesign. In enterprise ontology, data warehousing is part of what is called the D-organization (Dietz, 2006). Enterprise's explicit knowledge up to the point in time considered for re-engineering is buried in transactional dataset, which give us evidence what process tasks need to be revised and rule-based ways of revising them. This would open the door to automatic process fine-tuning.

*1.1.2 **RQ2**: To what extent the enterprise's business objectives blueprint the redesign of enterprise's business processes?*

Business processes of an enterprise arguably play a central role in enterprise re-engineering as they materialize enterprise's business objectives at the operational level. In enterprise ontology, they are at the core of what is coined the B-organization (Dietz, 2006). Over time, the operational level might unknowingly/unintentionally derail from the management level and/or diverge from what is actually happening in the enterprise. One way to make the enterprise function more effectively is to re-align them with high-level business objectives, which serve as an enabler for any attempt to redesign an existing process. In addition, since a redesign attempt could yield a multi-choice output, enterprise's business objectives guide us to the most prominent choice among all possible alternatives.

*1.1.3 **RQ3**: How to algorithmically ground enterprise's business objects in a formalized process redesign?*

In enterprise ontology, business objects essentially capture the informational dimension of an enterprise, i.e., the I-organization (Dietz, 2006). Taking them into account when redesigning business processes would make redesign attempts more insightful. Technically speaking, they need to be grounded in our redesign algorithms in conjunction with other blocks including data mining rules and process tasks. According to artifact-centric business process (ACP), business objects play a central role in a business process model that serves as inputs to our algorithms.

1.1.4 Our research methodology with respect to design science

Design science research (Peffers et al., 2008) sheds light on how to systematically conduct research in the relatively broad field of in information systems. Concretely, (Hevner et al., 2004) offer guidelines of how to reason about the relevance, artifact, evaluation, rigor and contributions of scholar work. Our research artifact is a framework we propose for enterprise re-engineering with an emphasis on business processes to achieve desired goals thanks to data mining, Technically, we devise rigorous algorithms that suggest performance-driven changes to be made to an enterprise's underlying business processes, which would contribute to the overall competitiveness of the enterprise in question. Blueprints for process redesign come from the analysis of data accumulated while the enterprise's operational levels are on the go (Truong and Lê, 2016). We analyze the redesigning effectiveness of the proposed algorithms with regard to high-level indicators described at the enterprise's strategic level. More specifically, our algorithms point out how to remove redundant tasks and resequence inefficiently-located tasks of an enterprise's business process (Truong et al., 2017).

1.2 Paper structure

Section 2 presents the state of the art and related work, followed by a case study described in Section 3. In Section 4, we discuss our research methodology and offer insights into our framework for process-focused enterprise re-engineering. Our implementation is given in Section 5 (see Appendix A and Appendix B for additional details). Section 6 is dedicated to the evaluation of our framework, for which an elaboration may be found in Appendix C. Section 7 concludes the paper and outlines our plausible future work.

2 State of the art and related work

In this section, we address state of the art and relevant related work in terms of goal-based business intelligence (Subsection 2.1), business process redesign (Subsection 2.2), data mining (Subsection 2.3), process mining (Subsection 2.4), intelligence-driven process re-engineering (Subsection 2.5), and strategically driven enterprise re-alignment (Subsection 2.6).

2.1 Goal-based business intelligence

Business intelligence (BI) has been a tendency of late as a tool for proposing data-driven innovative changes to an enterprise. Most of today's research in BI is focused too much on data-oriented approaches and do little for understanding their business layer (Horkoff et al., 2014). To bridge such a gap, Horkoff et al. proposed an enterprise modeling approach named the business intelligence model that is especially focused on reasoning about situation, influences, and indicators. For reasoning and evaluating alternatives, Horkoff and Yu introduced a qualitative, interactive assessment procedure for goal models (Horkoff and Yu, 2009) based on a modeling language for the early phase in system development called i^*. To support strategic decision-making, Maté et al. (Maté et al., 2015) offered a systematic analysis for business strategy, and Barone et al. (Barone et al., 2010) proposed a goal modeling language, namely Business Intelligence Model (BIM).(Paja et al., 2016; Giorgini et al., 2002) presented the applicability of goal models through analyzing and comparing the existing goal modeling and reasoning approaches.

2.2 Business process redesign

The redesign of business processes has a vast possibility in terms of decreasing times, costs, and enhancing customer satisfaction (Vanwersch et al., 2016). Vanwersch et al. made a systematic literature review to build a general methodological framework, which specifies six key methodological decision areas (namely aim, tool, technique, actors, input, and output). There are diverse approaches to process redesign, e.g., a heuristic-based method that concentrates on various business domains (Dumas et al., 2018), several case studies using a product-based design approach to boost the performance of processes (Reijers, 2003). Automatic and semi-automatic generation of process designs and redesigns have drawn scholars' attention in the aforementioned field (Vanderfeesten et al., 2011). An alternative view point to this approach considers the changes and progression of business data as a key driver of business processes, e.g., data-driven process structures (Müller et al., 2008). In this direction, Watson Research Center is one of the pioneers proposing IBM's artifact-centric business process models (Nigam and Caswell, 2003) and data-driven process structures (Müller et al., 2008). Conceptually, in artifact-centric business process (ACP), artifacts and the process control are dually taken into account when modeling processes (Kunchala et al., 2015). Contractually, business tasks are expressed in terms of pre-conditions (i.e., conditions that must be held for the task to be invoked) and post-conditions (i.e., conditions that will be held upon the completion of the task), which should both be grounded in the same logic used for reasoning about the state of business objects (Bhattacharya et al., 2007).

2.3 Data mining

In today's digital and information age, data warehouse and enterprise's transactional data give insight into activities, habits, tendencies, characteristics and the like of individuals or organizations who took part in enterprise's business processes. Data science is becoming a powerful tool for predicting the outcome of up-coming transactions (Bordeleau et al., 2018). In the context of our work, the primary concern is to leverage data mining in order to change enterprise' business processes in ways that the enterprise of interest becomes more competitive. As in many BI scenarios, classical mining techniques such as extracting association rules, building classification models from a transactional dataset are helpful in suggesting what process changes should be made.

2.4 Process mining

Over the last few decades, process mining has received much attention from both academia and industry. It has emerged as a crucial ingredient of initiatives on process management and process improvement. In this point of view, event logs are used to discover processes, check conformance, and analyze process performance (Dumas et al., 2018). Though promising, process mining still falls short in coping with concurrency (filtering concurrent instances may yield misleading results, e.g., diagrams with many nonexistent loops) (van der Aalst, 2020). At the present time, we do not use process mining to construct the as-is processes in our framework primarily because modeled processes have a tendency to be more abstract than the discovered process models. An abstract activity can correlate to many discovered events at different levels of granularity. In the future, when the gap between discovered and hand-made models could be bridged, process mining techniques can be employed to automatically construct as-is processes for the sake of process redesign.

2.5 Intelligence-driven process re-engineering

The rationale behind BI is bringing data-driven analysis into decision making and process engineering. Most notably, Michelberger et al. devise an approach to bridging the gap between process-related information and business processes using what they call process-oriented information logistics (Michelberger et al., 2012). By the same token, Mitschang's group (Niedermann et al., 2010) proposed a (semi-)automated business process optimization platform based on actual execution data of the car loan process by using "best-practice" optimization patterns. Furthermore, Wegener and Rüping (Wegener and Rüping, 2010a) discussed the integrating data mining into business processes and its evaluation in business process re-engineering context based on Business Process Modeling Notation (BPMN) and what is called cross-industry standard process for data mining. (Rupnik and Jaklič, 2009) proposed a deployment of data mining into extant business processes using a Java library for data mining. They went further in this regard advocating the reuse of effective data mining solutions in order to minimize manual coordination and adjustment in business processes (Wegener and Rüping, 2010b). At design-time, (Koliadis and Ghose, 2009) depict the problem of handling business process change based on a sophisticated scheme for annotating process models. Shahzad and Zdravkovic analyze and optimize business processes relied on a process-oriented data warehouses and goal-oriented approach (Shahzad and Zdravkovic, 2009).

2.6 Strategically driven enterprise re-alignment

In enterprise engineering, re-alignment is a term that refers to dynamic alterations done to enterprise building blocks in order to have the enterprise's operational levels executed in accordance to the enterprise's strategy and values. Enterprise data warehousing is notable for semantically connecting groups of enterprise building blocks that were originally devised for different purposes. Wegmann proposes a methodology of reasoning about business-IT alignment with respect to enterprise strategies (Wegmann et al., 2007). Pombinho et al. attempt to bridge value-modeling concepts and traditional enterprise engineering (Pombinho et al., 2014). Schmidt addresses the strategic alignment of big data applications (Schmidt and Möhring, 2013). Fill et al. enrich enterprise views by joining enterprise models and data representation (Fill and Johannsen, 2016).

Our work in enterprise re-engineering tackles the inefficiency of business processes taking into account the big picture of an enterprise architecture. Our approach takes the rationale of BI and contributes to the discipline of business process management, in particular process improvement. Our work in enterprise re-engineering is centered around process redesign that is enabled by enterprise's strategic goals and driven by data warehousing.

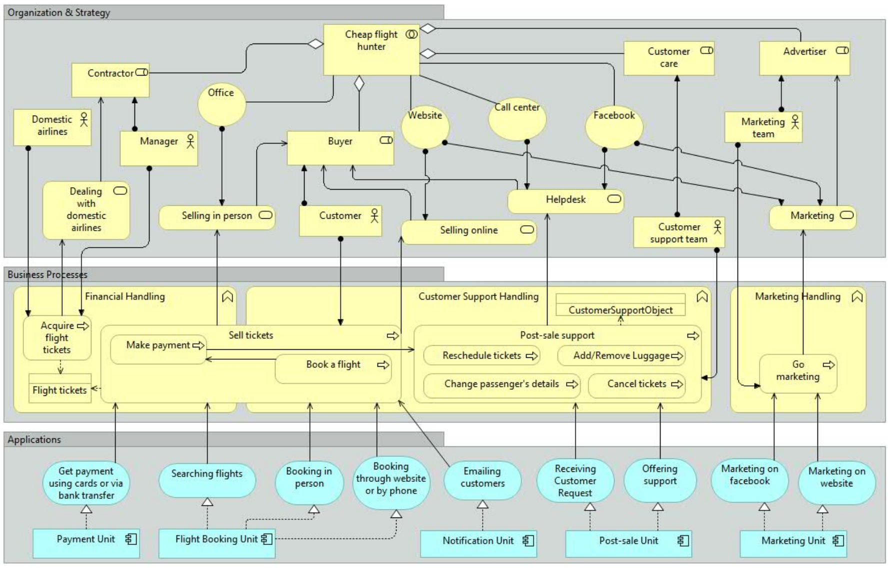


Fig. 1: The enterprise architecture of Company AD expressed in ArchiMate

3 Case study

In this section, we walk through our case study: first showing the architecture and goal models of an enterprise (Subsection 3.1), then describing its business processes (Subsection 3.2).

3.1 Case study

We describe in this subsection a real case study of a small-sized company, namely Atadi[1] (nicknamed thereafter AD), that retails low-cost domestic airfare in Vietnam sourced from a number of airlines including Vietnam Airlines[2], Jetstar Pacific Airlines[3] and VietJet Air[4]. The company's vision is to keep improving its operational levels by means of improving employee competencies, optimizing business processes and technological innovations. The company has approximately 30 employees divided into four divisions including financial, customer support, technical, and marketing departments (Figure 1).

Company AD, a typical example of successful start-ups in Vietnam, was established on March 14^{th}, 2013 and gone live in December of the same year.

1 https://www.atadi.vn
2 http://www.vietnamairlines.com
3 http://www.jetstar.com
4 http://www.vietjetair.com

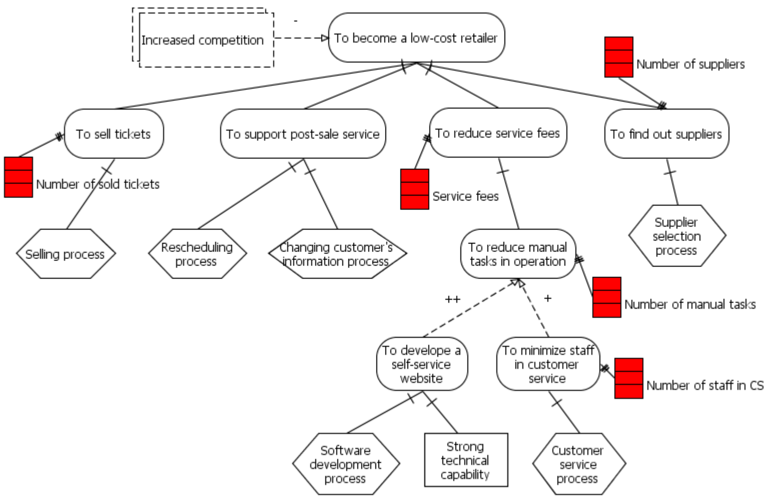


Fig. 2: Original goal model of Company AD

In running their business, they first faced a lot of challenges that are typical to most start-up businesses, leading to an initial strategy to establish a foothold in the domestic market of low-cost airfare (see Figure 2). After putting a lot of effort into operating and adapting to changing demand, they started to enjoy good growth[5] as of 2015. AD set their next goal being leader pricing in the busy market (see Figure 3). This goal is broken down into sub goals and further into leaf goals.

For the sake of depicting their business functions, in Figure 1 we present an ArchiMate[6] diagram describing the enterprise architecture of AD. Note that each business function depicted in Figure 1 comes with a fairly abstract business process that should be described in detail separately in another diagram, preferably using BPMN[7].

To stay competitive, the company's management would like to investigate if business functions `Customer Support Handling` and `Financial Handling` are aligned with their revised goals – represented in Figure 3. To do so, they attempt to redesign the processes of selling and rescheduling flight tickets separately taking into account their data warehouse. The company's first alternative is to redesign the process of selling flight tickets; the second – the rescheduling process.

[5] Until 2017, the company achieved net operating revenue of $11 millions for the same year, gross profit of more than $350 thousands. They gradually became passenger's first choice in hunting for cheap flights

[6] ArchiMate modeling language http://www.archimate.nl/

[7] The specification of BPMN is managed by the Object Management Group http://www.bpmn.org/

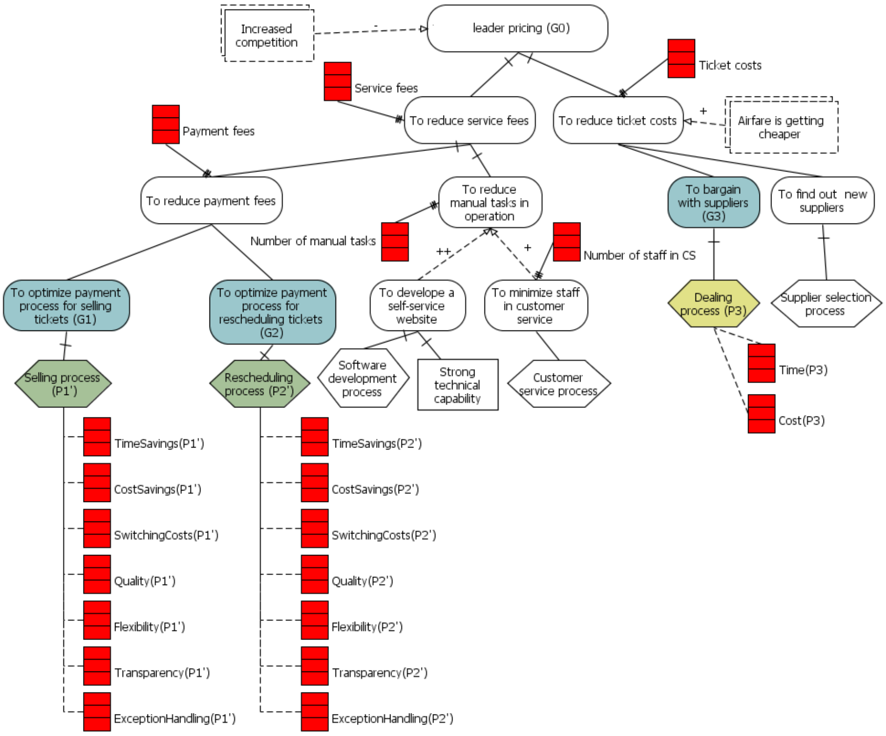


Fig. 3: Revised goal model of Company AD

3.2 Process description

Figure 4 gives a detailed representation of business process `Selling tickets` in BPMN. Company AD, its customers and the airlines providing flight tickets (under role `Provider`) are each put under a pool in this BPMN diagram. For regular air tickets, the customer searches for available flights on a departure date of her choosing until she agrees on a suitable ticket and eventually books it. The customer is notified via email listing details of her reservation and payment due date. For last-minutes tickets, the system checks her e-wallet balance and asks[8] her to top up if needed before opening a time-bound transaction for her to make a last-minutes booking. At some point in this process, Company AD would proceed in paying its provider. The customer receives a confirmation about her booking at the end of this process.

Figure 5 gives a detailed representation of business process `Reschedule tickets` in BPMN. The customer first makes a request to reschedule her air ticket. Company AD (actually its department of customer support) then checks the availability for her newly-proposed flight reschedule and contacts

[8] The process returns to regular tickets if she does not have enough e-balance.

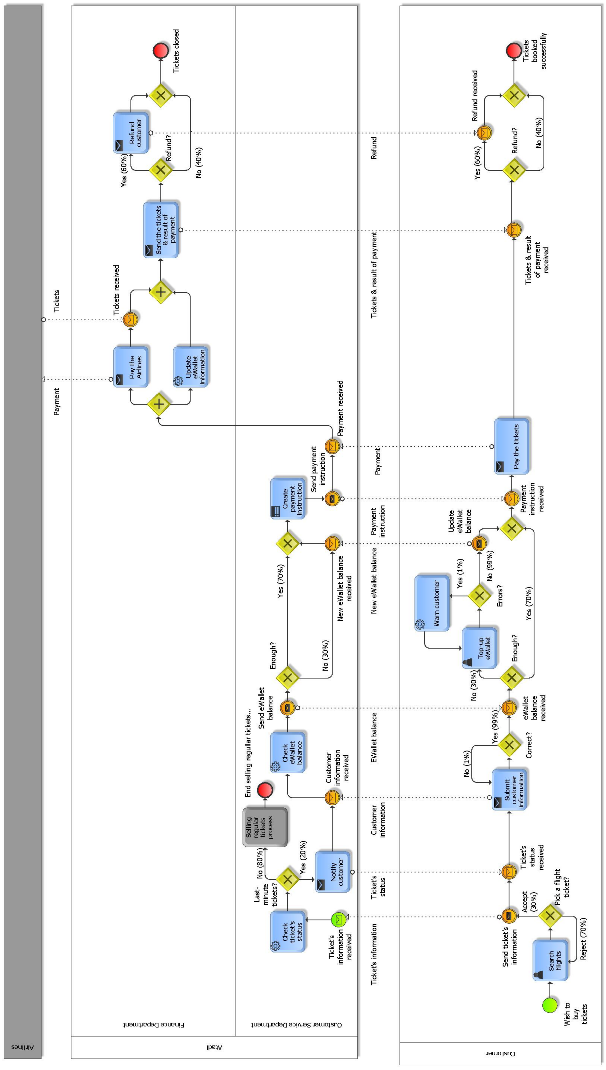


Fig. 4: [as-is] Selling tickets in Company AD

her with reschedule fees incurred if any. Otherwise, Company AD makes a list of reschedule options checking associated fees with her again. This activity is to be repeated until the customer agrees on her rescheduled flight and the associated fees. Next, Company AD asks her to pay a surcharge, usually higher[9] than the confirmed reschedule fees. Company AD then proceeds with

9 Company AD has to wait for customer's confirmation sent via email. Latency caused by any delay in making such a confirmation by the customer could result in a final charge for rescheduling her airfare that is actually higher than the reschedule fees previously confirmed by the provider. For this reason, Company AD puts a reasonable surcharge to cover this

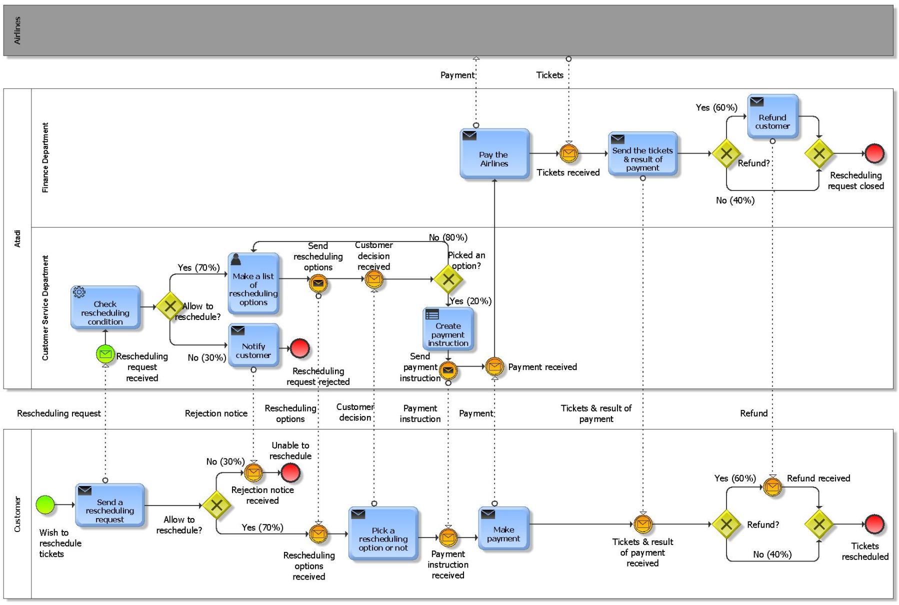


Fig. 5: [as-is] Rescheduling tickets in Company AD

payment to its provider and determines if they need to refund the customer. She receives a notification about the airfare reschedule (and possibly a refund) at the end of the process.

4 A framework for redesigning business processes

In this section, we walk through our enterprise re-engineering framework: first describing an overview of the framework itself (Subsection 4.2),then presenting formal definitions part of our framework (Subsection 4.3), and finally discussing algorithms for business process re-design (Subsection 4.4). Before going into this in-depth explanation, let us address the basis for our research methodology in line with design science in information systems research (Subsection 4.1).

4.1 Research methodology

Design science in information systems research has received increasing attention in recent years. Most notably, Hevner et al. described the performance of design science research via a concise conceptual framework and seven guidelines for conducting and evaluating good design science research

potential payment gap. The customer is entitled for a refund in case her actual reschedule fees finally go below the surcharge she paid.

(e.g., problem relevance, design as an artifact, research rigor, design evaluation, research contributions, communication of research) (Hevner et al., 2004). By the same token, Peffers et al. provided a nominal process model for doing design science research containing six steps (e.g., problem identification and motivation, design and development, evaluation), and a mental model for presenting and evaluating design science research (Peffers et al., 2008). We adopt the former because its guidelines are meant to address a whole range of aspects in information systems, and because of its relatively wide adoption (Chen et al., 2012; Peffers et al., 2008). In the following, we explain our research methodology following the guidelines we adopt.

- **Problem relevance** To stay competitive and dynamic, today's digital enterprises should be able to be changed in response to evolving needs. Majority of scholar work in process redesign is either heuristic-based or patterns-based. We target a semi-automatic, algorithmic approach taking into consideration enterprise's data, goals, processes and business rules altogether.
- **Research rigor** Our work takes its root in proven basis and rigorous theories: BIM for modeling goals, ACP for modeling enterprise's business objects and processes combined, data science for mining change-suggesting rules, and business rules for additional domain constraints. We devise (and implement) algorithms to redesign enterprise's business processes semi-automatically (see Subsection 4.4) and then measure the performance of their output (see Section 6).
- **Design as an artifact** Our work in the big picture is for enterprise re-engineering with an emphasis on business processes (see Subsection 4.2). In order to measure the performance of redesigned business processes (as-is vs to-be), we introduce a set of six indicators.
- **Design evaluation** We have tested and implemented a prototype of our redesign algorithms. We apply our framework to a real-life case study, and collect feedback from its stakeholders (see Section 5). Furthermore, we propose scales of measurement to facilitate the decision-making of what output processes to pick.
- **Research contributions** Work presented in this paper contributes to the realm of enterprise re-engineering in the big picture and business process improvement in particular. It is quite close to process mining.
- **Research communication** Academically, our research targets an audience who are familiar with enterprise engineering, business process management and business intelligence. Pragmatically, the work conveys helpful information for business practitioners, managers and decision makers alike. To facilitate the understanding this work, they should have general knowledge of goal-oriented requirements engineering, business process improvement, data mining and business rules, which are recapped in the preliminaries of this paper to make it self-contained.

4.2 Overview

Market-driven changes made to an enterprise's strategy will call for enterprise re-engineering. In our approach (see Figure 6), BIM is used to depict strategic goals and clarify the relationship between them and other enterprise's building blocks such as processes and data. Changes made to the enterprise's strategy serve as a trigger following which we might spot new/revised goals. Such a revised/new goal will likely be broken down into one or more sub-goals, each of which might further be decomposed until reaching leaf goals, i.e. goals that would not be further refined. Canonically, the goal in question will be fulfilled if all AND-decomposed[10] leaf-goals have been achieved (Gharib et al., 2018).

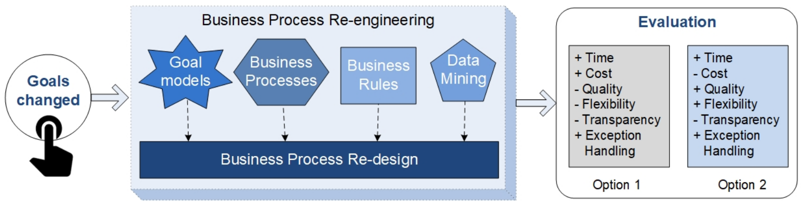


Fig. 6: The big picture of data-driven and goal-based enterprise re-engineering

For the sake of enterprise operation management, any newly added goal should be realized operationally by one or more processes of the said enterprise. To this end, we should figure out whether a process needs to be newly designed or existing ones should be redesigned to achieve this new goal. Our work is dedicated to the latter case where data mining plays a central role in redesigning business processes, possibly resulting in multiple alternatives of business processes in question. Last but not least, we assess these alternatives using a set of indicators, each of which comes with a priority level.

Additionally, the workings of the business process re-engineering block in Figure 6 are clarified by Figure 7 to describe step-by-step our approach to business process redesign. In this structure, data warehousing and enterprise's requirements both serve as inputs to our framework. Data warehousing involves pre-processing the operational data generated by business processes of the enterprise to get ready for data mining, leading to classification models (Han et al., 2011). Requirements described in the management levels of the enterprise enable us to specify non-key attributes in the data warehouse to be used as classifiers when it comes to data mining. Each resulting classification model consists of classification rules that will be assessed and chosen jointly by business analysts and data mining experts, with which they will associate several business rules refined from the enterprise's goals.

[10] Goals might logically be refined into sub-goals, thereby their satisfaction relies on that of their sub-goals. A goal might be achieved in several ways if it or its sub-goals are OR-refined (Horkoff et al., 2014).

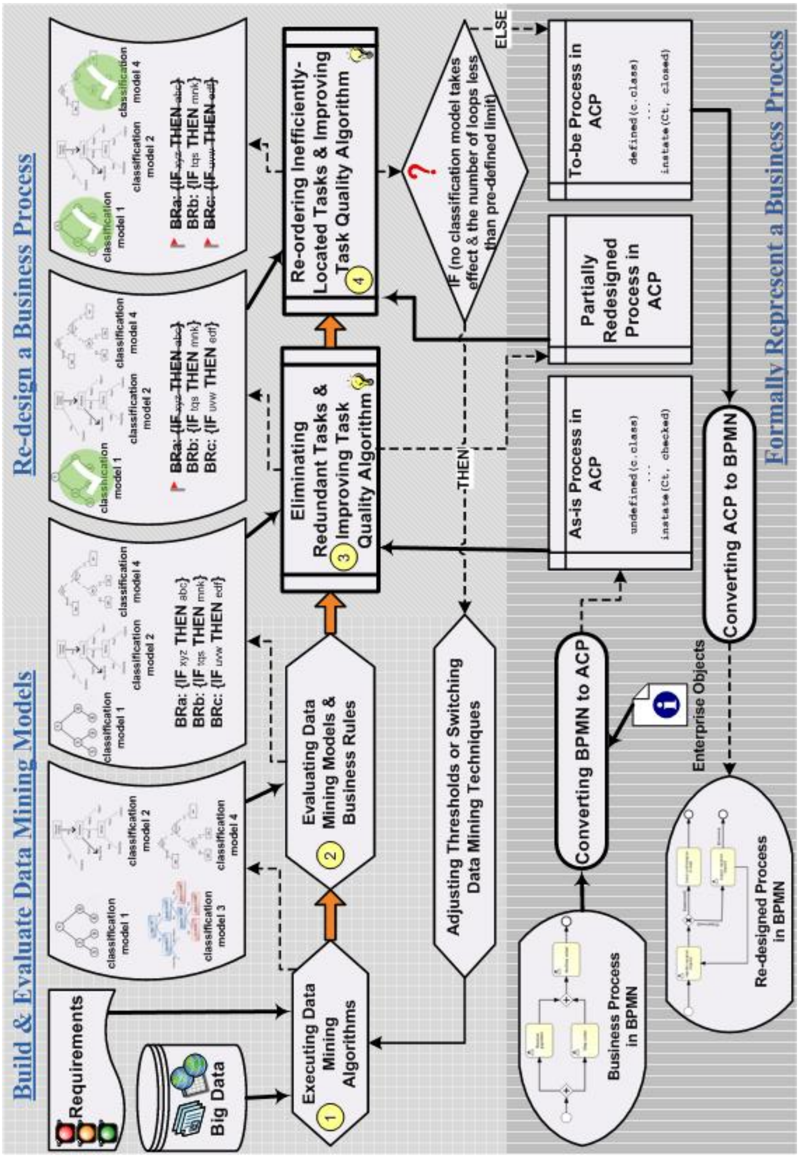


Fig. 7: Redesigning a business process in enterprise architecture taking requirements, data warehouse and business objects as inputs.

The next stage of our approach involves redesigning a specific business process of the enterprise in question. Specifically, the classification models generated, their associated business rules and the process represented in ACP are fed into the elimination step (Algorithm 1), which yields a partially redesigned process (in ACP again) where redundant tasks are either lessened or totally gotten rid of. Classification models that did not take effect in the elimination step will be the input of the next step – Algorithm 2. The output of this step is a final version in ACP that will be converted back to BPMN. In the end, we may have some classification models and business rules that went ineffective, i.e. they did not take effect in any step of our redesign framework.

Our choice of ACP in our re-engineering framework is justified by the fact that representing business objects in tandem with process tasks. It provides flexibility in performing processes by stating constraints as rules rather than sequence correlations (Voorberg et al., 2019). This approach is alternative to a common technique called process-centric modeling, which concentrates on activities, their sequence and just skims through data-aspects. Unlike the process-centric counterpart, the artifact-centric paradigm aims at business objects, describing crucial physical or conceptual entities related to the business (Sousa et al., 2020).

The underlying rationale of artifact-centric business process modeling is to have a coherent representation of business processes leveraging conceptual modeling (Bhattacharya et al., 2007). As such, we formulate formal artifact-centric representation of the business process to be redesigned based on: (a) pre-conditions and post-conditions declared in the business process, (b) conceptually-represented states and attributes of business objects that matter. Note that classification rules coming out of the classification models, the artifact-centric representation of business processes and necessary business rules, are all grounded in the same representational language. All the blocks depicted in Figure 7 are visually placed into the following three diagrammatic areas: (i) building and evaluating classification models – visualized using grid pattern; (ii) redesigning business processes – diagonal pattern; (ii) formal representation of business processes – vertical pattern.

4.3 Formal definitions

In this section, we put business tasks (in terms of pre- and post-conditions) and classification rules (in terms of conditions and consequences) in a same logical system to help them understand each other. They are broken down into literals that are considered to be compared together (e.g., including or subsuming) or provide the union or intersection of them.

Let us explain our pick on the logical system to reason about. We initially considered propositional logic and then switched to first-order logic (FOL, also called predicate logic) to enjoy its existential and universal quantifiers (Ben-Ari, 2012). Coping with business rules in our process re-design necessitates second-order (or even higher-order) structures, explaining our adoption of higher-order logic (HOL) in devising re-design algorithms (Benthem et al., 2001).

In ACP models, an artifact schema is defined as $\mathcal{M} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_n\}$ (Yongchareon and Liu, 2010), where $\mathcal{C}_i \in \mathcal{M}$ $(1 \leq i \leq n)$ is an artifact class. An artifact class $\mathcal{C}$ includes two ingredients: a set of attributes $\mathcal{A} \subseteq \{a_1, a_2, ..., a_x\}$,

$a_i \in \mathcal{A}$ $(1 \leq i \leq x)$ (we used $\subseteq$ instead of $=$ because $\mathcal{A}$ might be an empty set) and a set of states $\mathcal{S} = \{s_0, s_1, ..., s_y\}$, $s_i \in \mathcal{S}$ $(0 \leq i \leq y)$, where s_0 denotes the initial state.

A business process $\mathcal{P}$ involves some scenario labels, denoted as $\mathcal{P} = \{\mathcal{L}_1, \mathcal{L}_2, ..., \mathcal{L}_m\}$. Each scenario label $\mathcal{L}_i$ contains a precise list of business tasks that define a path leading from the start point to the point being considered (Hinge et al., 2009), denoted as $\mathcal{L}_i = \{t_1, t_2, ..., t_\ell\}$.

We adopt the concept of pre-condition and post-condition (Yongchareon and Liu, 2010) in formally representing business tasks. From now on, we define a business task t as a 4-tuple of the following elements: *pre-condition* (λ), *post-condition* (β), *service* (ν) and *set of involved artifacts* (σ)

$$t < \lambda, \beta, \nu, \sigma >,$$

where λ and β are pre-condition and post-condition of the task in question, respectively; ν is a service to be performed; σ is a set of artifact classes involved in task t, denoted as $\sigma = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_y\}$, $\mathcal{C}_i \in \mathcal{M}$ $(1 \leq i \leq y)$.

However, we argue that the pre-condition and post-condition of business tasks could be described formally in terms of attribute statements of the involved artifacts. We follow the rationale of design by contract (Meyer, 1988) according to which state machine could be ignored, as opposed to what was proposed by the ACP (Yongchareon and Liu, 2010). Based on that, we modify the definitions of pre-condition and post-condition as in the following.

Definition 1 (pre-condition, post-condition)

Both the *pre-condition* (denoted as λ) and *post-condition* (denoted as β) are defined as *logical expressions* including the following *attribute propositions* that are connected only by AND logical connective ($\wedge$) for the sake of simplicity.

- an attribute has been defined – denoted as $defined(\mathcal{C}_j.a_k)$
- an attribute is undefined – denoted as $\neg defined(\mathcal{C}_j.a_k)$
- an attribute has been re-defined, i.e., updated/changed – denoted as $redefined(\mathcal{C}_j.a_k)$
- a scalar comparison operator – denoted as $f(\mathcal{C}_j.a_k, ..., \mathcal{C}_p.a_q) \in \begin{cases} (-\infty, value], \\ [lower, upper], \\ [value, +\infty); \end{cases}$

where $j, ..., p \in \{1, ..., |\sigma|\}$; $k \in \{1, ..., |\mathcal{C}_j.\mathcal{A}|\}$; $q \in \{1, ..., |\mathcal{C}_p.\mathcal{A}|\}$; $value, lower, upper \in \mathbb{R}$ which is the set of real numbers; $\mathcal{C}_j, ..., \mathcal{C}_p \in \sigma$ are artifact classes; $\mathcal{C}_j.a_k \in \mathcal{C}_j.\mathcal{A}$ is an attribute of class $\mathcal{C}_j$; $\mathcal{C}_p.a_q \in \mathcal{C}_p.\mathcal{A}$ is an attribute of class $\mathcal{C}_p$; $\mathcal{C}_j.a_k \neq \mathcal{C}_p.a_q$.

Example 1 In the rescheduling process (Figure 5), the post-condition of task `Refund customer` (denoted as $t_{RefundCustomer}.\beta$) is described as follows:

$t_{RefundCustomer}.\beta = defined(Booking.bookingID) \wedge defined(Booking.customerID) \wedge defined(Booking.flightID) \wedge defined(Booking.staffID) \wedge defined(Booking.ticketType) \wedge defined(Booking.serviceClass) \wedge defined(Booking.addOns) \wedge defined(Booking.seatNumber) \wedge defined(Booking.bookingType) \wedge defined(Booking.bookingTime) \wedge defined(Booking.bookingFare) \wedge defined(Booking.paymentType) \wedge defined(Booking.paymentFare) \wedge defined(Booking.paymentTimeFromCustomer) \wedge defined(Booking.paymentTimeToProvider) \wedge defined(Booking.confirmationTimeFromProvider) \wedge defined(Booking.refundsToCustomer) \wedge Booking.refundsToCustomer \in (0, +\infty) \wedge defined(Booking.timeToRefund) \wedge \neg defined(Booking.timeToCheckIn) \wedge \neg defined(Booking.timeToCancel) \wedge \neg defined(Booking.timeToReschedule) \wedge \neg defined(Booking.cancelFees) \wedge defined(Booking.timeToSendTickets).$

A classification model includes a set of classification rules $\mathcal{CR} = \{cr_1, cr_2, ..., cr_n\}$. Each classification model $\mathcal{CR}$ is optionally associated with a set of related business rules $\mathcal{BR} = \{br_1, br_2, ..., br_m\}$. Both classification rules and business rules are given in the form of IF $\mathcal{C}ond$ THEN $\mathcal{C}onsqnc$, where $\mathcal{C}ond$ and $\mathcal{C}onsqnc$ are the condition and consequence of the represented rules, respectively. From now on, we denote the classification rule *cr* as the formula $cr < \mathcal{C}ond, \mathcal{C}onsqnc >$ and the business rule *br* as the similar form $br < \mathcal{C}ond, \mathcal{C}onsqnc >$. Similarly, the *condition* and *consequence* of business rules have the same formulas with the *pre-condition* and *post-condition* of business tasks, respectively. However, the *condition* and *consequence* of classification rules only contain scalar comparison operators.

Definition 2 (literals)
The *pre-condition, post-condition, condition* and *consequence*, each of which is presented by $\mathcal{X}$, are all broken down into a set of literals (denoted as $\mathcal{L}iterals(\mathcal{X}) = \{literal_1^{\mathcal{X}}, literal_2^{\mathcal{X}}, ..., literal_n^{\mathcal{X}}\}$), where, each $literal_i^{\mathcal{X}}$ has two ingredients including the variable expression ($\Lambda_i^{\mathcal{X}}$) and the range of this expression $range(\Lambda_i^{\mathcal{X}})$ – together denoted as $literal_i^{\mathcal{X}} < \Lambda_i^{\mathcal{X}}, range(\Lambda_i^{\mathcal{X}}) >$.

According to Definition 1, the range of the variable expression, denoted as $range(\Lambda_i^{\mathcal{X}})$, is given in the following form.

$$range(\Lambda_i^{\mathcal{X}}) = \begin{cases} dom(\mathcal{C}_j.a_k), & \text{if } \Lambda_i^{\mathcal{X}} = \mathcal{C}_j.a_k \text{ and } literal_i^{\mathcal{X}} = defined/redefined(\mathcal{C}_j.a_k) \\ \emptyset, & \text{if } \Lambda_i^{\mathcal{X}} = \mathcal{C}_j.a_k \text{ and } literal_i^{\mathcal{X}} = \neg defined(\mathcal{C}_j.a_k) \\ \begin{cases} (-\infty, value], \\ [lower, upper], \\ [value, +\infty); \end{cases} & \text{if } \Lambda_i^{\mathcal{X}} = f(\mathcal{C}_j.a_k, ..., \mathcal{C}_p.a_q) \end{cases}$$

where $\mathcal{C}_j \in \sigma$ is an artifact class; $j \in \{1, ..., |\sigma|\}$; $\mathcal{C}_j.a_k \in \mathcal{C}_j.\mathcal{A}$ is an attribute of class $\mathcal{C}_j$; $k \in \{1, ..., |\mathcal{C}_j.\mathcal{A}|\}$; $value, lower, upper \in \mathbb{R}$; $dom(\mathcal{C}_j.a_k)$ is a data domain of attribute a_k of class $\mathcal{C}_j$. Note that in some cases, *lower* and *upper* might be the same.

Example 2 In the rescheduling process (Figure 5), a set of literals of the post-condition of task `Refund customer`, denoted as $\mathcal{L}iterals(t_{RefundCustomer}.\beta)$, is broken down into the following literals:

$\mathcal{L}iterals(t_{RefundCustomer}.\beta) = \{defined(Booking.bookingID),\ defined(Booking.customerID),$ $defined(Booking.flightID),\ defined(Booking.staffID),\ defined(Booking.ticketType),$ $defined(Booking.serviceClass),\ defined(Booking.addOns),\ defined(Booking.seatNumber),$ $defined(Booking.bookingType),\ defined(Booking.bookingTime),\ defined(Booking.bookingFare),$ $defined(Booking.paymentType),\ defined(Booking.paymentTimeFromCustomer),$ $defined(Booking.paymentFare),\ defined(Booking.confirmationTimeFromProvider),$ $defined(Booking.paymentTimeToProvider),\ defined(Booking.refundsToCustomer),$ $Booking.refundsToCustomer \in (0,+\infty),\ defined(Booking.timeToRefund),$ $\neg defined(Booking.timeToCheckIn),\ \neg defined(Booking.timeToCancel),$ $\neg defined(Booking.timeToReschedule),\ \neg defined(Booking.cancelFees),$ $defined(Booking.timeToSendTickets)\}$,
where,
$literal_6^{t_{RefundCustomer}.\beta} < \Lambda = Booking.serviceClass,\ range(\Lambda) = dom(Booking.serviceClass) >$,
$literal_{18}^{t_{RefundCustomer}.\beta} < \Lambda = Booking.refundsToCustomer,\ range(\Lambda) = (0,+\infty) >$,
$literal_{24}^{t_{RefundCustomer}.\beta} < \Lambda = Booking.timeToSendTickets,\ range(\Lambda) = \emptyset >$.

Definition 3 (Operations over literals)
We state the ith element **includes** the jth element of a set of literals if both of them have the same variable expression and the range of the latter has to be contained by that of the former. Formally,

$$\forall\ i,j\ \in [1, |\mathcal{L}iterals(\mathcal{X})|],\ i \neq j,\ literal_i^{\mathcal{X}} < \Lambda_i^{\mathcal{X}}, range(\Lambda_i^{\mathcal{X}}) > \ \in \mathcal{L}iterals(\mathcal{X}),$$
$$literal_j^{\mathcal{X}} < \Lambda_j^{\mathcal{X}}, range(\Lambda_j^{\mathcal{X}}) > \ \in \mathcal{L}iterals(\mathcal{X})$$
$$\big((\Lambda_i^{\mathcal{X}} = \Lambda_j^{\mathcal{X}}) \wedge (range(\Lambda_i^{\mathcal{X}}) \supseteq range(\Lambda_j^{\mathcal{X}}))\big) \Longrightarrow literal_i^{\mathcal{X}} \succcurlyeq literal_j^{\mathcal{X}}.$$

We define that $\mathcal{L}iterals(\mathcal{A})$ **subsumes** $\mathcal{L}iterals(\mathcal{B})$, formally $\mathcal{L}iterals(\mathcal{A}) \Supset \mathcal{L}iterals(\mathcal{B})$, if every literal in $\mathcal{L}iterals(\mathcal{B})$ is included by at least one literal in $\mathcal{L}iterals(\mathcal{A})$,

$$\forall\ x \in \mathcal{L}iterals(\mathcal{B})\ \exists\ y \in \mathcal{L}iterals(\mathcal{A})\ (y \succcurlyeq x) \Longrightarrow \mathcal{L}iterals(\mathcal{A}) \Supset \mathcal{L}iterals(\mathcal{B}).$$

Given the ith element of a set of literals $literal_i^{\mathcal{A}} < \Lambda_i^{\mathcal{A}}, range(\Lambda_i^{\mathcal{A}}) >$ and the jth element of a set of literals $literal_j^{\mathcal{B}} < \Lambda_j^{\mathcal{B}}, range(\Lambda_j^{\mathcal{B}}) >$, their **aggregation** is defined in the following

$$literal_i^{\mathcal{A}} \cup literal_j^{\mathcal{B}} = \begin{cases} < \Lambda, range(\Lambda_i^{\mathcal{A}}) \cup range(\Lambda_j^{\mathcal{B}}) >, & \text{if } \Lambda_i^{\mathcal{A}} = \Lambda_j^{\mathcal{B}} = \Lambda \\ \{< \Lambda_i^{\mathcal{A}}, range(\Lambda_i^{\mathcal{A}}) >, < \Lambda_j^{\mathcal{B}}, range(\Lambda_j^{\mathcal{B}}) >\}, & \text{if } \Lambda_i^{\mathcal{A}} \neq \Lambda_j^{\mathcal{B}}. \end{cases}$$

Given the ith element of a set of literals $literal_i^{\mathcal{A}} < \Lambda_i^{\mathcal{A}}, range(\Lambda_i^{\mathcal{A}}) >$ and the jth element of a set of literals $literal_j^{\mathcal{B}} < \Lambda_j^{\mathcal{B}}, range(\Lambda_j^{\mathcal{B}}) >$, their **intersection** is defined below

$$literal_i^{\mathcal{A}} \cap literal_j^{\mathcal{B}} = \begin{cases} < \Lambda, range(\Lambda_i^{\mathcal{A}}) \cap range(\Lambda_j^{\mathcal{B}}) >, & \text{if } \Lambda_i^{\mathcal{A}} = \Lambda_j^{\mathcal{B}} = \Lambda \\ \emptyset, & \text{if } \Lambda_i^{\mathcal{A}} \neq \Lambda_j^{\mathcal{B}}. \end{cases}$$

Example 3 Given two sets of literals as follows:

$\mathcal{L}iterals(\mathcal{A}) = \{< Booking.serviceClass,\ dom(Booking.serviceClass) >, < Booking.bookingFare,\ dom(Booking.bookingFare) >\}$,
$\mathcal{L}iterals(\mathcal{B}) = \{< Booking.serviceClass,\ \{HotFare\_Plus\} >, < Booking.bookingFare,\ \{\$13.82\} >\}$,

we state that $literal_1^{\mathcal{A}} \succcurlyeq literal_1^{\mathcal{B}}$ because $\Lambda_1^{\mathcal{A}} = \Lambda_1^{\mathcal{B}} = Booking.serviceClass$ and $dom(Booking.serviceClass) \supseteq \{HotFare\_Plus\}$,
and $literal_2^{\mathcal{A}} \succcurlyeq literal_2^{\mathcal{B}}$ because $\Lambda_2^{\mathcal{A}} = \Lambda_2^{\mathcal{B}} = Booking.bookingFare$ and $dom(Booking.bookingFare) \supseteq \{\$13.82\}$.

Formally, we say that $\mathcal{L}iterals(\mathcal{A}) \Supset \mathcal{L}iterals(\mathcal{B})$. Furthermore, we also have
$literal_1^{\mathcal{A}} \cup literal_1^{\mathcal{B}} = < Booking.serviceClass,\ dom(Booking.serviceClass) >$,
$literal_1^{\mathcal{A}} \cup literal_2^{\mathcal{B}} = \{< Booking.serviceClass,\ dom(Booking.serviceClass) >, < Booking.bookingFare,\ \{\$13.82\} >\}$,
$literal_1^{\mathcal{A}} \cap literal_1^{\mathcal{B}} = < Booking.serviceClass,\ \{HotFare\_Plus\} >$,
and $literal_1^{\mathcal{A}} \cap literal_2^{\mathcal{B}} = \emptyset$.

4.4 Algorithms for business process redesign

The business tasks represented in BPMN diagrams (Figure 5 and Figure 4) are defined in ACP models, formally as a tuple $t < \lambda, \beta, \nu, \sigma >$ (e.g., the ACP model of rescheduling process (Figure 5) is depicted in Appendix B). To navigate back and forth in a scenario label, we define the denotations $prev^*(\mathcal{L}, t_i)$ and $next^*(\mathcal{L}, t_i)$ yielding the transitive closure of the preceding/succeeding tasks of t_i in scenario label $\mathcal{L}$, respectively.

4.4.1 Eliminating redundant tasks

We devise Algorithm 1 to eliminate redundant tasks of a business process with regard to a given (data mining) classification model. We offer an outline of this algorithm before we formally present it. We assume the following primitive: $\mathcal{E}liminate(t_i)$, meaning to get rid of task t_i from a scenario label denoted as $\mathcal{L}_h$ of process $\mathcal{P}$.

Step 1: We first identify a candidate task for being upgraded, into which the said classification model is integratable.
Step 2: Second, the post-condition of the upgraded task is strengthened by merging the consequences of all classification rules being integrated.
Step 3 a): Third, we figure out all redundant tasks that could to be eliminated if their post-condition is entailed by that of the upgraded task.
Step 3 b): We shrink tasks that now redundantly deliver the integrated classification model. Effectively, let's take out the classification rules of the integrated classification model from their post-condition.

4.4.2 Re-ordering inefficiently-located tasks

We devise Algorithm 2 to first pick a process task for being upgraded, into which a classification model might be integrated. The algorithm then scans for tasks that could be re-located backward in the sense that given the

Algorithm 1: Eliminating redundant tasks

Input: a classification model $\mathcal{CM}odel = \mathcal{CR} = \{cr_1, cr_2, ..., cr_n | cr < \mathcal{C}ond, \mathcal{C}onsqnc >\}$; a set of tasks in a specific scenario label $\mathcal{L}_h = \{t_1, t_2, ..., t_q \mid t < \lambda, \beta, \nu, \sigma >\}$; each task in $\mathcal{L}_h$ might have a set of collaborated business rules $\mathcal{BR}(t_i) = \{br_1^{t_i}, br_2^{t_i}, ..., br_m^{t_i} | br < \mathcal{C}ond, \mathcal{C}onsqnc >\}$

Output: $t_{integratedDM}$ a task to be upgraded; a redesigned version of process $\mathcal{P}$

Data: business process $\mathcal{P}$ is a set of scenario labels $\mathcal{P} = \{\mathcal{L}_1, \mathcal{L}_2, ..., \mathcal{L}_r\}$

Let $\mathcal{T}_{list} \leftarrow getTasks(\mathcal{L}_h)$; /* gets all tasks of $\mathcal{L}_h$ from $\mathcal{P}$. We assume that all tasks in scenario label $\mathcal{L}_h$ are ordered following their execution order in scenario label $\mathcal{L}_h$ of process $\mathcal{P}$ */

Let $t_{integratedDM} \leftarrow \emptyset$;

Let $\mathcal{COND} = \bigcup_{t=1}^{n} \mathcal{L}iterals(cr_t.\mathcal{C}ond)$;

Let $\mathcal{CONSQNC} = \bigcup_{t=1}^{n} \mathcal{L}iterals(cr_t.\mathcal{C}onsqnc)$;

foreach task $t_i \in \mathcal{T}_{list}$ **do**
- **if** $\mathcal{L}iterals(t_i.\beta) \cup \left(\bigcup_{j=1}^{m} \mathcal{L}iterals(br_j^{t_i}.\mathcal{C}ond) \right) \cup \left(\bigcup_{j=1}^{m} \mathcal{L}iterals(br_j^{t_i}.\mathcal{C}onsqnc) \right) \supseteq \mathcal{COND}$ **then**
  - $t_{integratedDM} \leftarrow t_i$;
  - **break**;
- **end**

end

if $t_{integratedDM} \neq \emptyset$ **then**
- $\mathcal{L}iterals(t_{integratedDM}.\beta) \leftarrow \mathcal{L}iterals(t_{integratedDM}.\beta) \cup \mathcal{CONSQNC}$;

end

foreach task $t_i \in next^*(\mathcal{T}_{list}, t_{integratedDM})$ **do**
- **if** $\mathcal{L}iterals(t_{integratedDM}.\beta) \supseteq \mathcal{L}iterals(t_i.\beta)$ **then**
  - $\mathcal{E}liminate(t_i)$;
  - $\mathcal{T}_{list} \leftarrow \mathcal{T}_{list} \setminus \{t_i\}$;
- **else**
  - **if** $\mathcal{L}iterals(t_i.\beta) \supseteq \mathcal{CONSQNC}$ **then**
    - **if** $\mathcal{L}iterals(t_i.\lambda) \supseteq \mathcal{CONSQNC}$ **then**
    - **else**
      - $\mathcal{C}$hanges the service ν of task t_i;
    - **end**
  - **else**
    - $\mathcal{L}iterals(t_i.\lambda) \leftarrow \mathcal{L}iterals(t_i.\lambda) \cup \mathcal{CONSQNC}$;
    - $\mathcal{L}iterals(t_i.\beta) \leftarrow \mathcal{L}iterals(t_i.\beta) \cup \mathcal{CONSQNC}$;
  - **end**
- **end**

end

classification model being integrated into the upgraded task, the earlier a task being done – the better. This algorithm comes with the following primitive: $\mathcal{R}eorder\mathcal{A}fter(\mathcal{L}_h, t_{integratedDM}, t_i)$, meaning to move task t_i backward to the position immediately succeeding task $t_{integratedDM}$ in a scenario label denoted as $\mathcal{L}_h$ of process $\mathcal{P}$.

Algorithm 2: Re-ordering inefficiently-located tasks

Input: a classification model $\mathcal{CM}odel = \mathcal{CR} = \{cr_1, cr_2, ..., cr_n | cr < \mathcal{C}ond, \mathcal{C}onsqnc >\}$; a set of tasks in a specific scenario label $\mathcal{L}_h = \{t_1, t_2, ..., t_q \mid t < \lambda, \beta, \nu, \sigma >\}$; each task in $\mathcal{L}_h$ might have a set of collaborated business rules $\mathcal{BR}(t_i) = \{br_1^{t_i}, br_2^{t_i}, ..., br_m^{t_i} | br < \mathcal{C}ond, \mathcal{C}onsqnc >\}$

Output: $t_{integratedDM}$ a task to be upgraded; a redesigned version of Process $\mathcal{P}$

Data: Business process $\mathcal{P}$ is a set of scenario labels $\mathcal{P} = \{\mathcal{L}_1, \mathcal{L}_2, ..., \mathcal{L}_r\}$

Let $\mathcal{T}_{list} \leftarrow getTasks(\mathcal{L}_h)$; /* gets all tasks of $\mathcal{L}_h$ from $\mathcal{P}$. We assume that all tasks in Scenario label $\mathcal{L}_h$ are ordered following their execution order in scenario label $\mathcal{L}_h$ of process $\mathcal{P}$ */
Let $t_{integratedDM} \leftarrow \emptyset$;
Let $\mathcal{COND} = \bigcup_{t=1}^{n} \mathcal{L}iterals(cr_t.\mathcal{C}ond)$;
Let $\mathcal{CONSQNC} = \bigcup_{t=1}^{n} \mathcal{L}iterals(cr_t.\mathcal{C}onsqnc)$;
foreach task $t_i \in \mathcal{T}_{list}$ **do**
    if $\mathcal{L}iterals(t_i.\beta) \cup \left(\bigcup_{j=1}^{m} \mathcal{L}iterals(br_j^{t_i}.\mathcal{C}ond)\right) \cup \left(\bigcup_{j=1}^{m} \mathcal{L}iterals(br_j^{t_i}.\mathcal{C}onsqnc)\right) \Supset \mathcal{COND}$ **then**
        $t_{integratedDM} \leftarrow t_i$;
        break;
    end
end
if $t_{integratedDM} \neq \emptyset$ **then**
    $\mathcal{L}iterals(t_{integratedDM}.\beta) \leftarrow \mathcal{L}iterals(t_{integratedDM}.\beta) \cup \mathcal{CONSQNC}$;
end
foreach task $t_i \in next^*(\mathcal{T}_{list}, t_{integratedDM})$ **do**
    if $\mathcal{L}iterals(t_i.\lambda) \Subset \mathcal{L}iterals(t_{integratedDM}.\beta)$ **then**
        $\mathcal{R}eorder\mathcal{A}fter(\mathcal{L}_h,\ t_{integratedDM},\ t_i)$;
        break;
    else
        if $\mathcal{L}iterals(t_i.\beta) \Supset \mathcal{CONSQNC}$ **then**
        else
            $\mathcal{L}iterals(t_i.\lambda) \leftarrow \mathcal{L}iterals(t_i.\lambda) \cup \mathcal{CONSQNC}$;
            $\mathcal{L}iterals(t_i.\beta) \leftarrow \mathcal{L}iterals(t_i.\beta) \cup \mathcal{CONSQNC}$;
        end
    end
end

Step 1: In order to identify a task into which the said classification model should be integrated, we logically compare the aggregation of all conditions specified for the classification rules in the classification model to the post-condition of the task in question combined with associated business rules.
Step 2: We then strengthen the post-condition of the aforementioned task to reflect the integration of the given classification model.

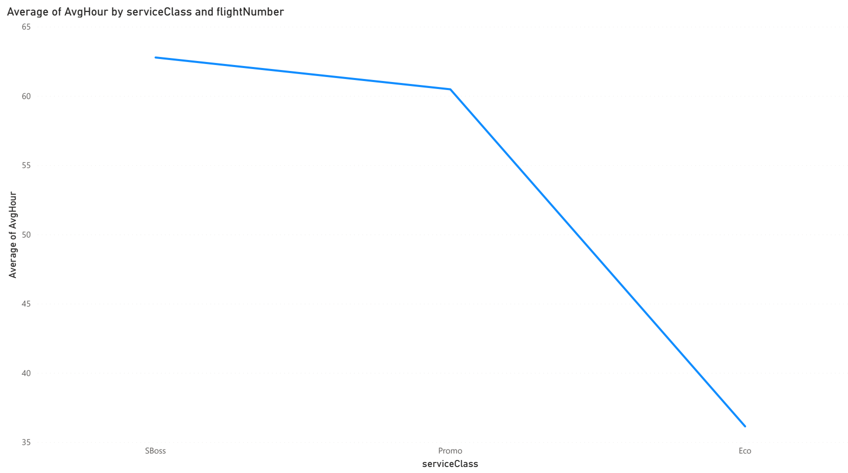
Step 3: Look for a task that ought to be done earlier in the process by matching its pre-condition against the post-condition of the upgraded task. Let's call it the candidate task.
Step 4: Move the candidate task backward to the position immediately succeeding the upgraded task if its pre-condition is subsumed by the post-condition of the upgraded task.

5 Implementation

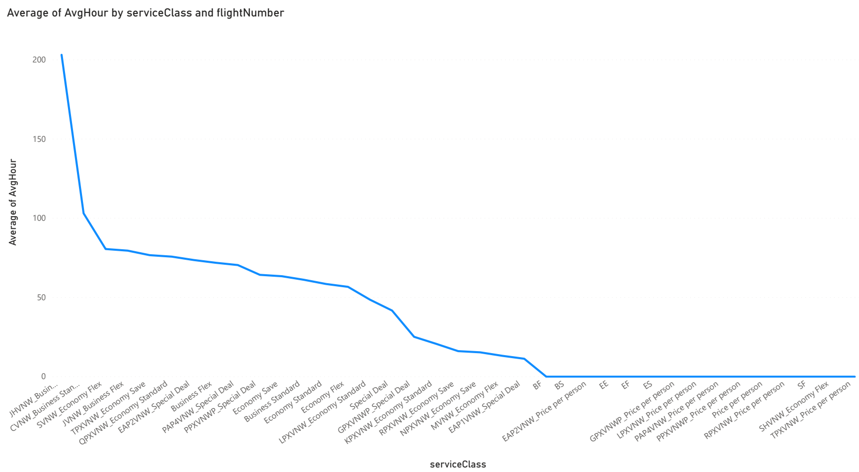
We have developed a proof-of-concept prototype that offers enterprise re-engineering to an organization by taking into account: (a) goal models representing the business strategies of the company; (b) gathered dataset (e.g., transactional data, survey data, partners' operational data); (c) as-is business processes depicted in BPMN and ACP; (d) necessary business rules. We make an assumption that data mining techniques used must generate knowledge in the of conditional expressions (e.g. rule-based classification techniques), which are able to combine with other rules (e.g., business rules, business process models in ACP) in our framework. Let us revisit the case of Company AD (presented in Section 3) where the input given to our redesign framework should be (a) goal models of Company AD (see Figure 2 and Figure 3); (b) schema and samples of the operational data provided by AD's technical department; (c) diagrammatic representation of the as-is processes (see Figure 4 and Figure 5), and their representation in ACP (e.g., see Appendix B for the rescheduling process); (d) some business rules that matter in Company AD.

In Figure 3, we need to take into account six leaf goals, some of which were satisfied (e.g., to develop a website for self-services, to minimize staff in customer service, to find out new suppliers) in the original goal model described in Figure 2, while others are brand-new ($G1$, $G2$, $G3$) meaning they do not exist in the original goal model. Existing processes related to newly-created sub-goals ($P1'$, $P2'$) in Figure 3 are supposed to be redesigned to fulfill the emerging goals. In this end, a new process ($P3$) needs to be constructed to achieve Goal $G3$. To reach Goal "To reduce payment fees", we have two potential options which are $G1$ related to "Selling process" ($P1'$) and $G2$ related to "Rescheduling process" ($P2'$). Both of them will be redesigned using our framework (see Figure 7) and assessed using our set of indicators proposed in Section 6.

After filtering the data according to the aforementioned goals, two datasets were taken as inputs to forecast the trend of airfare and the passenger's choices. First, real data of airfares from all domestic airlines in Vietnam were used to predict how airfare rate fluctuates (skyrockets, up, unchanged, down, plummets). The original data was formatted in json and its size is more than 16GB. We then converted it into the csv format to import to the data mining applications. Second, the customers' decisions were recorded through capturing exchanged emails. Its meta data (e.g., name of attributes, type of data, range of attribute values) and a part of its records were extracted from the datasets



(a) Vietjet Air



(b) Vietnam Airlines

Fig. 8: Time difference between 2 consecutive transactions that update airfare

themselves (email logs), but the rest of its records was generated using our tool according to the said meta data.

The dataset on airfares was preprocessed thanks to filling in missing values, reducing noise, resolving data inconsistencies alike. We recognized that the airfare depends on not only the flight distance but also the departure time and and the service class. For instance, airfare on national holidays and

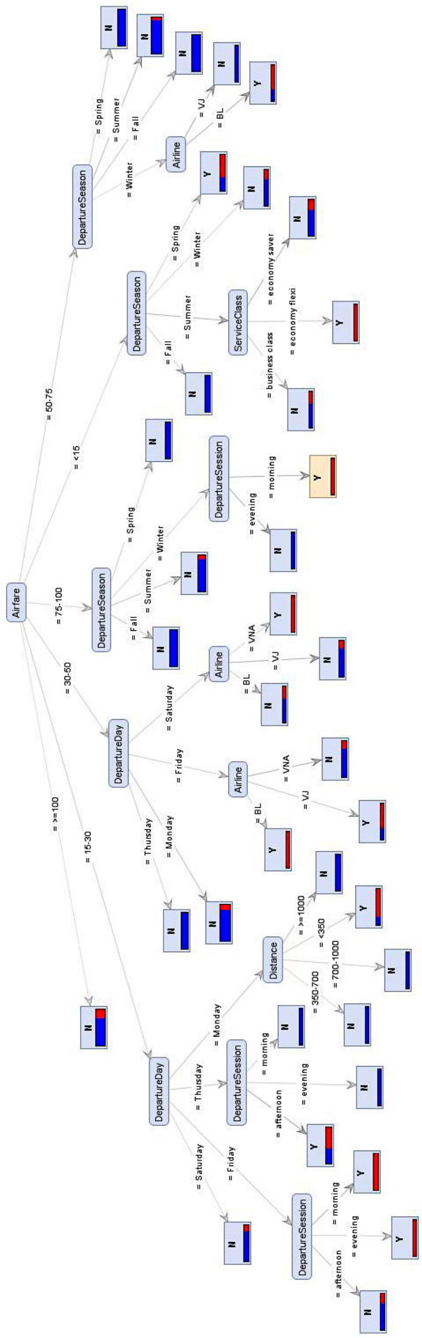


Fig. 9: Classification model $CModel_{acceptance4selling}$ explains the customer's choice over a flight ticket.

summertime are higher than that in the other seasons. Notably, the prices do not increase or decrease arbitrarily. They rather fall into some categorical values, leading to predictable airfare fluctuations.

We classify the airfares according to their routes because the fares are quite different from charter to charter. In Figure 10, we present the classification model of a flight route from Buon Me Thuot Airport to Tan Son Nhat Airport over 19 months (from October 2016 to May 2018). This model describes how airfare fluctuates (e.g., up, down, skyrockets, plummets, unchanged). Figure 8(a) confirms that VietJet Air is the airline having the most frequently updated fare (from nearly 36 hours to more than 63 hours). Vietnam Airlines has the largest amount of service classes (see Figure 8(b)).

Roughly speaking, the higher the ticket class is, the longer the average time between two consecutive updating transactions. As indicated in Figure 10, Vietnam Airlines when offering high-end domestic air tickets does not frequently update their fare. Whereas, Jetstar Pacific Airlines and VietJet Air – popular low-cost airlines in Vietnam, do so on a regular basis. In Jetstar Pacific Airlines ("provider" = "BL"), how airfare fluctuates depends on service class. For VietJet Air as yet another airline ("provider" = "VJ"), the price is the most influential attribute while the service class does not matter. For further details about the classification rules can be found in Appendix A. In the following, we describe some notable rules.

- In Jetstar Pacific Airlines, for tickets falling into the plus class, if their fare is no more than $18.27, it tends to rise. Unless, within the same service class but the prices are higher than $18.27, the airfares tend to decrease.
- In Jetstar Pacific Airlines, if the tickets fall into a class called starter, their price almost always rises except when the price equals $8.36 where it will plummet.
- In VietJet Air, if the airfares fall into these price intervals [$0, $1.29], [$4.4, $4.84], [$76.44, $83.11], they tend to rise. They will be unchanged if they fall into [$17.33, $19.07], [$25.78, $33.33]. Whereas, if they fall into [$41.07, $49.87], [$64.53, $74.31], [$84.09, $91.42], they tend to decrease.
- In VietJet Air, for tickets that cost $13.29 or $14.62 and depart in springtime or the departure is less than 10 days away from now, the airfares will skyrocket. Otherwise, these prices tend to decrease.

The classification model $CModel_{acceptance4rescheduling}$ depicted in Figure 11 represents main factors leading to the customer's choice over a flight. The root node of this decision tree is about fees incurred when rescheduling an air ticket. The smaller a fee is, the more likely it will be accepted. Another significant node tells time difference between the flight being searched for and the flight suggested actually. If the time difference is no more than 5 hours and the rescheduled ticket was booked before noon, the customer will likely accept even though her rescheduling fees is pretty high ([$75-$100]).

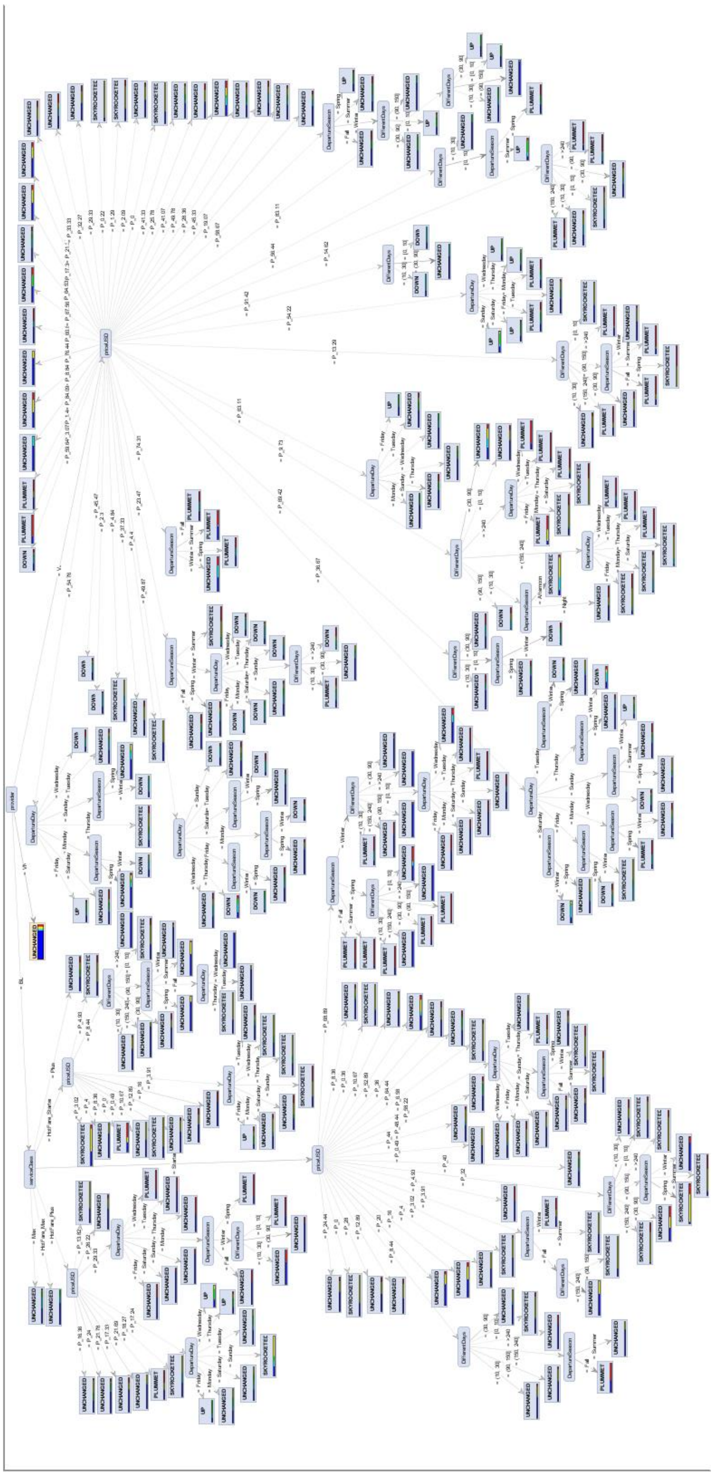


Fig. 10: Classification model $CModel_{fare}$ tells how airfares from Buon Me Thuot airport to Tan Son Nhat airport fluctuate. The full list of classification rules extracted from this tree are presented in Appendix A

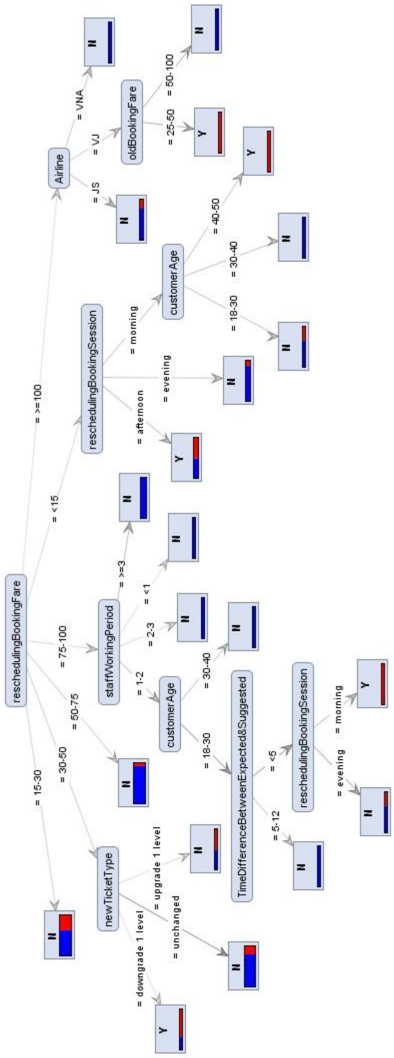


Fig. 11: Classification model $CModel_{acceptance4rescheduling}$ explains the customer's decision on suggested rescheduling options.

The classification model $CModel_{acceptance4selling}$ shown in Figure 9 represents main factors leading to the customer's choice over a flight ticket. In this classification tree, `the airfare` plays a major role in customer's decision to pay for or not. The smaller this fees is, the more likely it is accepted. The other significant elements in the said model are `the departure day` and `the departure season`. If the departure time is weekends or holiday seasons (e.g., lunar new year, summer vacations), the customer will likely accept it.

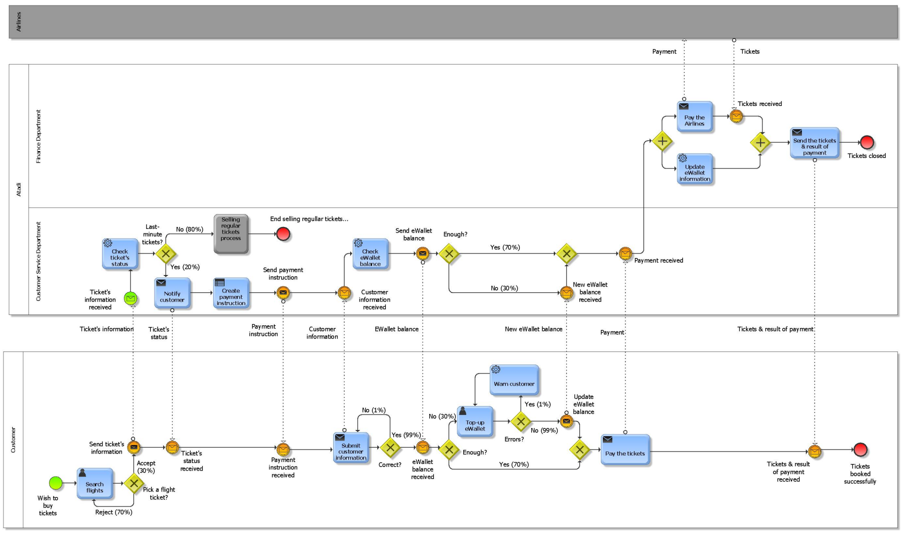


Fig. 12: [to-be] Selling last-minutes tickets in Company AD

The classification rules chosen, the business rules defined above, the as-is processes of selling tickets and rescheduling tickets presented in ACP models (see Appendix B) together serve as the input of our redesign framework. The core of our framework are Algorithm 1 and Algorithm 2, which correspond to Step 3 and Step 4 in Figure 7. We eliminate redundant tasks and re-order inefficiently located tasks. We formalize some notable business rules as follows.

- $br1 \equiv \text{IF } currentTime - Booking.bookingTime > 6\ hours \wedge \neg defined(Booking.paymentTimeFromCustomer) \text{ THEN } instate(Booking, canceled)$,
- $br2 \equiv \text{IF } Booking.ticketStatus = \text{“}last-minute\ tickets'' \text{ THEN } Booking.paymentType = \text{“}e-wallet''$,
- $br3 \equiv \text{IF } currentTime - Flight.departureTime < 2\ hours \wedge Booking.flightID = Flight.flightID \text{ THEN } instate(Booking, closed)$.

In Table 1, we explain the mechanism for transferring from the as-is to the to-be processes using Algorithms 1 and 2. In the first ten lines of both algorithms, we try to identify a candidate task into which the classification model might be integrated. In the rest of the algorithms, we try to figure out a redundant task that needs to be eliminated, and a potential task that might be reordered in Algorithms 1 and 2, respectively.

The as-is business processes (in Figure 4 and Figure 5) are transformed into the to-be processes (in Figure 12 and Figure 13). The redesigned processes differ from the original processes primarily on: (1) tasks `Refund customer` in selling and rescheduling processes are eliminated. Instead, according to $CModel_{fare}$, the department of customer support advise their customer when her airfare may increase, decrease or stay unchanged. The customer holds her initiative in payment; (2) In selling process, task `Create`

Table 1: Walking through the transformation from an as-is process to to-be's

Algo.	**Line**	**Purpose**	**Process**	**Input**	**Output**
Algo. 1	5-10	Identify a candidate task into which the classification model might be integrated ($t_{integratedDM}$)	Selling Process	T_{list} of Selling Process, $COND$ of $CModel_{fare}$, BR related to Selling Process	$t_{integratedDM} \leftarrow$ `Create payment instruction`
			Rescheduling Process	T_{list} of Rescheduling Process, $COND$ of $CModel_{fare}$, BR related to Rescheduling Process	$t_{integratedDM} \leftarrow$ `Create payment instruction`
	11-13	Update the post-condition of $t_{integratedDM}$ by integrating the classification model	Selling Process	$t_{integratedDM}$ = `Create payment instruction`, $CONSQNC$ of $CModel_{fare}$	Task `Refund customer` eliminated
	14-18	Find out a redundant task that needs to be eliminated			
	19-23	Figure out an overlapping task that will be reduced	Rescheduling Process	$t_{integratedDM}$ = `Create payment instruction`, $CONSQNC$ of $CModel_{fare}$	Task `Refund customer` eliminated
	24-29	Update the pre-condition and post-condition of all effected tasks			
Algo. 2	5-10	Identify a candidate task into which the classification model might be integrated ($t_{integratedDM}$)	Selling Process	T_{list} of Selling Process, $COND$ of $CModel_{acceptance4selling}$, BR related to Selling Process	$t_{integratedDM} \leftarrow$ `Notify customer`
			Rescheduling Process	T_{list} of Rescheduling Process, $COND$ of $CModel_{acceptance4rescheduling}$, BR related to Rescheduling Process	$t_{integratedDM} \leftarrow$ `Make a list of rescheduling options`
	11-13	Update the post-condition of $t_{integratedDM}$ by integrating the classification model	Selling Process	$t_{integratedDM}$ = `Notify customer`, $CONSQNC$ of $CModel_{acceptance4selling}$	Task `Create payment instruction` resequenced to go after Task `Notify customer`
	14-17	Figure out a potential task that needs to be reordered			
	18-24	Update the pre-condition and post-condition of all effected tasks	Rescheduling Process	$t_{integratedDM}$ = `Make a list of rescheduling options`, $CONSQNC$ of $CModel_{acceptance4rescheduling}$	Task `Create payment instruction` resequenced to go after Task `Make a list of rescheduling options`

`payment instruction` is moved forward to be operated before checking and topping-up e-wallet, as well as in rescheduling process, task `Create payment instruction` is also moved forward so that it can be done before sending rescheduling suggestion to customers. Thanks to $CModel_{acceptance4rescheduling}$ and $CModel_{acceptance4selling}$, the department of customer support only suggests the most likely acceptable reschedule options instead of listing all available flights as in the as-is process. Each of these options is associated with appropriate payment instructions. As such, the processing time is shortened as less emails need to be exchanged. Moreover, staff needed for customer support is cut down and the risk of airfare rate fluctuation is reduced because the customer transmits her payment faster.

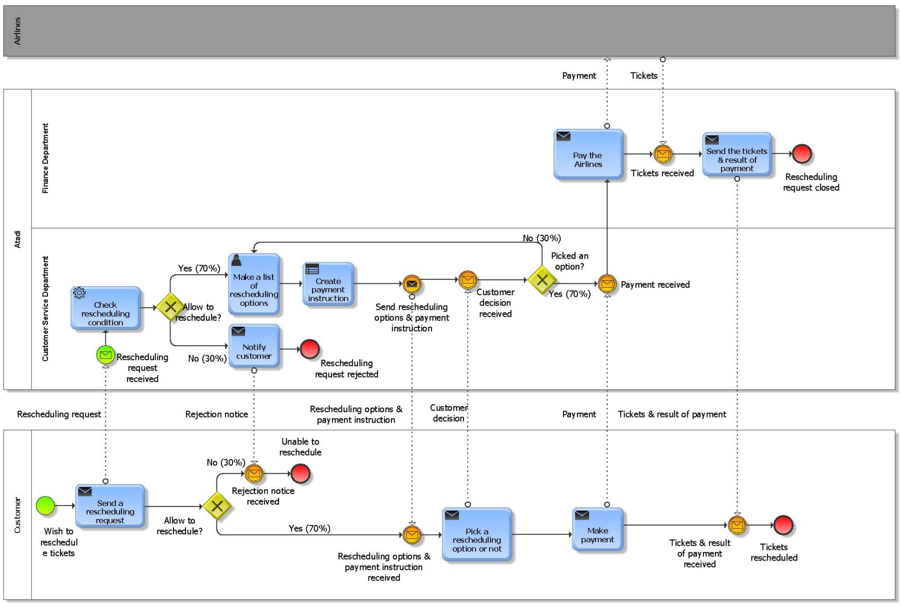


Fig. 13: [to-be] Rescheduling tickets in Company AD

Figure 12 and Figure 13 describe the to-be processes for selling flight tickets and handling the customer's reschedule requests, respectively. Both of them are evaluated on our indicators presented in Section 6. Based on that comparison, AD will take the initiative in deciding the solutions for re-engineering their enterprise to reach their target in question.

We have received positive responses from the management of Company AD on the redesigned business processes suggested by our prototype. Its chief executive officer took a quick look and stressed that the suggested re-engineering made sense and could be implemented soon to improve his company's competitiveness through saving time, cutting down operational costs and enhancing customers loyalties. One of the chief technology officers of the company was keen on up-scaling the data warehouse in order to obtain a thorough redesign suggestion. In his view, the to-be processes not only cut down processing time but also reduce the risk of airfare rate fluctuation the customer might suffer from while reviewing reschedule options. Moreover, it allows the company to improve the quality of their customer support services by suggesting highly-acceptable rescheduling options and advising the customer how to make timely payment.

6 Evaluation

After having networked with eight industry partners of ours, we recognized that the framework we propose may not be appropriate for large enterprises (e.g., state institutions, multinational corporations). It is rather suitable for medium-sized, small-sized and micro-sized companies[11]. We summarize our lessons learned regarding the suitability of our re-engineering framework in Table 2.

Table 2: Suitability of our enterprise re-engineering framework

Type of enterprises		**Lessons learned**	**Suitability**
State institutions		Their processes are bulky and hard to be changed, their dataset are unstructured, heterogeneous and sometimes manually handled.	No
Multinational corporations		Their processes are well documented, their enterprise data warehousing is satisfactorily performed. But due to cumbersome management apparatus, it is generally difficult to change their operational levels. Additionally, they are not ready to disclose information because of strict security requirements.	No
Domestic companies	Large-sized companies	Similar to that of multinational corporations.	No
Domestic companies	Medium-sized companies	Their management apparatus is compact, their processes are quite flexible and the amount of data collected is sufficient for mining. It was straightforward to go with goal-oriented modeling to capture their high-level requirements. Much of our effort was nevertheless put on describing business objects and formulating business rules to glue these ingredients together, which were key to the success of our re-engineering attempt.	Yes
Domestic companies	Small-sized companies	Similar to that of medium-sized companies.	Yes
Domestic companies	Micro-sized companies	Their processes are quite sketchy, unclear and but highly flexible. As such, they are subject to improvement. Their dataset are not huge but it is in general possible to gather additional data sources from their partners and customers.	Yes

[11] A large-sized companies typically has more than 300 employees; medium-sized 200–300 employees; small-sized 10–200 employees; micro-sized – no more than 10 employees.

Let us revisit the case of company AD (described in Section 3) once more time. In line with general recommendations for gaining essential understanding of the business processes in question before we could propose any improvement to them (Adesola and Baines, 2005; Dumas et al., 2018), we organized a total of eight face-to-face meetings over a period of three years with senior positions of company AD, for which results are elaborated in Table 3. The rightmost column in this table tells who from the side of company AD actually participated in these meetings.

Table 3: Meetings with company AD were conducted over a substantially long period of time to establish a deep engagement for the sake of enterprise re-engineering

Year	Meeting	Meeting's objectives	Participants
1	m1	Getting introduced and making a request for a proposal of enterprise re-engineering.	CEO[12], COO[13].
	m2	Brainstorming to clarify their enterprise goals and requirements.	CEO, COO.
	m3	Investigating their business domain and processes.	COO, CTO[14], a financial employee, a customer support employee.
2	m4	Asking for dataset recording their customer support practices via email log.	CTO, a technical employee.
	m5	Presenting our first attempt to apply our re-engineering framework taking into account goals, processes, email log, business objects and rules; obtaining their feedback in face-to-face mode.	CEO, CTO.
	m6	Requesting to gain access to a big dataset of airfare fluctuation provided by domestic airlines.	CTO, a technical employee.
3	m7	Showing the outcome of our second attempt to re-engineer over the dataset of airfare history; receiving their comments in an interview-based fashion.	CEO, CTO.
	m8	Wrapping things up, presenting the resulting re-engineering model systematically with guidance for further application of our framework if they wish to, and collecting their general feedback on the suitability and usability of our re-engineering framework.	CEO, CTO.

In Table 4, let us present detailed comments and feedback we received on the adoption of our re-engineering framework. In this table, the leftmost two columns list comments on the outcome of our re-engineering attempts while the rightmost column is dedicated to feedback on the adoption of our framework. In general, they enjoyed the reduction of time and cost when having their business processes redesigned, as well as a likely enhancement of customer satisfaction. They believed that re-engineering attempts were helpful and would improve their competitiveness if properly implemented. However, they raised their concern on the accuracy and confidence level of the established data mining models and remained skeptical about budget for implementing the changes proposed.

Table 4: Feedback of company AD obtained from face-to-face interviews

	On the results that came out of meeting m5	**On the outcomes that came out of meeting m7**	**On the adoption of our framework that came out of meeting m8**
Pros	Making customer's choices predictable meant enhanced customer satisfaction by improving the quality of their customer support service.	It was good that we can forecast the airfare rate fluctuation. It leads to the reduction of the process tasks that ask customers to pay a surcharge. It would increase the customers' trust and loyalty.	The changes pointed out by our re-engineering attempts were useful and implementable to improve their competitiveness through saving time, cutting down operational cost, and enhancing customer loyalty.
	Lowering the total time of communication spent on exchanging email with their customers. Payment information would be sent to customers sooner rather later.	Reducing the time and cost of the business processes related to this task.	
Cons	They worried about the accuracy and reliability of the prediction models in question.	They raised their concern on the trustworthiness of the aforementioned forecast model.	They worried about the support and confidence levels of data mining rules being applied. Inherent lack of demanded expertise.

In the next subsection, we will describe the method of assessing business processes according to process performance measurements (e.g., time, cost, quality, flexibility) to compare the as-is processes to the to-be processes.

6.1 Process performance measurement

Devil's Quadrangle (Mansar and Reijers, 2007) introduces four main criteria to evaluate a business process, namely *time* (i.e., how long will be taken to complete the process), *cost* (i.e., the necessary resources need to be used to operate the process), *quality* (i.e., the repetitions could be used to evaluate the quality of processes because the loop of rework pattern commonly happens when a task has not been finished successfully and it increases the cycle time of a task), and *flexibility* (i.e., the degree of variation that a process allows). To make these indicators measurable, we borrow Dumas's work on performance analysis of business processes (Dumas et al., 2018). In addition, the processes should be considered in many different perspectives to have a comprehensive assessment. We propose two more evaluation criteria: *transparency* (i.e., the clarity level of a process that depends on the number of explicit tasks), and *exception handling* (i.e., the degree of errors in a process that could be handled), for measuring the clearness and stability of a process respectively.

We employ a theory concerning the actual, threshold, target and worst values for measuring the above-mentioned indicators (Horkoff et al., 2014). Each indicator comes with a performance level that measures how far the actual value of the indicator is from the expectation or the worst. Note that we obtained the target, threshold and worst values by interviewing the key people in the management of Company AD. Being inspired by this work, we slightly revise their formula determining the performance level as follows to allow a negative performance level.

$$pl(current\ v.) = \begin{cases} \frac{current\ v. - threshold\ v.}{target\ v. - threshold\ v.}, & \text{if } current\ v. >= threshold\ v. \\ \\ \frac{current\ v. - threshold\ v.}{threshold\ v. - worst\ v.}, & \text{if } current\ v. < threshold\ v. \end{cases}$$

where *pl* represents the performance level of a specific indicator.

6.1.1 Time

In business process evaluation, time is one of the most common indicators that addresses a cycle time, which is a period of time taking to operate one scenario from begin to end. It includes processing time (a.k.a. theoretical cycle time) and waiting time. Processing time is the time used for doing actual work without waiting. Waiting time is the time in idle mode. In performance measurement of business processes, Dumas et al. used flow analysis technique to calculate the overall cycle time of a process based on the cycle time of each activity (Dumas et al., 2018). In the same way, this method could be used to calculate the total theoretical cycle time of a process by using the processing time of each activity. As a shorthand, the term cycle time will be used to refer to either cycle time or theoretical cycle time in the rest of this paper. In order to understand how flow analysis works, it is useful to introduce some common patterns constructing a process as below. In sequential pattern, the cycle time of sequential fragment of process is the sum of cycle time of each activity in this fragment. In a XOR-block pattern, the cycle time of a XOR-block is the weighted average of the cycle times of branches in-between XOR-split and XOR-join. In an AND-block pattern, the cycle time of an AND-block is the maximum cycle time of the longest task in this fragment. In the rework pattern, the cycle time of a repetition block is the multiplicity of the average number of times that task t is expected to be executed with its cycle time (Dumas et al., 2018).

By determining the cycle time of the as-is and to-be processes (see Appendix C.1), we calculate their *Time Savings* in Table 5.

6.1.2 Cost

Cost measurement is one of the most relevant indicators as it impacts the revenue and profit of an organization. In traditional activity-based costing

Table 5: Time savings earned when redesigning business processes

Process	$TimeSavings_{Process} = CycleTime_{As-isProcess} - CycleTime_{To-beProcess}$
Selling process	$51.2 minutes/transaction$ $(= 110.425 - 59.225)$
Rescheduling process	$41.4 minutes/transaction$ $(= 73.9 - 32.5)$

models, the activity costs of an individual product or customer are determined based on the output of each activity (e.g., amount of orders, number of receipts, amount of complaints). In time-driven activity-based costing models, the cost of a process can be calculated by estimating time needed to perform its tasks and by looking into the unit cost of supplying capacity (Kaplan and Anderson, 2003).

In the case of Company AD, we would like to measure the cost of the selling and the rescheduling processes, after being redesigned. These processes belong to customer service and the financial department having 17 members combined. Every staff member works 480 minutes/day, which is equivalent to 8 hours/day, meaning each employee contributes 10,560 minutes/month because one month has 22 working days, making the theoretical capacity of the customer service team 179,520 minutes/month. Experimentally, a practical capacity is often equal to 80% or 85% of theoretical capacity (Kaplan and Anderson, 2003). In this case study, we assume that the practical capacity accounts for 80% of theoretical capacity that is equivalent to 143,616 minutes/month.

Cost of capacity supplied includes the resources required to perform varied functions (e.g., space, salary, computers, telecommunications, furniture) and other support resources (e.g., information technology, human resources, utilities) (Kaplan and Anderson, 2003). Company AD estimated the cost of all these resources for customer service and financial departments is about \$11,363 per month lead to the unit cost is about \$0.08 per minute ($Unit\ Cost = \frac{\$11{,}363/month}{143{,}616 minutes/month} = \$0.08/minute$).

According to the aforementioned *Unit Cost*, *Time Savings* and the activity cost of tasks in selling process and rescheduling process presented in Appendix C.2, we propose a formula presented in Table 6 to calculate the *Cost Savings* of processes.

Table 6: Cost savings earned when redesigning business processes

Process	$CostSavings_{Process} = TimeSavings_{Process} * UnitCost + OtherCosts$
Selling process	$\$4.096/transaction$ $(= 51.2 * 0.08 + 0)$
Rescheduling process	$\$3.312/transaction$ $(= 41.4 * 0.08 + 0)$

The investment for switching from an as-is process to a to-be process requires a significant budget. In Table 7, we calculate the incurred switching cost.

Table 7: The switching cost of business processes

Process	$SwitchingCost = ImplementationCost + TrainingCost + DevicesCost + OtherCosts$
Selling process	$5,800 $(= 5,300 + 500 + 0 + 0)$
Rescheduling process	$5,000 $(= 4,500 + 500 + 0 + 0)$

6.1.3 *Quality*

Evaluating precisely the quality of a process is not trivial, because it is concerned with many different process perspectives. For simplification purposes, Dumas et al. proposed that the repetitions could be used to evaluate the quality of processes in the sense that repetitive work increases their cycle time (Dumas et al., 2018).

According to Dumas's work, we measure the quality of as-is/to-be processes based on the repetition probability (r) of rework blocks affected by the business process redesign framework. In addition, the quality of a process is inversely proportional to the repetition probability, mathematically $100\% - r$. Ultimately, we inherit scholarly work on strategic business modeling (Horkoff et al., 2014) to calculate the performance level of quality of the selling process and rescheduling process as in Table 8.

Table 8: Comparing the quality of as-is/to-be processes

Process		**Rework block**	**r**	**Quality (100% -r)**	**Performance level**
Selling last-minutes tickets process	As-is	Search flights	70%	30%	$\frac{60\%-30\%}{100\%-30\%} = 0.43$
	To-be	Search flights	40%	60%	
Rescheduling process	As-is	Make a list of rescheduling options, Pick a rescheduling option or not	80%	20%	$\frac{70\%-20\%}{100\%-20\%} = 0.63$
	To-be	Make a list of rescheduling options, Create payment instruction, Pick a rescheduling option or not	30%	70%	

6.1.4 *Flexibility*

The flexibility of a process is represented by the amount of variations allowed. It is important for an enterprise to compare its expected flexibility with the actual one. From a business perspective, it might turn out that the process is more/less flexible than what is expected. Dumas et al. proposed an approach to quantify the degree of optionality as a fraction of the number of optional tasks and the amount of all tasks in the process (Dumas et al., 2018).

We apply this approach to the case study of Company AD for measuring the flexibility of the selling process and rescheduling process, as presented in

Table 9. In order to calculate the performance level of flexibility depicted in Table 9, we also use the aforementioned formulas on indicator performance levels of Horkoff et al. presented in the second paragraph of Section 6.

Table 9: Comparing the flexibility of as-is/to-be processes

Process		**Optional tasks**	**Number of total tasks**	**Flexibility**	**Performance level**
Selling process	As-is	Top-up e-Wallet, Warn customer, Refund customer	13	$\frac{3}{13}$	$\frac{\frac{2}{12}-\frac{3}{13}}{\frac{3}{13}-\frac{0}{13}}=-0.278$
	To-be	Top-up e-Wallet, Warn customer	12	$\frac{2}{12}$	
Rescheduling process	As-is	Notify customer, Make a list of rescheduling options, Pick a rescheduling option or not, Create payment instruction, Make payment, Pay the Airlines, Send the tickets & result of payment, Refund customer	10	$\frac{8}{10}$	$\frac{\frac{7}{9}-\frac{8}{10}}{\frac{8}{10}-\frac{0}{10}}=-0.028$
	To-be	Notify customer, Make a list of rescheduling options, Pick a rescheduling option or not, Create payment instruction, Make payment, Pay the Airlines, Send the tickets & result of payment	9	$\frac{7}{9}$	

6.1.5 Transparency

Whether a process is transparent depends on its participants' view. The level of transparency of a process in a specific perspective refers to the ratio of the number of transparent tasks to the total number of tasks. A process or sub-process is called transparent if all of its tasks or sub-processes are transparent. We call a task explicit if it is neither a sub-process, nor a call activity (i.e., an activity that references an external process).

We propose the following formula to measure the transparency of a process.

$$Transparent\ Level\ (specific\ view) = \frac{Number\ of\ explicit\ tasks\ (in\ specific\ view)}{Number\ of\ explicit\ tasks\ (in\ full\ view)}$$

The aforementioned formulas of Horkoff et al. presented in the second paragraph of Section 6 are used to calculate performance level of transparency of processes.

In our case study, we view both the selling and rescheduling processes in four perspectives. We determine the performance level of transparency for each view in two different modes as-is/to-be in Table 10 (selling) and Table 11 (rescheduling) respectively. Appendix C.3 elaborates how we compute the values given in these two tables.

Table 10: Transparency of the selling process

View	Current value	Threshold value	Target value	Worst value	Performance level
Customer service department's view	33.3%	30.8%	100%		$\frac{33.3\%-30.8\%}{100\%-30.8\%} = 0.036$
Customer's view	41.7%	38.5%	100%		$\frac{41.7\%-38.5\%}{100\%-38.5\%} = 0.052$
Finance department's view	25%	30.8%		0%	$\frac{25\%-30.8\%}{30.8\%-0\%} = -0.188$

Table 11: Transparency of the rescheduling process

View	Current value	Threshold value	Target value	Worst value	Performance level
Customer service department's view	44.5%	40%	100%		$\frac{44.5\%-40\%}{100\%-40\%} = 0.075$
Customer's view	33.3%	30%	100%		$\frac{33.3\%-30\%}{100\%-30\%} = 0.0475$
Finance department's view	22.2%	30%		0%	$\frac{22.2\%-30\%}{30\%-0\%} = -0.26$

6.1.6 Exception handling

Exceptions are events that diverge a process from its normal scenarios to unexpected situations rarely happening. They include business faults (e.g., business rule violation, out-of-stock) and technology faults (e.g., database crashed, constraint violation) (Dumas et al., 2018). They might cause the abortion or interruption of the running process. To classify exceptions, Russel et al. addressed that exceptions are divided into various types, such as resource unavailability, deadline expiry, work item failure, external trigger, and constraint violation (Russell et al., 2006). While Dumas et al. assign the exceptions to process abortion, internal exceptions, external exceptions, activity timeouts, non-interrupting event and complex exceptions, interlude event sub-processes, and activity compensation (Dumas et al., 2018). For modeling these exceptions, BPMN provides the *terminate* event to abort the process, the *error* event to interrupt the running process, the *non-interrupting* event to handle complex exceptions, and the *intermediate timer* event to constrain an activity must be completed within a given timeframe.

We propose a formula to measure the exception handling as the ratio of the number of exceptions that are actually handled with respect to the amount of all potential exceptions.

$$Exception\ Handling = \frac{Number\ of\ Handled\ Exceptions}{Number\ of\ Handled\ Exceptions + Number\ of\ Unhandled\ Exceptions}$$

Also in this case, we apply the aforementioned formulas of Horkoff et al. presented in the second paragraph of Section 6 to our case study, in order to calculate the performance level of exception handling of the selling and rescheduling processes as in Table 12.

Table 12: Process exception handling

Process		**All potential exceptions**	**Number of handled exceptions**	**Number of unhandled exceptions**	**Exception handling**	**Performance level**
Selling process	As-is	Error occurs while topping up eWallet, Tickets sold out before AD pays the Airlines, Error occurs if eWallet balance is not enough for paying the tickets, Error occurs if the customers submit their information incorrectly	4	0	$\frac{4}{4}$	$\frac{\frac{3}{4}-\frac{4}{4}}{\frac{4}{4}-\frac{0}{4}} = -0.25$
	To-be		3	1	$\frac{3}{4}$	
Rescheduling process	As-is	Error occurs the tickets cannot be rescheduled, Tickets sold out before AD pays the Airlines	2	0	$\frac{2}{2}$	$\frac{\frac{1}{2}-\frac{2}{2}}{\frac{2}{2}-\frac{0}{2}} = -0.5$
	To-be		1	1	$\frac{1}{2}$	

6.1.7 Comparing the alternatives

In Figure 3, in order to satisfy the goal "leader pricing" (G0), all its AND-refined sub-goals, including the goals "to reduce service fees" and "to reduce ticket costs", are satisfied. Similarly, the goals "to reduce payment fees" (new goal) and "to reduce manual tasks in operation" (it and its sub-goals all exist in Figure 2, i.e., they are all satisfied in the original goal model) have to be achieved to obtain the goal "to reduce service fees". In contrast, the goal "to reduce ticket costs" might be acquired in several ways by achieving either the goal "to bargain with suppliers" (it is a new goal related to a new process – P3, which needs to be designed) or the goal "to find out new suppliers" (exists in Figure 2) because its sub-goals are OR-refined. By the same token, the goal "to reduce payment fees" will be satisfied by obtaining either the goal "to optimize payment process for selling tickets" (G1) or the goal "to optimize payment process for rescheduling tickets" (G2). Goals G1, G2 are related to processes P1' (redesigned selling process) and P2' (redesigned rescheduling process) respectively. Overall, in order to satisfy the goal G0, we only need to achieve either the goal G1 or the goal G2 corresponding to redesigning selling process (P1') or rescheduling process (P2').

In alternative 1 (P1'), we have achieved the goal G1 by redesigning the selling process to save time (Time Savings = 51.2 minutes/transaction) and cost (Cost Savings = \$4.096/transaction) of selling process leading to decrease payment fees for customer. In alternative 2 (P2'), we have achieved the goal G2 by redesigning rescheduling process to save time (Time Savings = 41.4 minutes/transaction) and cost (Cost Savings = \$3.312/transaction) of rescheduling process also leading to decrease payment fees for customer. Moreover, we address the comparison of all indicators of Alternative 1 and Alternative 2 in Table 13 for evaluating our re-engineering outcomes. The former is better than the latter when it comes to time savings and cost savings. In terms of switching cost, the former process is worse. Both alternatives are not doing well in terms of flexibility and exception handling. To conclude which

Table 13: Putting all indicators together

Indicators / Performance level of indicators	**Priority**	**Alternative 1**		**Alternative 2**	
Time Savings	10	51.2 minutes/transaction		41.4 minutes/transaction	
Cost Savings	9	$4.096/transaction		$3.312/transaction	
Switching Costs	7	$5,800		$5,000	
Performance level of quality	5	+0.43		+0.63	
Performance level of flexibility	2	-0.278		-0.028	
Performance level of transparency	2	Customer service department's view	+0.036	Customer service department's view	+0.075
		Customer's view	+0.052	Customer's view	+0.0475
		Finance department's view	-0.188	Finance department's view	-0.26
Performance level of exception handling	3	-0.25		-0.5	

alternative is preferable, we suggest assigning a priority level to each indicator to match the expectation of the enterprise's decision makers.

7 Final considerations

While the literature in information systems has enjoyed a growing interest in enterprise re-engineering especially since Industry 4.0 has gained pace, mainstream research in the field falls short in pinpointing crisply what and how to improve the operational level. We take the original rationale of business intelligence whereby data analytics offers an insight into how the enterprise in question has been operationalized of late, in order to introduce a novel approach to enterprise re-engineering with a business process focus. Our work is shaped in light of design science in information systems research. Our re-engineering approach tackles three research questions: (RQ1) actionable redesign of business processes thanks to data mining; (RQ2) business objectives as blueprint and motivation for redesign attempts; (RQ3) grounding business objects in redesign algorithms. More specifically, we find out tasks that could be enhanced and digitized thanks to data mining. We opt to eliminate redundant tasks and re-sequence inefficiently-located tasks. Our re-engineering framework has been applied to a small-sized company retailing domestic airfare in Vietnam. We propose an evaluation grid for assessing the possible to-be processes using common indicators.

In the big picture, our contribution is threefold. First, we have made business intelligence, requirements analysis and data mining applicable to the field of process improvement (addressing research questions *RQ1* and *RQ2*). Second, by devising algorithms that pinpoint where in the enterprise operational levels the inefficiency originates from, we open the door to automatic process fine-tuning (answering research question *RQ3*).

Third, methodologically we confirm the guidelines proposed in design science of information systems research such as the problem relevance, rigor, design artifact, evaluation and communication over the realm of enterprise re-engineering.

Discussions: Our approach has some shortcomings – leaving room for improvement, which are openly discussed below. Moreover, as digital transformation would draw traction in business engineering in general (and in business process discipline in particular), the concept of process improvement might be revisited when more emerging technologies become readily available.

- *Only AND connectives considered.* Currently, for simplification purposes, we support only AND connectives in the logical decomposition of goal-oriented requirements while several types of connectives are actually on the table. We intend to tackle this in future work, as a way to explore more alternatives.
- *Gateways and tacit knowledge still untouched.* Our redesign algorithms may move some process tasks without mentioning the consequence of changing dependencies between the tasks that were actually moved. This limitation is due to the fact that we formalize our processes in terms of scenario labels following fixed gateways. We intend to enhance the expressiveness of our ACP models and revise our redesign algorithms in order to cover the dependencies between tasks. Additionally, in our algorithms, all tasks are assumed to be fully transparent (i.e., no tacit knowledge or hidden attributes allowed) despite the fact that tacit knowledge might exist in real-life processes.
- *Human intervention and the completeness of measurement indicators.* Despite being positioned as a semi-automatic re-engineering framework, a good part of it is made of human intervention, for instance, modeling the as-is processes, developing goal models, assessing data mining models, and specifying business rules. This heavy bootstrapping leads to too much background knowledge and domain expertise being required when we deploy the said framework. Although our supported indicators form a good basis for measuring the process performance comprehensively, we offer no evidence of the completeness of the set of indicators being used. This could lead in the future to new indicators being supported and evaluated also in relation to companies' needs.
- *Numerous factors influencing the adoption of our framework.* There are still some concerns lingering on the trustworthiness of data mining models being applied, the available budget for re-engineering (e.g. for some companies this can represent prohibitive costs of change implementation and having employees trained), the conceptual quality of goal models being specified, and the reliance of necessary expertise in constructing workable process models. All of these factors may inhibit the adoption of our framework. The factors on model quality and required expertise to build them are overarching with all model based approaches, and we still use them for

their added benefits in terms of exploration of alternatives, and facilitation of communication with stakeholders (i.e. decision makers in our case). We see room for improvement with respect to making re-engineering more affordable for companies, by guiding them into finding a trade-off between the gained value and the cost they have to invest to re-engineer their processes, which might consider a subset of the proposed changes.

Acknowledgements Work presented in this article is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number C2019-20-15. The authors are thankful to our colleague Khuong Nguyen-An for correcting and commenting on the mathematical formalization based on which re-design algorithms could have been devised.

References

van der Aalst WMP (2020) Academic View: Development of the Process Mining Discipline. In: Process Mining in Action, Springer, pp 181–196

Adesola S, Baines T (2005) Developing and evaluating a methodology for business process improvement. Business Process Management Journal 11(1):37–46

Alter S (2014) Engineering Enterprises for Emergent Change. In: Proceedings of 16^{th} IEEE Conference on Business Informatics, IEEE Computer Society, Geneva, Switzerland, pp 113–123

Barone D, Yu E, Won J, Jiang L, Mylopoulos J (2010) Enterprise Modeling for Business Intelligence. In: Proceedings of IFIP Working Conference on The Practice of Enterprise Modeling, PoEM, Springer Verlag, Delft, The Netherlands, pp 31–45

Ben-Ari M (2012) Mathematical Logic for Computer Science, 3rd edn. Springer

Benthem V, Johan, Doets K (2001) Higher-order logic. Handbook of philosophical logic, 2nd Edition, 2nd edn. Springer

Bhattacharya K, Gerede C, Hull R, Liu R, Su J (2007) Towards Formal Analysis of Artifact-Centric Business Process Models. In: Proceedings of 5^{th} International Conference on Business Process Management, Springer, Brisbane, Australia, pp 288–304

Bordeleau F, Mosconi E, de Santa-Eulalia LA (2018) Business Intelligence in Industry 4.0: State of the art and research opportunities. In: Proceedings of 51^{st} Hawaii International Conference on System Sciences, IEEE Computer Society, Hawaii, USA, pp 1–10

Chen H, Chiang R, Storey VC (2012) Business Intelligence and Analytics: From Big Data to Big Impact. MIS Quarterly 36(4):1165–1188

Dam H, Lê LS, Ghose A (2016) Managing changes in the enterprise architecture modelling context. Enterprise Information Systems 10(6):666–696

Deokar AV, Sarnikar S (2016) Understanding process change management in electronic health record implementations. Information System and e-Business Management 14(4):733–766

Dietz JL (2006) Enterprise Ontology: Theory and Methodology. Springer

Donaldson WM, Blackburn TD, Blessner P, Olson BA (2015) An Examination of the Role of Enterprise Architecture Frameworks in Enterprise Transformation. Journal of Enterprise Transformation 5(3):218–240

Dumas M, Rosa ML, Mendling J, Reijers HA (2018) Fundamentals of Business Process Management, Second Edition. Springer

Fill HG, Johannsen F (2016) A Knowledge Perspective on Big Data by Joining Enterprise Modeling and Data Analyses. In: Proceedings of 49^{th} Hawaii International Conference on System Sciences, IEEE Computer Society, Hawaii, USA, pp 4052–4061

Gharib M, Giorgini P, Mylopoulos J (2018) Analysis of information quality requirements in business processes, revisited. Requirements Engineering 23(2):227–249

Giorgini P, Mylopoulos J, Nicchiarelli E, Sebastiani R (2002) Reasoning with Goal Models. In: Proceedings of the 21^{st} International Conference on Conceptual Modeling, Tampere, Finland, pp 167–181

Hacks S, Steffens A, Hansen P, Rajashekar N (2019) A Continuous Delivery Pipeline for EA Model Evolution. In: Proceedings of Enterprise, Business-Process and Information Systems Modeling, BPMDS, EMMSAD, Springer, Rome, Italy, pp 141–155

Han J, Kamber M, Pei J (2011) Data mining: Concepts and Techniques, 3rd edn. Morgan Kaufmann

Hevner AR, March ST, Park J, Ram S (2004) Design Science in Information Systems Research. MIS Quarterly 28(1):75–105

Hinge K, Ghose A, Koliadis G (2009) Process SEER: a Tool for Semantic Effect Annotation of Business Process Models. In: Proceedings of 13^{th} International Conference on Enterprise Distributed Object Computing, IEEE Computer Society, Auckland, New Zealand, pp 49–58

Hinkelmann K, Pasquini A (2014) Supporting Business and IT Alignment by Modeling Business and IT Strategy and Its Relations to Enterprise Architecture. In: Proceedings of 2^{nd} Enterprise Systems Conference, IEEE Computer Society, Shanghai, China, pp 149–154

Hinkelmann K, Gerber A, Karagiannis D, Thoenssen B, van der Merwe A, Woitsch R (2016) A new paradigm for the continuous alignment of business and it: Combining enterprise architecture modelling and enterprise ontology. Computers in Industry 79:77–86, special Issue on Future Perspectives On Next Generation Enterprise Information Systems

Horkoff J, Yu E (2009) Evaluating Goal Achievement in Enterprise Modeling – An Interactive Procedure and Experiences. In: Proceedings of the Second Working Conference on The Practice of Enterprise Modeling, Stockholm, Sweden, pp 145–160

Horkoff J, Barone D, Jiang L, Yu E, Amyot D, Borgida A, Mylopoulos J (2014) Strategic business modeling: representation and reasoning. Software and System Modeling 13(3):1015–1041

Iacob ME, D Q, Jonkers H (2012) Capturing Business Strategy and Value in Enterprise Architecture to Support Portfolio Valuation. In: Proceedings of 16^{th} International Conference on Enterprise Distributed Object Computing, IEEE Computer Society, Beijing, China, pp 11–20

Jarvenpaa SL, Stoddard DB (1998) Business process redesign: Radical and evolutionary change. Journal of Business Research 41(1):15–27

Kallio J, Saarinen T, Salo S, Tinnilä M, Vepsäläinen A (1999) Drivers and tracers of business process changes. The Journal of Strategic Information Systems 8(2):125–142

Kaplan RS, Anderson SR (2003) Time-Driven Activity-Based Costing. Tech. Rep. 485443, Harvard Business School: Accounting & Management

Koliadis G, Ghose AK (2009) A Conceptual Framework for Business Process Redesign. In: Proceedings of the 10^{th} International Conference on Enterprise, Business-Process and Information Systems Modeling, held at 21^{st} CAiSE, Springer, Amsterdam, The Netherlands, pp 14–26

Kunchala J, Yu J, Yongchareon S (2015) A Survey on Approaches to Modeling Artifact-Centric Business Processes. In: Proceedings of 16^{th} International Conference on Web Information System Engineering, Springer, Miami, USA, pp 16–18

Labusch N, Aier S, Winter R (2013) Beyond Enterprise Architecture Modeling – What are the Essentials to Support Enterprise Transformations? In: Proceedings of 10^{th} International Workshop on Enterprise Modeling and Information Systems Architectures, St. Gallen, Switzerland, pp 13–26

Labusch N, Aier S, Rothenberger M, Winter R (2014) Architectural Support of Enterprise Transformations: Insights from Corporate Practice. In: Tagungsband Multikonferenz Wirtschaftsinformatik, Universität Paderborn, Paderborn, Germany, pp 1048–1060

Laudon KC, Laudon JP (2019) Management Information Systems: Managing the Digital Firm, 16th edn. Pearson

Mansar SL, Reijers HA (2007) Best Practices in Business Process Redesign: Use and Impact. Business Process Management Journal 13(2):193–213

Maté A, Trujillo J, Mylopoulos J (2015) Stress Testing Strategic Goals with SWOT Analysis. In: Proceedings of the 34^{th} International Conference on Conceptual Modeling, Stockholm, Sweden, pp 65–78

Meyer B (1988) Object-Oriented Software Construction, 1st edn. Prentice-Hall

Michelberger B, Mutschler B, Reichert M (2012) Process-oriented Information Logistics: Aligning Enterprise Information with Business Processes. In: Proceedings of 16^{th} International Conference on Enterprise Distributed Object Computing, IEEE Computer Society, Beijing, China, pp 21–30

Müller D, Reichert M, Herbst J (2008) A New Paradigm for the Enactment and Dynamic Adaptation of Data-Driven Process Structures. In: Proceedings of the 20^{th} International Conference on Advanced Information Systems Engineering, France, pp 48–63

Niedermann F, Radeschütz S, Mitschang B (2010) Design-Time Process Optimization through Optimization Patterns and Process Model Matching. In: Proceedings of 12^{th} IEEE Conference on Commerce and Enterprise Computing, IEEE Computer Society, Shanghai, China, pp 48–55

Nigam A, Caswell NS (2003) Business artifacts: An approach to operational specification. IBM Systems Journal 42(3):428–445

Österle H (2013) Business in the Information Age: Heading for New Processes. Springer Science & Business Media

Paja E, Maté A, Woo CC, Mylopoulos J (2016) Can Goal Reasoning Techniques Be Used for Strategic Decision-Making? In: Proceedings of the 35^{th} International Conference on Conceptual Modeling, Gifu, Japan, pp 530–543

Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S (2008) A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24(3):45–77

Pombinho J, Aveiro D, Tribolet J (2014) A Matching Ontology for e^3 Value and DEMO – A Sound Bridging of Business Modelling and Enterprise Engineering. In: Proceedings of 16^{th} IEEE Conference on Business Informatics, IEEE Computer Society, Geneva, Switzerland, pp 17–24

Qu Y, Ming X, Ni Y, Li X, Liu Z, Zhang X, Xie L (2019) An integrated framework of enterprise information systems in smart manufacturing system via business process reengineering. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 233(11):2210–2224

Reijers HA (2003) Design and Control of Workflow Processes: Business Process Management for the Service Industry, Lecture Notes in Computer Science, vol 2617. Springer

Rouse WB (2005) A Theory of Enterprise Transformation. Systems Engineering 8(4):279–295

Rupnik R, Jaklič J (2009) The Deployment of Data Mining into Operational Business Processes. In: Ponce J, Karahoca A (eds) Data Mining and Knowledge Discovery in Real Life Applications, I-Tech Education and Publishing

Russell N, van der Aalst W, Ter-Hofstede A (2006) Exception Handling Patterns in Process-Aware Information Systems. Tech. rep., BPM Center Report BPM-06-04, BPMcenter. org

Salgado CE, Machado RJ, Maciel R (2014) An OMG-based Meta-Framework for Alignment of IS/IT Architecture with Business Models. In: Proceedings of 9^{th} International Conference on the Quality of Information and Communications Technology, IEEE Computer Society, Guimaraes, Portugal, pp 288–293

Schmidt R, Möhring M (2013) Strategic alignment of cloud-based architectures for big data. In: Proceedings of 5^{th} Workshop on Service-Oriented Enterprise Architecture for Enterprise Engineering, in conjunction with 17^{th} EDOC conference, IEEE Computer Society, Vancouver, Canada, pp 136–143

Shahzad K, Zdravkovic J (2009) A Goal-Oriented Approach for Business Process Improvement Using Process Warehouse Data. In: Proceedings of the 2^{nd} IFIP Working Conference on The Practice of Enterprise Modeling, Springer, Stockholm, Sweden, pp 84–98

Sousa VAD, Burnay C, Snoeck M (2020) B-MERODE: A Model-Driven Engineering and Artifact-Centric Approach to Generate Blockchain-Based Information Systems. In: Proceedings of 32^{nd} International Conference on Advanced Information Systems Engineering, CAiSE 2020, Grenoble, France, pp 117–133

Ştefănescu L, Ungureanu L, Matei IV (2008) Reengineering as an Efficient Solution to Redesign Activities and Processes of an Enterprise. Journal of Applied Economic Sciences 3(5):161–167

Teng J, Grover V, Fiedler K (1996) Developing strategic perspectives on business process reengineering: From process reconfiguration to organizational change. Omega 24(3):271–294

Tribolet J (2014) An Engineering Approach to Natural Enterprise Dynamics: From Top-down Purposeful Systemic Steering to Bottom-up Adaptive Guidance Control. In: Proceedings of 9^{th} International Conference on Evaluation of Novel Approaches to Software Engineering, IEEE Computer Society, Lisbon, Portugal, p keynote

Truong TM, Lê LS (2016) Towards a Formal Framework for Business Process Re-Design Based on Data Mining. In: Proceedings of 17^{th} Working Conference on Business Process Modeling, Development and Support, Springer, Ljubljana, Slovenia, pp 250–265

Truong TM, Lê LS, Tôn LP (2017) Re-engineering Enterprises Using Data Warehouse as a Driver and Requirements as an Enabler. In: Proceedings of the 21^{st} International Conference on Enterprise Distributed Object Computing, IEEE Computer Society, Quebec City, Canada, pp 67–72

Umoh E, Sampaio PRF (2014) The 'REFINTO' Framework and Tool: Supporting Business-IT Alignment in Enterprise Financial Application Development. In: Proceedings of 18^{th} International Conference on Enterprise Distributed Object Computing, IEEE Computer Society, Ulm, Germany, pp 406–409

Vanderfeesten ITP, Reijers HA, van der Aalst WMP (2011) Product-Based Workflow Support. Information Systems 36(2):517–535

Vanwersch R, Shahzad K, Vanderfeesten I, Vanhaecht K, Grefen P, Pintelon L, Mendling J, van Merode G, Reijers HA (2016) A Critical Evaluation and Framework of Business Process Improvement Methods. Business & Information Systems Engineering 58(1):43–53

Voorberg S, Eshuis R, van Jaarsveld W, van Houtum G (2019) Decision Support for Declarative Artifact-Centric Process Models. In: Proceedings of Business Process Management Forum - BPM Forum 2019, Vienna, Austria, pp 36–52

Wegener D, Rüping S (2010a) On Integrating Data Mining into Business Processes. In: Proceedings of the 13^{th} International Conference on Business Information Systems, Springer, Berlin, Germany, pp 183–194

Wegener D, Rüping S (2010b) On Reusing Data Mining in Business Processes - a Pattern-Based Approach. In: Proceedings of 8^{th} BPM International Workshops and Education Track, Springer, Hoboken, USA, pp 264–276

Wegmann A, Regev G, Rychkova I, Lê LS, Julia P (2007) Business and IT alignment with SEAM for enterprise architecture. In: Proceedings of 11^{th} International Conference on Enterprise Distributed Object Computing, IEEE Computer Society, Annapolis, USA, p 111

Winter R (2001) Working for E-Business – The Business Engineering Approach. International Journal of Business Studies 9(1):101–117

Xu LD, Xu EL, Li L (2018) Industry 4.0: state of the art and future trends. International Journal of Production Research 56(8):2941–2962

Yongchareon S, Liu C (2010) A Process View Framework for Artifact-Centric Business Processes. In: Proceedings Part I of On the Move to Meaningful Internet Systems: Confederated International Conferences: CoopIS, IS, DOA and ODBASE, Springer, Crete, Greece, pp 26–43

Yousfi A, Saidi R, Dey AK (2016) Variability patterns for business processes in BPMN. Information System and e-Business Management 14(3):443–467

Zimmermann A, Schmidt R, Sandkuhl K, Wißotzki M, Jugel D, Möhring M (2015) Digital Enterprise Architecture – Transformation for the Internet of Things. In: 19^{th} International Enterprise Distributed Object Computing Workshops, IEEE Computer Society, Adelaide, Australia, pp 130–138

A Classification rules

Tree

```
provider = BL
|   serviceClass = HotFare_Max: UNCHANGED {UNCHANGED=1020, DOWN=355, UP=243, SKYROCKETS=0, PLUMMETS=1}
|   serviceClass = HotFare_Plus
|   |   priceUSD = P_13.82: SKYROCKETS {UNCHANGED=18, DOWN=0, UP=0, SKYROCKETS=32, PLUMMETS=0}
|   |   priceUSD = P_16.36: UNCHANGED {UNCHANGED=12, DOWN=0, UP=8, SKYROCKETS=12, PLUMMETS=2}
|   |   priceUSD = P_17.24
|   |   |   DepartureDay = Friday: UP {UNCHANGED=1, DOWN=1, UP=3, SKYROCKETS=0, PLUMMETS=0}
|   |   |   DepartureDay = Monday: UNCHANGED {UNCHANGED=3, DOWN=1, UP=2, SKYROCKETS=2, PLUMMETS=0}
|   |   |   DepartureDay = Saturday: UNCHANGED {UNCHANGED=3, DOWN=2, UP=2, SKYROCKETS=2, PLUMMETS=0}
|   |   |   DepartureDay = Sunday: SKYROCKETS {UNCHANGED=2, DOWN=1, UP=1, SKYROCKETS=4, PLUMMETS=0}
|   |   |   DepartureDay = Thursday: UP {UNCHANGED=3, DOWN=0, UP=4, SKYROCKETS=2, PLUMMETS=1}
|   |   |   DepartureDay = Tuesday: UNCHANGED {UNCHANGED=3, DOWN=1, UP=2, SKYROCKETS=1, PLUMMETS=0}
|   |   |   DepartureDay = Wednesday: UP {UNCHANGED=2, DOWN=0, UP=3, SKYROCKETS=1, PLUMMETS=0}
|   |   priceUSD = P_17.33: UNCHANGED {UNCHANGED=232, DOWN=8, UP=4, SKYROCKETS=78, PLUMMETS=1}
|   |   priceUSD = P_18.27: SKYROCKETS {UNCHANGED=6, DOWN=4, UP=0, SKYROCKETS=7, PLUMMETS=0}
|   |   priceUSD = P_21.69: PLUMMETS {UNCHANGED=4, DOWN=0, UP=0, SKYROCKETS=3, PLUMMETS=5}
|   |   priceUSD = P_21.78: UNCHANGED {UNCHANGED=249, DOWN=0, UP=0, SKYROCKETS=72, PLUMMETS=56}
|   |   priceUSD = P_24: UNCHANGED {UNCHANGED=170, DOWN=23, UP=47, SKYROCKETS=11, PLUMMETS=22}
|   |   priceUSD = P_26.22: UNCHANGED {UNCHANGED=131, DOWN=14, UP=0, SKYROCKETS=53, PLUMMETS=20}
|   |   priceUSD = P_29.33
|   |   |   DepartureDay = Friday: UNCHANGED {UNCHANGED=15, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=9}
|   |   |   DepartureDay = Monday
|   |   |   |   DepartureSeason = Fall: UNCHANGED {UNCHANGED=25, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=12}
|   |   |   |   DepartureSeason = Spring: PLUMMETS {UNCHANGED=2, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=7}
|   |   |   |   DepartureSeason = Winter
|   |   |   |   |   DifferentDays = (10, 30]: UNCHANGED {UNCHANGED=4, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=2}
|   |   |   |   |   DifferentDays = (30, 90]: UNCHANGED {UNCHANGED=20, DOWN=0, UP=0, SKYROCKETS=0,
PLUMMETS=4}
|   |   |   |   |   DifferentDays = [0, 10]: PLUMMETS {UNCHANGED=2, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=4}
|   |   |   DepartureDay = Saturday: UNCHANGED {UNCHANGED=21, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=2}
|   |   |   DepartureDay = Sunday: UNCHANGED {UNCHANGED=65, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=10}
|   |   |   DepartureDay = Thursday: UNCHANGED {UNCHANGED=10, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=10}
|   |   |   DepartureDay = Tuesday: UNCHANGED {UNCHANGED=14, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=8}
|   |   |   DepartureDay = Wednesday: PLUMMETS {UNCHANGED=4, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=11}
|   serviceClass = HotFare_Starter
|   |   priceUSD = P_0: UNCHANGED {UNCHANGED=6, DOWN=0, UP=0, SKYROCKETS=6, PLUMMETS=0}
|   |   priceUSD = P_0.49: SKYROCKETS {UNCHANGED=14, DOWN=0, UP=0, SKYROCKETS=36, PLUMMETS=0}
|   |   priceUSD = P_10.67: UNCHANGED {UNCHANGED=579, DOWN=0, UP=0, SKYROCKETS=299, PLUMMETS=80}
|   |   priceUSD = P_12.89: UNCHANGED {UNCHANGED=488, DOWN=0, UP=0, SKYROCKETS=272, PLUMMETS=171}
|   |   priceUSD = P_16: UNCHANGED {UNCHANGED=873, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=344}
|   |   priceUSD = P_3.02: SKYROCKETS {UNCHANGED=12, DOWN=0, UP=0, SKYROCKETS=19, PLUMMETS=3}
|   |   priceUSD = P_3.91
|   |   |   DepartureDay = Friday: UP {UNCHANGED=1, DOWN=0, UP=2, SKYROCKETS=1, PLUMMETS=1}
|   |   |   DepartureDay = Monday: UNCHANGED {UNCHANGED=3, DOWN=0, UP=1, SKYROCKETS=3, PLUMMETS=1}
|   |   |   DepartureDay = Saturday: UNCHANGED {UNCHANGED=3, DOWN=0, UP=1, SKYROCKETS=3, PLUMMETS=2}
|   |   |   DepartureDay = Sunday: SKYROCKETS {UNCHANGED=2, DOWN=0, UP=0, SKYROCKETS=3, PLUMMETS=3}
|   |   |   DepartureDay = Thursday: SKYROCKETS {UNCHANGED=3, DOWN=0, UP=2, SKYROCKETS=5, PLUMMETS=0}
|   |   |   DepartureDay = Tuesday: UNCHANGED {UNCHANGED=3, DOWN=0, UP=1, SKYROCKETS=1, PLUMMETS=2}
|   |   |   DepartureDay = Wednesday: SKYROCKETS {UNCHANGED=2, DOWN=0, UP=1, SKYROCKETS=3, PLUMMETS=0}
|   |   priceUSD = P_4: UNCHANGED {UNCHANGED=391, DOWN=2, UP=0, SKYROCKETS=149, PLUMMETS=8}
|   |   priceUSD = P_4.93: SKYROCKETS {UNCHANGED=6, DOWN=0, UP=0, SKYROCKETS=8, PLUMMETS=3}
|   |   priceUSD = P_8.36: PLUMMETS {UNCHANGED=4, DOWN=0, UP=0, SKYROCKETS=2, PLUMMETS=6}
|   |   priceUSD = P_8.44
|   |   |   DifferentDays = (10, 30]: UNCHANGED {UNCHANGED=89, DOWN=0, UP=0, SKYROCKETS=48, PLUMMETS=7}
|   |   |   DifferentDays = (150, 240]: UNCHANGED {UNCHANGED=73, DOWN=0, UP=0, SKYROCKETS=8, PLUMMETS=0}
|   |   |   DifferentDays = (30, 90]: UNCHANGED {UNCHANGED=165, DOWN=0, UP=0, SKYROCKETS=60, PLUMMETS=51}
|   |   |   DifferentDays = (90, 150]
|   |   |   |   DepartureSeason = Fall
|   |   |   |   |   DepartureDay = Thursday: SKYROCKETS {UNCHANGED=2, DOWN=0, UP=0, SKYROCKETS=4, PLUMMETS=0}
|   |   |   |   |   DepartureDay = Tuesday: UNCHANGED {UNCHANGED=8, DOWN=0, UP=0, SKYROCKETS=1, PLUMMETS=0}
|   |   |   |   |   DepartureDay = Wednesday: UNCHANGED {UNCHANGED=7, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=0}
|   |   |   |   DepartureSeason = Spring: UNCHANGED {UNCHANGED=48, DOWN=0, UP=0, SKYROCKETS=8, PLUMMETS=0}
|   |   |   |   DepartureSeason = Summer: UNCHANGED {UNCHANGED=36, DOWN=0, UP=0, SKYROCKETS=20, PLUMMETS=2}
|   |   |   |   DepartureSeason = Winter: UNCHANGED {UNCHANGED=60, DOWN=0, UP=0, SKYROCKETS=5, PLUMMETS=3}
|   |   |   DifferentDays = >240: UNCHANGED {UNCHANGED=32, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=0}
|   |   |   DifferentDays = [0, 10]: SKYROCKETS {UNCHANGED=9, DOWN=0, UP=0, SKYROCKETS=12, PLUMMETS=4}
|   serviceClass = Max: UNCHANGED {UNCHANGED=689, DOWN=540, UP=282, SKYROCKETS=59, PLUMMETS=58}
|   serviceClass = Plus: UNCHANGED {UNCHANGED=693, DOWN=51, UP=276, SKYROCKETS=349, PLUMMETS=268}
|   serviceClass = Starter
|   |   priceUSD = P_0: SKYROCKETS {UNCHANGED=16, DOWN=0, UP=0, SKYROCKETS=34, PLUMMETS=0}
|   |   priceUSD = P_0.36: SKYROCKETS {UNCHANGED=6, DOWN=0, UP=0, SKYROCKETS=10, PLUMMETS=0}
|   |   priceUSD = P_0.49: UNCHANGED {UNCHANGED=61, DOWN=0, UP=0, SKYROCKETS=58, PLUMMETS=0}
|   |   priceUSD = P_10.67: UNCHANGED {UNCHANGED=293, DOWN=0, UP=0, SKYROCKETS=289, PLUMMETS=21}
|   |   priceUSD = P_12.89: UNCHANGED {UNCHANGED=254, DOWN=0, UP=0, SKYROCKETS=240, PLUMMETS=83}
|   |   priceUSD = P_16: UNCHANGED {UNCHANGED=421, DOWN=0, UP=0, SKYROCKETS=291, PLUMMETS=99}
|   |   priceUSD = P_20: UNCHANGED {UNCHANGED=897, DOWN=0, UP=0, SKYROCKETS=405, PLUMMETS=131}
|   |   priceUSD = P_24.44: UNCHANGED {UNCHANGED=611, DOWN=0, UP=0, SKYROCKETS=332, PLUMMETS=130}
|   |   priceUSD = P_28: UNCHANGED {UNCHANGED=408, DOWN=0, UP=0, SKYROCKETS=274, PLUMMETS=97}
|   |   priceUSD = P_3.02: SKYROCKETS {UNCHANGED=19, DOWN=0, UP=0, SKYROCKETS=21, PLUMMETS=2}
|   |   priceUSD = P_3.91
|   |   |   DepartureSeason = Fall
|   |   |   |   DifferentDays = (150, 240]: UNCHANGED {UNCHANGED=34, DOWN=0, UP=0, SKYROCKETS=20, PLUMMETS=0}
```

```
|   |   |   |   DifferentDays = (90, 150]: SKYROCKETS {UNCHANGED=2, DOWN=0, UP=0, SKYROCKETS=8, PLUMMETS=0}
|   |   |   DepartureSeason = Summer: SKYROCKETS {UNCHANGED=7, DOWN=0, UP=0, SKYROCKETS=8, PLUMMETS=0}
|   |   |   DepartureSeason = Winter: PLUMMETS {UNCHANGED=26, DOWN=0, UP=0, SKYROCKETS=17, PLUMMETS=29}
|   |   priceUSD = P_32
|   |   |   DifferentDays = (10, 30]: UNCHANGED {UNCHANGED=58, DOWN=0, UP=0, SKYROCKETS=37, PLUMMETS=30}
|   |   |   DifferentDays = (150, 240]: SKYROCKETS {UNCHANGED=4, DOWN=0, UP=0, SKYROCKETS=131, PLUMMETS=39}
|   |   |   DifferentDays = (30, 90]: UNCHANGED {UNCHANGED=18, DOWN=0, UP=0, SKYROCKETS=15, PLUMMETS=15}
|   |   |   DifferentDays = (90, 150]
|   |   |   |   DepartureSeason = Spring: SKYROCKETS {UNCHANGED=3, DOWN=0, UP=0, SKYROCKETS=41, PLUMMETS=16}
|   |   |   |   DepartureSeason = Summer: SKYROCKETS {UNCHANGED=1, DOWN=0, UP=0, SKYROCKETS=5, PLUMMETS=3}
|   |   |   |   DepartureSeason = Winter: UNCHANGED {UNCHANGED=9, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=2}
|   |   |   DifferentDays = >240: SKYROCKETS {UNCHANGED=1, DOWN=0, UP=0, SKYROCKETS=43, PLUMMETS=9}
|   |   |   DifferentDays = [0, 10]: SKYROCKETS {UNCHANGED=206, DOWN=0, UP=0, SKYROCKETS=216, PLUMMETS=54}
|   |   priceUSD = P_36: UNCHANGED {UNCHANGED=266, DOWN=0, UP=0, SKYROCKETS=201, PLUMMETS=87}
|   |   priceUSD = P_4: UNCHANGED {UNCHANGED=62, DOWN=0, UP=0, SKYROCKETS=55, PLUMMETS=0}
|   |   priceUSD = P_4.93: UNCHANGED {UNCHANGED=16, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=11}
|   |   priceUSD = P_40: UNCHANGED {UNCHANGED=189, DOWN=41, UP=120, SKYROCKETS=52, PLUMMETS=37}
|   |   priceUSD = P_44: UNCHANGED {UNCHANGED=175, DOWN=40, UP=0, SKYROCKETS=138, PLUMMETS=32}
|   |   priceUSD = P_48.44: UNCHANGED {UNCHANGED=162, DOWN=26, UP=105, SKYROCKETS=28, PLUMMETS=47}
|   |   priceUSD = P_52.89: UNCHANGED {UNCHANGED=114, DOWN=29, UP=0, SKYROCKETS=95, PLUMMETS=36}
|   |   priceUSD = P_58.22
|   |   |   DepartureDay = Friday: UNCHANGED {UNCHANGED=13, DOWN=3, UP=0, SKYROCKETS=13, PLUMMETS=8}
|   |   |   DepartureDay = Monday: UNCHANGED {UNCHANGED=31, DOWN=2, UP=0, SKYROCKETS=16, PLUMMETS=6}
|   |   |   DepartureDay = Saturday: UNCHANGED {UNCHANGED=22, DOWN=0, UP=0, SKYROCKETS=9, PLUMMETS=4}
|   |   |   DepartureDay = Sunday
|   |   |   |   DepartureSeason = Fall: UNCHANGED {UNCHANGED=4, DOWN=0, UP=0, SKYROCKETS=1, PLUMMETS=2}
|   |   |   |   DepartureSeason = Spring: UNCHANGED {UNCHANGED=14, DOWN=1, UP=0, SKYROCKETS=4, PLUMMETS=3}
|   |   |   |   DepartureSeason = Summer: SKYROCKETS {UNCHANGED=5, DOWN=0, UP=0, SKYROCKETS=6, PLUMMETS=2}
|   |   |   |   DepartureSeason = Winter: SKYROCKETS {UNCHANGED=3, DOWN=0, UP=0, SKYROCKETS=6, PLUMMETS=3}
|   |   |   DepartureDay = Thursday: PLUMMETS {UNCHANGED=1, DOWN=2, UP=0, SKYROCKETS=4, PLUMMETS=7}
|   |   |   DepartureDay = Tuesday: UNCHANGED {UNCHANGED=15, DOWN=2, UP=0, SKYROCKETS=12, PLUMMETS=0}
|   |   |   DepartureDay = Wednesday: UNCHANGED {UNCHANGED=9, DOWN=3, UP=0, SKYROCKETS=8, PLUMMETS=5}
|   |   priceUSD = P_6.58: SKYROCKETS {UNCHANGED=6, DOWN=0, UP=0, SKYROCKETS=16, PLUMMETS=2}
|   |   priceUSD = P_64.44: UNCHANGED {UNCHANGED=74, DOWN=15, UP=48, SKYROCKETS=0, PLUMMETS=28}
|   |   priceUSD = P_68.89
|   |   |   DepartureSeason = Fall: PLUMMETS {UNCHANGED=24, DOWN=1, UP=0, SKYROCKETS=0, PLUMMETS=89}
|   |   |   DepartureSeason = Spring
|   |   |   |   DifferentDays = (10, 30]: PLUMMETS {UNCHANGED=4, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=12}
|   |   |   |   DifferentDays = (150, 240]: UNCHANGED {UNCHANGED=15, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=0}
|   |   |   |   DifferentDays = (30, 90]: PLUMMETS {UNCHANGED=3, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=16}
|   |   |   |   DifferentDays = (90, 150]: PLUMMETS {UNCHANGED=3, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=53}
|   |   |   |   DifferentDays = >240: UNCHANGED {UNCHANGED=76, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=0}
|   |   |   |   DifferentDays = [0, 10]: UNCHANGED {UNCHANGED=11, DOWN=3, UP=0, SKYROCKETS=0, PLUMMETS=11}
|   |   |   DepartureSeason = Summer: PLUMMETS {UNCHANGED=14, DOWN=2, UP=0, SKYROCKETS=0, PLUMMETS=124}
|   |   |   DepartureSeason = Winter
|   |   |   |   DifferentDays = (10, 30]: PLUMMETS {UNCHANGED=1, DOWN=2, UP=0, SKYROCKETS=0, PLUMMETS=3}
|   |   |   |   DifferentDays = (150, 240]: UNCHANGED {UNCHANGED=103, DOWN=0, UP=0, SKYROCKETS=0,
PLUMMETS=53}
|   |   |   |   DifferentDays = (30, 90]: UNCHANGED {UNCHANGED=20, DOWN=1, UP=0, SKYROCKETS=0, PLUMMETS=2}
|   |   |   |   DifferentDays = (90, 150]: UNCHANGED {UNCHANGED=66, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=3}
|   |   |   |   DifferentDays = >240: UNCHANGED {UNCHANGED=161, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=3}
|   |   |   |   DifferentDays = [0, 10]
|   |   |   |   |   DepartureDay = Friday: UNCHANGED {UNCHANGED=5, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=4}
|   |   |   |   |   DepartureDay = Monday: UNCHANGED {UNCHANGED=10, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=3}
|   |   |   |   |   DepartureDay = Saturday: UNCHANGED {UNCHANGED=9, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=4}
|   |   |   |   |   DepartureDay = Sunday: UNCHANGED {UNCHANGED=6, DOWN=2, UP=0, SKYROCKETS=0, PLUMMETS=4}
|   |   |   |   |   DepartureDay = Thursday: PLUMMETS {UNCHANGED=2, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=4}
|   |   |   |   |   DepartureDay = Tuesday: UNCHANGED {UNCHANGED=3, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=3}
|   |   |   |   |   DepartureDay = Wednesday: UNCHANGED {UNCHANGED=6, DOWN=2, UP=0, SKYROCKETS=0, PLUMMETS=2}
|   |   priceUSD = P_8.36: UNCHANGED {UNCHANGED=64, DOWN=0, UP=10, SKYROCKETS=25, PLUMMETS=11}
|   |   priceUSD = P_8.44
|   |   |   DifferentDays = (10, 30]: UNCHANGED {UNCHANGED=54, DOWN=2, UP=0, SKYROCKETS=26, PLUMMETS=5}
|   |   |   DifferentDays = (150, 240]
|   |   |   |   DepartureSeason = Fall: PLUMMETS {UNCHANGED=18, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=25}
|   |   |   |   DepartureSeason = Summer: UNCHANGED {UNCHANGED=20, DOWN=0, UP=0, SKYROCKETS=6, PLUMMETS=0}
|   |   |   DifferentDays = (30, 90]: UNCHANGED {UNCHANGED=88, DOWN=15, UP=0, SKYROCKETS=47, PLUMMETS=3}
|   |   |   DifferentDays = (90, 150]: UNCHANGED {UNCHANGED=50, DOWN=0, UP=0, SKYROCKETS=30, PLUMMETS=5}
|   |   |   DifferentDays = >240: UNCHANGED {UNCHANGED=14, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=5}
|   |   |   DifferentDays = [0, 10]: SKYROCKETS {UNCHANGED=8, DOWN=1, UP=0, SKYROCKETS=11, PLUMMETS=2}
provider = VJ
|   priceUSD = P_0: SKYROCKETS {UNCHANGED=15, DOWN=0, UP=0, SKYROCKETS=35, PLUMMETS=0}
|   priceUSD = P_0.22: SKYROCKETS {UNCHANGED=5, DOWN=0, UP=0, SKYROCKETS=52, PLUMMETS=0}
|   priceUSD = P_1.29: SKYROCKETS {UNCHANGED=10, DOWN=0, UP=0, SKYROCKETS=18, PLUMMETS=5}
|   priceUSD = P_1.42: PLUMMETS {UNCHANGED=2, DOWN=0, UP=0, SKYROCKETS=1, PLUMMETS=2}
|   priceUSD = P_13.29
|   |   DifferentDays = (10, 30]: UNCHANGED {UNCHANGED=339, DOWN=0, UP=0, SKYROCKETS=200, PLUMMETS=51}
|   |   DifferentDays = (150, 240]: PLUMMETS {UNCHANGED=25, DOWN=0, UP=0, SKYROCKETS=17, PLUMMETS=54}
|   |   DifferentDays = (30, 90]: UNCHANGED {UNCHANGED=218, DOWN=0, UP=0, SKYROCKETS=152, PLUMMETS=84}
|   |   DifferentDays = (90, 150]
|   |   |   DepartureSeason = Fall: PLUMMETS {UNCHANGED=11, DOWN=0, UP=0, SKYROCKETS=11, PLUMMETS=37}
|   |   |   DepartureSeason = Spring: SKYROCKETS {UNCHANGED=0, DOWN=0, UP=0, SKYROCKETS=11, PLUMMETS=11}
|   |   |   DepartureSeason = Summer: PLUMMETS {UNCHANGED=3, DOWN=0, UP=0, SKYROCKETS=1, PLUMMETS=11}
|   |   |   DepartureSeason = Winter: UNCHANGED {UNCHANGED=14, DOWN=0, UP=0, SKYROCKETS=10, PLUMMETS=1}
|   |   DifferentDays = >240: PLUMMETS {UNCHANGED=4, DOWN=0, UP=0, SKYROCKETS=1, PLUMMETS=10}
```

```
|   |   DifferentDays = [0, 10]: SKYROCKETS {UNCHANGED=60, DOWN=0, UP=0, SKYROCKETS=76, PLUMMETS=2}
|   priceUSD = P_14.62
|   |   DifferentDays = (10, 30]: UNCHANGED {UNCHANGED=43, DOWN=42, UP=0, SKYROCKETS=41, PLUMMETS=12}
|   |   DifferentDays = (150, 240]: PLUMMETS {UNCHANGED=17, DOWN=5, UP=0, SKYROCKETS=4, PLUMMETS=48}
|   |   DifferentDays = (30, 90]: UNCHANGED {UNCHANGED=96, DOWN=56, UP=0, SKYROCKETS=65, PLUMMETS=40}
|   |   DifferentDays = (90, 150]: PLUMMETS {UNCHANGED=1, DOWN=3, UP=0, SKYROCKETS=1, PLUMMETS=7}
|   |   DifferentDays = >240: PLUMMETS {UNCHANGED=11, DOWN=1, UP=0, SKYROCKETS=2, PLUMMETS=48}
|   |   DifferentDays = [0, 10]: SKYROCKETS {UNCHANGED=4, DOWN=5, UP=0, SKYROCKETS=7, PLUMMETS=2}
|   priceUSD = P_17.33: UNCHANGED {UNCHANGED=1005, DOWN=0, UP=0, SKYROCKETS=733, PLUMMETS=265}
|   priceUSD = P_19.07: UNCHANGED {UNCHANGED=1112, DOWN=402, UP=0, SKYROCKETS=263, PLUMMETS=216}
|   priceUSD = P_2.09: UNCHANGED {UNCHANGED=259, DOWN=0, UP=0, SKYROCKETS=192, PLUMMETS=58}
|   priceUSD = P_2.3: DOWN {UNCHANGED=104, DOWN=180, UP=0, SKYROCKETS=7, PLUMMETS=22}
|   priceUSD = P_21.33: UNCHANGED {UNCHANGED=626, DOWN=0, UP=0, SKYROCKETS=358, PLUMMETS=154}
|   priceUSD = P_23.47
|   |   DepartureSeason = Fall: UNCHANGED {UNCHANGED=15, DOWN=5, UP=1, SKYROCKETS=2, PLUMMETS=8}
|   |   DepartureSeason = Spring: UNCHANGED {UNCHANGED=85, DOWN=39, UP=7, SKYROCKETS=23, PLUMMETS=36}
|   |   DepartureSeason = Summer: SKYROCKETS {UNCHANGED=3, DOWN=8, UP=5, SKYROCKETS=24, PLUMMETS=23}
|   |   DepartureSeason = Winter
|   |   |   DepartureDay = Friday: DOWN {UNCHANGED=1, DOWN=6, UP=0, SKYROCKETS=1, PLUMMETS=1}
|   |   |   DepartureDay = Monday: DOWN {UNCHANGED=1, DOWN=7, UP=2, SKYROCKETS=0, PLUMMETS=0}
|   |   |   DepartureDay = Saturday: UNCHANGED {UNCHANGED=5, DOWN=4, UP=3, SKYROCKETS=2, PLUMMETS=1}
|   |   |   DepartureDay = Sunday
|   |   |   |   DifferentDays = (10, 30]: PLUMMETS {UNCHANGED=2, DOWN=2, UP=1, SKYROCKETS=1, PLUMMETS=2}
|   |   |   |   DifferentDays = (30, 90]: UNCHANGED {UNCHANGED=5, DOWN=1, UP=1, SKYROCKETS=1, PLUMMETS=0}
|   |   |   |   DifferentDays = >240: DOWN {UNCHANGED=0, DOWN=2, UP=0, SKYROCKETS=1, PLUMMETS=2}
|   |   |   DepartureDay = Thursday: DOWN {UNCHANGED=2, DOWN=5, UP=0, SKYROCKETS=3, PLUMMETS=0}
|   |   |   DepartureDay = Tuesday: DOWN {UNCHANGED=5, DOWN=10, UP=1, SKYROCKETS=3, PLUMMETS=1}
|   |   |   DepartureDay = Wednesday: DOWN {UNCHANGED=4, DOWN=7, UP=1, SKYROCKETS=1, PLUMMETS=2}
|   priceUSD = P_25.78: UNCHANGED {UNCHANGED=457, DOWN=9, UP=0, SKYROCKETS=321, PLUMMETS=171}
|   priceUSD = P_28.36: UNCHANGED {UNCHANGED=138, DOWN=102, UP=9, SKYROCKETS=70, PLUMMETS=63}
|   priceUSD = P_29.33: UNCHANGED {UNCHANGED=326, DOWN=5, UP=0, SKYROCKETS=224, PLUMMETS=105}
|   priceUSD = P_3.07: PLUMMETS {UNCHANGED=10, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=28}
|   priceUSD = P_32.27: UNCHANGED {UNCHANGED=88, DOWN=50, UP=8, SKYROCKETS=32, PLUMMETS=61}
|   priceUSD = P_33.33: UNCHANGED {UNCHANGED=202, DOWN=3, UP=0, SKYROCKETS=173, PLUMMETS=83}
|   priceUSD = P_36.67
|   |   DepartureDay = Friday
|   |   |   DepartureSeason = Spring: DOWN {UNCHANGED=1, DOWN=3, UP=0, SKYROCKETS=2, PLUMMETS=2}
|   |   |   DepartureSeason = Winter: SKYROCKETS {UNCHANGED=0, DOWN=2, UP=1, SKYROCKETS=3, PLUMMETS=0}
|   |   DepartureDay = Monday
|   |   |   DepartureSeason = Spring: PLUMMETS {UNCHANGED=1, DOWN=2, UP=0, SKYROCKETS=0, PLUMMETS=3}
|   |   |   DepartureSeason = Winter: UNCHANGED {UNCHANGED=2, DOWN=1, UP=0, SKYROCKETS=1, PLUMMETS=1}
|   |   DepartureDay = Saturday
|   |   |   DepartureSeason = Spring: UNCHANGED {UNCHANGED=4, DOWN=3, UP=0, SKYROCKETS=4, PLUMMETS=0}
|   |   |   DepartureSeason = Winter: DOWN {UNCHANGED=3, DOWN=6, UP=1, SKYROCKETS=1, PLUMMETS=0}
|   |   DepartureDay = Sunday
|   |   |   DepartureSeason = Spring: UNCHANGED {UNCHANGED=6, DOWN=2, UP=1, SKYROCKETS=5, PLUMMETS=0}
|   |   |   DepartureSeason = Summer: UNCHANGED {UNCHANGED=7, DOWN=3, UP=1, SKYROCKETS=1, PLUMMETS=4}
|   |   |   DepartureSeason = Winter: UP {UNCHANGED=2, DOWN=1, UP=3, SKYROCKETS=2, PLUMMETS=1}
|   |   DepartureDay = Thursday
|   |   |   DepartureSeason = Spring: UNCHANGED {UNCHANGED=5, DOWN=2, UP=0, SKYROCKETS=0, PLUMMETS=1}
|   |   |   DepartureSeason = Winter: DOWN {UNCHANGED=2, DOWN=3, UP=0, SKYROCKETS=1, PLUMMETS=1}
|   |   DepartureDay = Tuesday
|   |   |   DepartureSeason = Spring: UNCHANGED {UNCHANGED=4, DOWN=1, UP=0, SKYROCKETS=2, PLUMMETS=0}
|   |   |   DepartureSeason = Winter: DOWN {UNCHANGED=1, DOWN=3, UP=2, SKYROCKETS=1, PLUMMETS=1}
|   |   DepartureDay = Wednesday: DOWN {UNCHANGED=5, DOWN=5, UP=2, SKYROCKETS=2, PLUMMETS=2}
|   priceUSD = P_37.33: UNCHANGED {UNCHANGED=172, DOWN=8, UP=0, SKYROCKETS=146, PLUMMETS=66}
|   priceUSD = P_4.4: SKYROCKETS {UNCHANGED=728, DOWN=0, UP=0, SKYROCKETS=854, PLUMMETS=0}
|   priceUSD = P_4.84: SKYROCKETS {UNCHANGED=288, DOWN=212, UP=0, SKYROCKETS=304, PLUMMETS=0}
|   priceUSD = P_41.07: UNCHANGED {UNCHANGED=37, DOWN=26, UP=10, SKYROCKETS=26, PLUMMETS=24}
|   priceUSD = P_41.33: UNCHANGED {UNCHANGED=155, DOWN=35, UP=64, SKYROCKETS=78, PLUMMETS=26}
|   priceUSD = P_45.33: UNCHANGED {UNCHANGED=110, DOWN=25, UP=47, SKYROCKETS=48, PLUMMETS=23}
|   priceUSD = P_45.47: DOWN {UNCHANGED=50, DOWN=51, UP=12, SKYROCKETS=12, PLUMMETS=17}
|   priceUSD = P_49.78: UNCHANGED {UNCHANGED=68, DOWN=16, UP=42, SKYROCKETS=42, PLUMMETS=27}
|   priceUSD = P_49.87
|   |   DepartureDay = Friday
|   |   |   DepartureSeason = Spring: UNCHANGED {UNCHANGED=2, DOWN=1, UP=1, SKYROCKETS=0, PLUMMETS=1}
|   |   |   DepartureSeason = Winter: DOWN {UNCHANGED=1, DOWN=5, UP=3, SKYROCKETS=0, PLUMMETS=0}
|   |   DepartureDay = Monday
|   |   |   DepartureSeason = Spring: UNCHANGED {UNCHANGED=10, DOWN=1, UP=0, SKYROCKETS=2, PLUMMETS=1}
|   |   |   DepartureSeason = Winter: DOWN {UNCHANGED=2, DOWN=4, UP=0, SKYROCKETS=1, PLUMMETS=2}
|   |   DepartureDay = Saturday
|   |   |   DepartureSeason = Spring: UNCHANGED {UNCHANGED=6, DOWN=2, UP=1, SKYROCKETS=0, PLUMMETS=1}
|   |   |   DepartureSeason = Winter: DOWN {UNCHANGED=1, DOWN=3, UP=1, SKYROCKETS=0, PLUMMETS=2}
|   |   DepartureDay = Sunday: DOWN {UNCHANGED=10, DOWN=13, UP=3, SKYROCKETS=0, PLUMMETS=3}
|   |   DepartureDay = Thursday: DOWN {UNCHANGED=2, DOWN=3, UP=2, SKYROCKETS=0, PLUMMETS=1}
|   |   DepartureDay = Tuesday: UNCHANGED {UNCHANGED=6, DOWN=1, UP=2, SKYROCKETS=2, PLUMMETS=2}
|   |   DepartureDay = Wednesday: UNCHANGED {UNCHANGED=2, DOWN=2, UP=0, SKYROCKETS=1, PLUMMETS=1}
|   priceUSD = P_54.22
|   |   DepartureDay = Friday: PLUMMETS {UNCHANGED=7, DOWN=4, UP=7, SKYROCKETS=2, PLUMMETS=7}
|   |   DepartureDay = Monday: PLUMMETS {UNCHANGED=4, DOWN=2, UP=3, SKYROCKETS=0, PLUMMETS=5}
|   |   DepartureDay = Saturday: UP {UNCHANGED=3, DOWN=1, UP=6, SKYROCKETS=0, PLUMMETS=1}
|   |   DepartureDay = Sunday: UP {UNCHANGED=17, DOWN=6, UP=32, SKYROCKETS=4, PLUMMETS=3}
|   |   DepartureDay = Thursday: UP {UNCHANGED=2, DOWN=1, UP=3, SKYROCKETS=1, PLUMMETS=0}
|   |   DepartureDay = Tuesday: UNCHANGED {UNCHANGED=7, DOWN=1, UP=3, SKYROCKETS=4, PLUMMETS=1}
```

```
|   |   DepartureDay = Wednesday: UP {UNCHANGED=2, DOWN=0, UP=5, SKYROCKETS=1, PLUMMETS=0}
|   priceUSD = P_54.76
|   |   DepartureDay = Friday: UP {UNCHANGED=3, DOWN=2, UP=4, SKYROCKETS=0, PLUMMETS=1}
|   |   DepartureDay = Monday
|   |   |   DepartureSeason = Spring: UNCHANGED {UNCHANGED=11, DOWN=2, UP=2, SKYROCKETS=2, PLUMMETS=1}
|   |   |   DepartureSeason = Winter: DOWN {UNCHANGED=3, DOWN=4, UP=0, SKYROCKETS=2, PLUMMETS=0}
|   |   DepartureDay = Saturday: UNCHANGED {UNCHANGED=4, DOWN=1, UP=2, SKYROCKETS=1, PLUMMETS=3}
|   |   DepartureDay = Sunday
|   |   |   DepartureSeason = Spring: UNCHANGED {UNCHANGED=11, DOWN=4, UP=1, SKYROCKETS=3, PLUMMETS=0}
|   |   |   DepartureSeason = Winter: DOWN {UNCHANGED=3, DOWN=8, UP=0, SKYROCKETS=1, PLUMMETS=0}
|   |   DepartureDay = Thursday: SKYROCKETS {UNCHANGED=1, DOWN=1, UP=1, SKYROCKETS=2, PLUMMETS=1}
|   |   DepartureDay = Tuesday: UNCHANGED {UNCHANGED=4, DOWN=3, UP=1, SKYROCKETS=1, PLUMMETS=1}
|   |   DepartureDay = Wednesday: DOWN {UNCHANGED=2, DOWN=3, UP=1, SKYROCKETS=0, PLUMMETS=3}
|   priceUSD = P_56.44
|   |   DifferentDays = (10, 30]: UNCHANGED {UNCHANGED=7, DOWN=0, UP=2, SKYROCKETS=0, PLUMMETS=0}
|   |   DifferentDays = [0, 10]
|   |   |   DepartureSeason = Spring: PLUMMETS {UNCHANGED=6, DOWN=1, UP=5, SKYROCKETS=0, PLUMMETS=7}
|   |   |   DepartureSeason = Summer: UP {UNCHANGED=6, DOWN=6, UP=19, SKYROCKETS=0, PLUMMETS=3}
|   priceUSD = P_58.67
|   |   DepartureSeason = Fall: UNCHANGED {UNCHANGED=4, DOWN=0, UP=2, SKYROCKETS=0, PLUMMETS=2}
|   |   DepartureSeason = Spring: UP {UNCHANGED=10, DOWN=2, UP=12, SKYROCKETS=0, PLUMMETS=4}
|   |   DepartureSeason = Summer: UNCHANGED {UNCHANGED=13, DOWN=9, UP=13, SKYROCKETS=1, PLUMMETS=10}
|   |   DepartureSeason = Winter
|   |   |   DifferentDays = (30, 90]: UNCHANGED {UNCHANGED=11, DOWN=0, UP=8, SKYROCKETS=1, PLUMMETS=0}
|   |   |   DifferentDays = (90, 150]: UNCHANGED {UNCHANGED=19, DOWN=1, UP=4, SKYROCKETS=0, PLUMMETS=0}
|   |   |   DifferentDays = [0, 10]: UP {UNCHANGED=9, DOWN=1, UP=11, SKYROCKETS=3, PLUMMETS=2}
|   priceUSD = P_59.64: DOWN {UNCHANGED=25, DOWN=26, UP=7, SKYROCKETS=6, PLUMMETS=12}
|   priceUSD = P_60.89: UNCHANGED {UNCHANGED=46, DOWN=15, UP=0, SKYROCKETS=0, PLUMMETS=35}
|   priceUSD = P_63.11
|   |   DepartureDay = Friday: UP {UNCHANGED=3, DOWN=0, UP=4, SKYROCKETS=0, PLUMMETS=0}
|   |   DepartureDay = Monday: UNCHANGED {UNCHANGED=6, DOWN=0, UP=3, SKYROCKETS=0, PLUMMETS=0}
|   |   DepartureDay = Sunday: UNCHANGED {UNCHANGED=15, DOWN=1, UP=5, SKYROCKETS=1, PLUMMETS=6}
|   |   DepartureDay = Thursday: UNCHANGED {UNCHANGED=5, DOWN=0, UP=2, SKYROCKETS=1, PLUMMETS=0}
|   |   DepartureDay = Tuesday: UNCHANGED {UNCHANGED=4, DOWN=0, UP=2, SKYROCKETS=1, PLUMMETS=1}
|   |   DepartureDay = Wednesday: UNCHANGED {UNCHANGED=3, DOWN=1, UP=1, SKYROCKETS=0, PLUMMETS=0}
|   priceUSD = P_64.53: UNCHANGED {UNCHANGED=29, DOWN=19, UP=13, SKYROCKETS=4, PLUMMETS=9}
|   priceUSD = P_67.56: UNCHANGED {UNCHANGED=231, DOWN=2, UP=161, SKYROCKETS=0, PLUMMETS=104}
|   priceUSD = P_69.42
|   |   DifferentDays = (10, 30]: UNCHANGED {UNCHANGED=5, DOWN=1, UP=1, SKYROCKETS=0, PLUMMETS=2}
|   |   DifferentDays = (30, 90]: UNCHANGED {UNCHANGED=4, DOWN=4, UP=0, SKYROCKETS=0, PLUMMETS=4}
|   |   DifferentDays = [0, 10]
|   |   |   DepartureSeason = Spring: UNCHANGED {UNCHANGED=4, DOWN=1, UP=1, SKYROCKETS=0, PLUMMETS=1}
|   |   |   DepartureSeason = Winter: DOWN {UNCHANGED=1, DOWN=11, UP=5, SKYROCKETS=0, PLUMMETS=5}
|   priceUSD = P_74.31
|   |   DepartureSeason = Fall: PLUMMETS {UNCHANGED=22, DOWN=23, UP=0, SKYROCKETS=0, PLUMMETS=23}
|   |   DepartureSeason = Spring: PLUMMETS {UNCHANGED=26, DOWN=20, UP=0, SKYROCKETS=0, PLUMMETS=27}
|   |   DepartureSeason = Summer: PLUMMETS {UNCHANGED=63, DOWN=4, UP=0, SKYROCKETS=0, PLUMMETS=69}
|   |   DepartureSeason = Winter: UNCHANGED {UNCHANGED=21, DOWN=18, UP=0, SKYROCKETS=0, PLUMMETS=13}
|   priceUSD = P_76.44: UNCHANGED {UNCHANGED=3575, DOWN=0, UP=0, SKYROCKETS=967, PLUMMETS=0}
|   priceUSD = P_8.84: UNCHANGED {UNCHANGED=816, DOWN=0, UP=0, SKYROCKETS=651, PLUMMETS=354}
|   priceUSD = P_83.11
|   |   DifferentDays = (10, 30]: UNCHANGED {UNCHANGED=3, DOWN=1, UP=2, SKYROCKETS=0, PLUMMETS=0}
|   |   DifferentDays = (30, 90]: UP {UNCHANGED=25, DOWN=0, UP=26, SKYROCKETS=0, PLUMMETS=0}
|   |   DifferentDays = (90, 150]: UNCHANGED {UNCHANGED=117, DOWN=0, UP=2, SKYROCKETS=0, PLUMMETS=0}
|   |   DifferentDays = [0, 10]: UP {UNCHANGED=3, DOWN=0, UP=3, SKYROCKETS=0, PLUMMETS=0}
|   priceUSD = P_84.09: UNCHANGED {UNCHANGED=2514, DOWN=965, UP=0, SKYROCKETS=0, PLUMMETS=0}
|   priceUSD = P_9.73
|   |   DifferentDays = (10, 30]
|   |   |   DepartureSession = Afternoon: SKYROCKETS {UNCHANGED=4, DOWN=10, UP=0, SKYROCKETS=15, PLUMMETS=2}
|   |   |   DepartureSession = Night: UNCHANGED {UNCHANGED=4, DOWN=0, UP=0, SKYROCKETS=2, PLUMMETS=3}
|   |   DifferentDays = (150, 240]
|   |   |   DepartureDay = Friday: SKYROCKETS {UNCHANGED=2, DOWN=2, UP=0, SKYROCKETS=12, PLUMMETS=9}
|   |   |   DepartureDay = Monday: SKYROCKETS {UNCHANGED=1, DOWN=7, UP=0, SKYROCKETS=14, PLUMMETS=2}
|   |   |   DepartureDay = Saturday: SKYROCKETS {UNCHANGED=1, DOWN=4, UP=0, SKYROCKETS=14, PLUMMETS=7}
|   |   |   DepartureDay = Thursday: PLUMMETS {UNCHANGED=1, DOWN=1, UP=0, SKYROCKETS=0, PLUMMETS=15}
|   |   |   DepartureDay = Tuesday: PLUMMETS {UNCHANGED=4, DOWN=2, UP=0, SKYROCKETS=1, PLUMMETS=15}
|   |   |   DepartureDay = Wednesday: PLUMMETS {UNCHANGED=6, DOWN=4, UP=0, SKYROCKETS=0, PLUMMETS=17}
|   |   DifferentDays = (30, 90]: UNCHANGED {UNCHANGED=111, DOWN=88, UP=0, SKYROCKETS=93, PLUMMETS=37}
|   |   DifferentDays = (90, 150]: DOWN {UNCHANGED=1, DOWN=8, UP=0, SKYROCKETS=6, PLUMMETS=5}
|   |   DifferentDays = >240
|   |   |   DepartureDay = Friday: PLUMMETS {UNCHANGED=3, DOWN=0, UP=0, SKYROCKETS=9, PLUMMETS=11}
|   |   |   DepartureDay = Monday: SKYROCKETS {UNCHANGED=3, DOWN=1, UP=0, SKYROCKETS=10, PLUMMETS=5}
|   |   |   DepartureDay = Saturday: SKYROCKETS {UNCHANGED=0, DOWN=0, UP=0, SKYROCKETS=9, PLUMMETS=5}
|   |   |   DepartureDay = Thursday: PLUMMETS {UNCHANGED=3, DOWN=1, UP=0, SKYROCKETS=3, PLUMMETS=13}
|   |   |   DepartureDay = Tuesday: PLUMMETS {UNCHANGED=2, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=15}
|   |   |   DepartureDay = Wednesday: PLUMMETS {UNCHANGED=4, DOWN=0, UP=0, SKYROCKETS=0, PLUMMETS=15}
|   |   DifferentDays = [0, 10]: UNCHANGED {UNCHANGED=4, DOWN=0, UP=0, SKYROCKETS=2, PLUMMETS=1}
|   priceUSD = P_91.42
|   |   DifferentDays = (10, 30]: DOWN {UNCHANGED=2, DOWN=6, UP=0, SKYROCKETS=0, PLUMMETS=0}
|   |   DifferentDays = (30, 90]: UNCHANGED {UNCHANGED=40, DOWN=21, UP=0, SKYROCKETS=0, PLUMMETS=0}
|   |   DifferentDays = [0, 10]: DOWN {UNCHANGED=2, DOWN=5, UP=0, SKYROCKETS=0, PLUMMETS=0}
provider = VN: UNCHANGED {UNCHANGED=47139, DOWN=1557, UP=1405, SKYROCKETS=2816, PLUMMETS=2434}
```

B Process representation in ACP

In this appendix, we walk the readers through the ACP representation of the rescheduling process. We proceed by pointing out the relevant artifacts of this process: *Flight*, *CustomerSupport* and *Booking*. For each of them, we then list all possible states both informally and formally. There are a range of visual modeling techniques available for informally investigating the states of a given artifact (or an object). A domain expert might employ a UML statechart, finite state machine or Harel state chart to work out the states. Regardless of what modeling technique being employed, a state should be grounded in the attribute domain of the artifact in question.

Table 14: Logical description of all possible states of artifact *CustomerSupport*, which are grounded in predicates over its attributes

$instate(CustomerSupport, initial)$	$= \neg defined(CustomerSupport.customerSupportID) \wedge \neg defined(CustomerSupport.oldBookingID) \wedge \neg defined(CustomerSupport.staffID) \wedge \neg defined(CustomerSupport.supportingType) \wedge \neg defined(CustomerSupport.startTime) \wedge \neg defined(CustomerSupport.finishTime) \wedge \neg defined(CustomerSupport.expectedReschedulingTime) \wedge \neg defined(CustomerSupport.newBookingID) \wedge \neg defined(CustomerSupport.customerDecision)$
$instate(CustomerSupport, opened)$	$= defined(CustomerSupport.customerSupportID) \wedge defined(CustomerSupport.oldBookingID) \wedge defined(CustomerSupport.staffID) \wedge defined(CustomerSupport.supportingType) \wedge defined(CustomerSupport.startTime) \wedge defined(CustomerSupport.expectedReschedulingTime) \wedge \neg defined(CustomerSupport.newBookingID) \wedge \neg defined(CustomerSupport.customerDecision) \wedge \neg defined(CustomerSupport.finishTime)$
$instate(CustomerSupport, confirmedFromCustomer)$	$= defined(CustomerSupport.customerSupportID) \wedge defined(CustomerSupport.oldBookingID) \wedge defined(CustomerSupport.staffID) \wedge defined(CustomerSupport.supportingType) \wedge defined(CustomerSupport.startTime) \wedge defined(CustomerSupport.expectedReschedulingTime) \wedge defined(CustomerSupport.newBookingID) \wedge defined(CustomerSupport.customerDecision) \wedge \neg defined(CustomerSupport.finishTime)$
$instate(CustomerSupport, done)$	$= defined(CustomerSupport.customerSupportID) \wedge defined(CustomerSupport.oldBookingID) \wedge defined(CustomerSupport.staffID) \wedge defined(CustomerSupport.supportingType) \wedge defined(CustomerSupport.startTime) \wedge defined(CustomerSupport.finishTime) \wedge defined(CustomerSupport.expectedReschedulingTime) \wedge defined(CustomerSupport.newBookingID) \wedge defined(CustomerSupport.customerDecision)$

Table 15: Logical description of all possible states of artifact *Flight*, which are grounded in predicates over its attributes

$instate(Flight, initial)$	$= \neg defined(Flight.flightID) \wedge \neg defined(Flight.departurePlace) \wedge \neg defined(Flight.arrivalPlace) \wedge \neg defined(Flight.departureTime) \wedge \neg defined(Flight.flightTime) \wedge \neg defined(Flight.flightDistance) \wedge \neg defined(Flight.airline) \wedge \neg defined(Flight.airplaneType) \wedge \neg defined(Flight.airplaneNumber) \wedge \neg defined(Flight.totalSeats) \wedge \neg defined(Flight.reservedSeats) \wedge \neg defined(Flight.vacantSeats) \wedge currentTime < Flight.departureTime$
$instate(Flight, opened)$	$= defined(Flight.flightID) \wedge defined(Flight.departurePlace) \wedge defined(Flight.arrivalPlace) \wedge defined(Flight.departureTime) \wedge defined(Flight.flightTime) \wedge defined(Flight.flightDistance) \wedge defined(Flight.airline) \wedge defined(Flight.airplaneType) \wedge defined(Flight.airplaneNumber) \wedge defined(Flight.totalSeats) \wedge defined(Flight.reservedSeats) \wedge \neg defined(Flight.vacantSeats) \wedge currentTime < Flight.departureTime \wedge Flight.reservedSeats < Flight.totalSeats$
$instate(Flight, soldout)$	$= defined(Flight.flightID) \wedge defined(Flight.departurePlace) \wedge defined(Flight.arrivalPlace) \wedge defined(Flight.departureTime) \wedge defined(Flight.flightTime) \wedge defined(Flight.flightDistance) \wedge defined(Flight.airline) \wedge defined(Flight.airplaneType) \wedge defined(Flight.airplaneNumber) \wedge defined(Flight.totalSeats) \wedge defined(Flight.reservedSeats) \wedge Flight.vacantSeats = 0 \wedge currentTime < Flight.departureTime \wedge Flight.reservedSeats = Flight.totalSeats$
$instate(Flight, closed)$	$= defined(Flight.flightID) \wedge defined(Flight.departurePlace) \wedge defined(Flight.arrivalPlace) \wedge defined(Flight.departureTime) \wedge defined(Flight.flightTime) \wedge defined(Flight.flightDistance) \wedge defined(Flight.airline) \wedge defined(Flight.airplaneType) \wedge defined(Flight.airplaneNumber) \wedge defined(Flight.totalSeats) \wedge defined(Flight.reservedSeats) \wedge defined(Flight.vacantSeats) \wedge currentTime >= Flight.departureTime$

We rigorously describe the states of *CustomerSupport*, *Flight* and *Booking* in Table 14, Table 15, Table 16, respectively. Note that we ground them in logical predicates that are formulated over the attributes of the said artifacts in an attempt to formally describe their states. Based on this, we should fundamentally be able to produce an ACP model of the rescheduling process as described in Table 17 where the pre- and post-conditions of all the tasks of this process are formulated.

Table 16: Logical description of all possible states of artifact *Booking*, which are grounded in predicates over its attributes

$instate(Booking, initial)$	$= \neg defined(Booking.bookingID) \wedge \neg defined(Booking.customerID) \wedge \neg defined(Booking.flightID) \wedge \neg defined(Booking.staffID) \wedge \neg defined(Booking.ticketStatus) \wedge \neg defined(Booking.serviceClass) \wedge \neg defined(Booking.addOns) \wedge \neg defined(Booking.seatNumber) \wedge \neg defined(Booking.bookingType) \wedge \neg defined(Booking.bookingTime) \wedge \neg defined(Booking.bookingFare) \wedge \neg defined(Booking.paymentType) \wedge \neg defined(Booking.paymentFare) \wedge \neg defined(Booking.paymentTimeFromCustomer) \wedge \neg defined(Booking.paymentTimeToProvider) \wedge \neg defined(Booking.refundsToCustomer) \wedge \neg defined(Booking.timeToRefund) \wedge \neg defined(Booking.timeToCheckIn) \wedge \neg defined(Booking.timeToCancel) \wedge \neg defined(Booking.timeToReschedule) \wedge \neg defined(Booking.cancelFees) \wedge \neg defined(Booking.timeToSendTickets)$
$instate(Booking, booked)$	$= defined(Booking.bookingID) \wedge defined(Booking.customerID) \wedge defined(Booking.flightID) \wedge defined(Booking.staffID) \wedge defined(Booking.ticketStatus) \wedge defined(Booking.serviceClass) \wedge defined(Booking.addOns) \wedge defined(Booking.seatNumber) \wedge defined(Booking.bookingType) \wedge defined(Booking.bookingTime) \wedge defined(Booking.bookingFare) \wedge \neg defined(Booking.paymentType) \wedge \neg defined(Booking.paymentFare) \wedge \neg defined(Booking.paymentTimeFromCustomer) \wedge \neg defined(Booking.paymentTimeToProvider) \wedge \neg defined(Booking.refundsToCustomer) \wedge \neg defined(Booking.timeToRefund) \wedge \neg defined(Booking.timeToCheckIn) \wedge \neg defined(Booking.timeToCancel) \wedge \neg defined(Booking.timeToReschedule) \wedge \neg defined(Booking.cancelFees) \wedge \neg defined(Booking.timeToSendTickets)$
$instate(Booking, paidFromCustomer)$	$= defined(Booking.bookingID) \wedge defined(Booking.customerID) \wedge defined(Booking.flightID) \wedge defined(Booking.staffID) \wedge defined(Booking.ticketStatus) \wedge defined(Booking.serviceClass) \wedge defined(Booking.addOns) \wedge defined(Booking.seatNumber) \wedge defined(Booking.bookingType) \wedge defined(Booking.bookingTime) \wedge defined(Booking.bookingFare) \wedge defined(Booking.paymentType) \wedge defined(Booking.paymentFare) \wedge defined(Booking.paymentTimeFromCustomer) \wedge \neg defined(Booking.paymentTimeToProvider) \wedge \neg defined(Booking.refundsToCustomer) \wedge \neg defined(Booking.timeToRefund) \wedge \neg defined(Booking.timeToCheckIn) \wedge \neg defined(Booking.timeToCancel) \wedge \neg defined(Booking.timeToReschedule) \wedge \neg defined(Booking.cancelFees) \wedge \neg defined(Booking.timeToSendTickets)$
$instate(Booking, paidToProvider)$	$= defined(Booking.bookingID) \wedge defined(Booking.customerID) \wedge defined(Booking.flightID) \wedge defined(Booking.staffID) \wedge defined(Booking.ticketStatus) \wedge defined(Booking.serviceClass) \wedge defined(Booking.addOns) \wedge defined(Booking.seatNumber) \wedge defined(Booking.bookingType) \wedge defined(Booking.bookingTime) \wedge defined(Booking.bookingFare) \wedge defined(Booking.paymentType) \wedge defined(Booking.paymentFare) \wedge defined(Booking.paymentTimeFromCustomer) \wedge defined(Booking.paymentTimeToProvider) \wedge \neg defined(Booking.refundsToCustomer) \wedge \neg defined(Booking.timeToRefund) \wedge \neg defined(Booking.timeToCheckIn) \wedge \neg defined(Booking.timeToCancel) \wedge \neg defined(Booking.timeToReschedule) \wedge \neg defined(Booking.cancelFees) \wedge \neg defined(Booking.timeToSendTickets)$
$instate(Booking, issuedTickets)$	$= defined(Booking.bookingID) \wedge defined(Booking.customerID) \wedge defined(Booking.flightID) \wedge defined(Booking.staffID) \wedge defined(Booking.ticketStatus) \wedge defined(Booking.serviceClass) \wedge defined(Booking.addOns) \wedge defined(Booking.seatNumber) \wedge defined(Booking.bookingType) \wedge defined(Booking.bookingTime) \wedge defined(Booking.bookingFare) \wedge defined(Booking.paymentType) \wedge defined(Booking.paymentFare) \wedge defined(Booking.paymentTimeFromCustomer) \wedge defined(Booking.paymentTimeToProvider) \wedge \neg defined(Booking.refundsToCustomer) \wedge \neg defined(Booking.timeToRefund) \wedge defined(Booking.timeToSendTickets) \wedge \neg defined(Booking.timeToCheckIn) \wedge \neg defined(Booking.timeToCancel) \wedge \neg defined(Booking.timeToReschedule) \wedge \neg defined(Booking.cancelFees)$
$instate(Booking, refunded)$ (only be used for last-minutes tickets)	$= defined(Booking.bookingID) \wedge defined(Booking.customerID) \wedge defined(Booking.flightID) \wedge defined(Booking.staffID) \wedge defined(Booking.ticketStatus) \wedge defined(Booking.serviceClass) \wedge defined(Booking.addOns) \wedge defined(Booking.seatNumber) \wedge defined(Booking.bookingType) \wedge defined(Booking.bookingTime) \wedge defined(Booking.bookingFare) \wedge defined(Booking.paymentType) \wedge defined(Booking.paymentFare) \wedge defined(Booking.paymentTimeFromCustomer) \wedge defined(Booking.paymentTimeToProvider) \wedge defined(Booking.refundsToCustomer) \wedge Booking.refundsToCustomer > 0 \wedge defined(Booking.timeToRefund) \wedge \neg defined(Booking.timeToCheckIn) \wedge \neg defined(Booking.timeToCancel) \wedge \neg defined(Booking.timeToReschedule) \wedge \neg defined(Booking.cancelFees) \wedge defined(Booking.timeToSendTickets)$
$instate(Booking, used)$	$= defined(Booking.bookingID) \wedge defined(Booking.customerID) \wedge defined(Booking.flightID) \wedge defined(Booking.staffID) \wedge defined(Booking.ticketStatus) \wedge defined(Booking.serviceClass) \wedge defined(Booking.addOns) \wedge defined(Booking.seatNumber) \wedge defined(Booking.bookingType) \wedge defined(Booking.bookingTime) \wedge defined(Booking.bookingFare) \wedge defined(Booking.paymentType) \wedge defined(Booking.paymentFare) \wedge defined(Booking.paymentTimeFromCustomer) \wedge defined(Booking.paymentTimeToProvider) \wedge defined(Booking.refundsToCustomer) \wedge defined(Booking.timeToRefund) \wedge defined(Booking.timeToSendTickets) \wedge defined(Booking.timeToCheckIn) \wedge Booking.timeToCheckIn < Flight.departureTime \wedge Booking.flightID = Flight.flightID \wedge \neg defined(Booking.timeToCancel) \wedge \neg defined(Booking.timeToReschedule) \wedge \neg defined(Booking.cancelFees)$
$instate(Booking, expired)$	$= defined(Booking.bookingID) \wedge defined(Booking.customerID) \wedge defined(Booking.flightID) \wedge defined(Booking.staffID) \wedge defined(Booking.ticketStatus) \wedge defined(Booking.serviceClass) \wedge defined(Booking.addOns) \wedge defined(Booking.seatNumber) \wedge defined(Booking.bookingType) \wedge defined(Booking.bookingTime) \wedge defined(Booking.bookingFare) \wedge defined(Booking.paymentType) \wedge defined(Booking.paymentFare) \wedge defined(Booking.paymentTimeFromCustomer) \wedge defined(Booking.paymentTimeToProvider) \wedge defined(Booking.refundsToCustomer) \wedge defined(Booking.timeToRefund) \wedge defined(Booking.timeToSendTickets) \wedge \neg defined(Booking.timeToCheckIn) \wedge \neg defined(Booking.timeToCancel) \wedge \neg defined(Booking.timeToReschedule) \wedge \neg defined(Booking.cancelFees) \wedge currentTime >= Flight.departureTime \wedge Booking.flightID = Flight.flightID$
$instate(Booking, rescheduled)$ (only regular tickets are allowed to be rescheduled)	$= defined(Booking.bookingID) \wedge defined(Booking.customerID) \wedge defined(Booking.flightID) \wedge defined(Booking.staffID) \wedge defined(Booking.ticketStatus) \wedge defined(Booking.serviceClass) \wedge defined(Booking.addOns) \wedge defined(Booking.seatNumber) \wedge defined(Booking.bookingType) \wedge defined(Booking.bookingTime) \wedge defined(Booking.bookingFare) \wedge defined(Booking.paymentType) \wedge defined(Booking.paymentFare) \wedge defined(Booking.paymentTimeFromCustomer) \wedge defined(Booking.paymentTimeToProvider) \wedge \neg defined(Booking.refundsToCustomer) \wedge \neg defined(Booking.timeToRefund) \wedge defined(Booking.timeToSendTickets) \wedge defined(Booking.timeToReschedule) \wedge Booking.timeToReschedule < Flight.departureTime \wedge Booking.newBooking.flightID = Flight.flightID \wedge \neg defined(Booking.timeToCheckIn) \wedge \neg defined(Booking.timeToCancel) \wedge \neg defined(Booking.cancelFees)$
$instate(Booking, canceled)$	$= defined(Booking.bookingID) \wedge defined(Booking.customerID) \wedge defined(Booking.flightID) \wedge defined(Booking.staffID) \wedge defined(Booking.ticketStatus) \wedge defined(Booking.serviceClass) \wedge defined(Booking.addOns) \wedge defined(Booking.seatNumber) \wedge defined(Booking.bookingType) \wedge defined(Booking.bookingTime) \wedge defined(Booking.bookingFare) \wedge defined(Booking.paymentType) \wedge defined(Booking.paymentFare) \wedge defined(Booking.paymentTimeFromCustomer) \wedge defined(Booking.paymentTimeToProvider) \wedge defined(Booking.refundsToCustomer) \wedge defined(Booking.timeToRefund) \wedge defined(Booking.timeToSendTickets) \wedge defined(Booking.timeToCancel) \wedge defined(Booking.cancelFees) \wedge Booking.timeToCancel < Flight.departureTime \wedge Booking.flightID = Flight.flightID \wedge \neg defined(Booking.timeToCheckIn) \wedge \neg defined(Booking.timeToReschedule)$

C Business process measurements

In this appendix, we offer a non-trivial elaboration of the process measurements for the (to-be) selling and rescheduling processes with respect to the following indicators: time, cost and transparency.

Table 17: The rescheduling process expressed in artifact-centric business process modeling

t_1 Send a rescheduling request	
Artifact classes	$Flight, CustomerSupport$
Pre-condition	$instate(Flight, opened)$
Service	$SendAReschedulingRequest(Flight, CustomerSupport)$
Post-condition	$instate(CustomerSupport, initial)$
t_2 Check rescheduling condition	
Artifact classes	$CustomerSupport, Booking$
Pre-condition	$instate(CustomerSupport, initial)$
Service	$CheckReschedulingCondition(CustomerSupport, Booking)$
Post-condition	$instate(CustomerSupport, opened)$
t_3 Notify customer	
Artifact classes	$CustomerSupport$
Pre-condition	$instate(CustomerSupport, opened)$
Service	$NotifyCustomer(CustomerSupport)$
Post-condition	$instate(CustomerSupport, done)$
t_4 Make a list of rescheduling options	
Artifact classes	$Booking, CustomerSupport$
Pre-condition	$instate(Booking, issuedTickets)$
Service	$MakeAListOfReschedulingOptions(Booking, CustomerSupport)$
Post-condition	$instate(Booking, rescheduled) \wedge instate(Booking.newBooking, initial)$
t_5 Pick a rescheduling option or not	
Artifact classes	$CustomerSupport, Booking$
Pre-condition	$instate(CustomerSupport, opened) \wedge instate(Booking, rescheduled) \wedge instate(Booking.newBooking, initial)$
Service	PickAReschedulingOptionOrNot(CustomerSupport, Booking.newBooking)
Post-condition	$instate(CustomerSupport, confirmedFromCustomer) \wedge instate(Booking.newBooking, booked)$
t_6 Create payment instruction	
Artifact classes	$CustomerSupport, Booking$
Pre-condition	$instate(CustomerSupport, confirmedFromCustomer)$
Service	$CreatePaymentInstruction(CustomerSupport, Booking.newBooking)$
Post-condition	$instate(CustomerSupport, done)$
t_7 Make payment	
Artifact classes	$Booking$
Pre-condition	$instate(Booking.newBooking, booked)$
Service	$MakePayment(Booking.newBooking)$
Post-condition	$instate(Booking.newBooking, paidFromCustomer)$
t_8 Pay the Airlines	
Artifact classes	$Booking$
Pre-condition	$instate(Booking.newBooking, paidFromCustomer)$
Service	$PayTheAirlines(Booking.newBooking)$
Post-condition	$instate(Booking.newBooking, paidToProvider)$
t_9 Send the tickets & result of payment	
Artifact classes	$Booking$
Pre-condition	$instate(Booking.newBooking, paidToProvider)$
Service	$SendTheTickets\&ResultOfPayment(Booking.newBooking)$
Post-condition	$instate(Booking.newBooking, issuedTickets)$
t_{10} Refund customer	
Artifact classes	$Booking$
Pre-condition	$instate(Booking.newBooking, issuedTickets) \wedge Booking.newBooking.refundsToCustomer > 0$
Service	$RefundCustomer(Booking.newBooking)$
Post-condition	$instate(Booking.newBooking, refunded)$

C.1 Time

As a recap, we describe a flow analysis technique in Subsection 6.1.1 and present the unit time of each process task in Table 18 and Table 19. We now proceed in calculating the cycle time of the selling process as presented in Table 20 (for its as-is version) and Table 21 (for its to-be version). The cycle time of the rescheduling process is given in Table 22 (for its as-is version) and Table 23 (for its to-be version).

C.2 Cost

As recap, we present activity-based costing theory in Subsection 6.1.2. We work out the activity cost of each task of the selling process and rescheduling process in Table 24 and Table 25, respectively.

Table 18: The unit time measured for the selling process

Task	**Processing Time (minutes)**
Search flights (t_1)	30
Check ticket's status (t_2)	0.5
Notify customer (t_3)	0.5
Submit customer information (t_4)	10
Check eWallet balance (t_5)	0.5
Create payment instruction (t_6)	1
Pay the tickets (t_7)	20
Pay the Airlines (t_8)	5
Update eWallet information (t_9)	1
Send the tickets & result of payment (t_{10})	5
Refund customer (t_{11})	10
Top-up eWallet (t_{12})	5
Warn customer (t_{13})	1

Table 19: The unit time measured for the rescheduling process

Task	**Processing Time (minutes)**
Send a rescheduling request (t_1)	1
Check rescheduling condition (t_2)	0.5
Notify customer (t_3)	1
Make a list of rescheduling options (t_4)	5
Pick a rescheduling option or not (t_5)	10
Create payment instruction (t_6)	1
Make payment (t_7)	15
Pay the Airlines (t_8)	5
Send the tickets & result of payment (t_9)	1
Refund customer (t_10)	10

Table 20: We calculate the cycle time of the (as-is) process of selling tickets

Fragment	**Included Tasks**	**Type of Pattern**	**Cycle Time (minutes)**
1	t_1	Rework	$CT_{f_1} = \frac{ct_{t_1}}{1-r} = \frac{30}{1-0.7} = 100$
2	t_2	Sequential	$CT_{f_2} = ct_{t_2} = 0.5$
3	t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}	XOR-block	$CT_{f_3} = (0.8 * 0 + 0.2 * ct_{f_3}) = 0.2*(CT_{f_{31}} + CT_{f_{32}} + CT_{f_{33}} + CT_{f_{34}} + CT_{f_{35}} + CT_{f_{36}})$
3.1	t_3, t_4, t_5		$CT_{f_{31}} = CT_{f_{311}} + CT_{f_{312}} + CT_{f_{313}}$
3.1.1	t_3	Sequential	$CT_{f_{311}} = ct_{t_3} = 0.5$
3.1.2	t_4	Rework	$CT_{f_{312}} = \frac{ct_{t_4}}{1-r} = \frac{10}{1-0.01} = 10.1$
3.1.3	t_5	Sequential	$CT_{f_{313}} = ct_{t_5} = 0.5$
3.2	t_{12}, t_{13}	Rework	$CT_{f_{32}} = \frac{CT_{f_{321}} + CT_{f_{322}}}{1-r}$
3.2.1	t_{12}	XOR-block	$CT_{f_{321}} = p_{yes}*0+p_{no}*ct_{t_{12}} = 0.7 * 0 + 0.3 * 5 = 1.5$
3.2.2	t_{13}	XOR-block	$CT_{f_{322}} = p_{no}*0+p_{yes}*ct_{t_{13}} = 0.99 * 0 + 0.01 * 1 = 0.01$
3.3	t_6, t_7	Sequential	$CT_{f_{33}} = ct_{t_6} + ct_{t_7} = 1 + 20 = 21$
3.4	t_8, t_9	AND-block	$CT_{f_{34}} = Max(ct_{t_8}, ct_{t_9}) = Max(5, 1) = 5$
3.5	t_{10}	Sequential	$CT_{f_{35}} = ct_{t_{10}} = 5$
3.6	t_{11}	XOR-block	$CT_{f_{36}} = p_{no}*0+p_{yes}*ct_{t_{11}} = 0.4 * 0 + 0.6 * 10 = 6$
Entire Process	t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}		$CT_{Process} = CT_{f_1} + CT_{f_2} + CT_{f_3} = 100 + 0.5 + 0.2 * (0.5 + 10.1 + 0.5 + \frac{1.5+0.01}{1-0.01} + 21 + 5 + 5 + 6) = 110.425$

C.3 Transparency

We articulate four different views that correspond to the as-is version of the selling process as shown in Figure 4, Figure 14, Figure 15 and Figure 16. Similarly, views for the to-be

Table 21: We calculate the cycle time of the (to-be) process of selling tickets

Fragment	Included Tasks	Type of Pattern	Cycle Time (minutes)
1	t_1	Rework	$CT_{f_1} = \frac{ct_{t_1}}{1-r} = \frac{30}{1-0.4} = 50$
2	t_2	Sequential	$CT_{f_2} = ct_{t_2} = 0.5$
3	t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{11}, t_{12}, t_{13}	XOR-block	$CT_{f_3} = (0.8*0 + 0.2*ct_{f_3}) = 0.2*(CT_{f_{31}} + CT_{f_{32}} + CT_{f_{33}} + CT_{f_{34}} + CT_{f_{35}})$
3.1	t_3, t_6, t_4, t_5		$CT_{f_{31}} = CT_{f_{311}} + CT_{f_{312}} + CT_{f_{313}}$
3.1.1	t_3, t_6	Sequential	$CT_{f_{311}} = ct_{t_3} + ct_{t_6} = 0.5 + 1 = 1.5$
3.1.2	t_4	Rework	$CT_{f_{312}} = \frac{ct_{t_4}}{1-r} = \frac{10}{1-0.01} = 10.1$
3.1.3	t_5	Sequential	$CT_{f_{313}} = ct_{t_5} = 0.5$
3.2	t_{12}, t_{13}	Rework	$CT_{f_{32}} = \frac{CT_{f_{321}} + CT_{f_{322}}}{1-r}$
3.2.1	t_{12}	XOR-block	$CT_{f_{321}} = p_{yes}*0 + p_{no}*ct_{t_{12}} = 0.7*0 + 0.3*5 = 1.5$
3.2.2	t_{13}	XOR-block	$CT_{f_{322}} = p_{no}*0 + p_{yes}*ct_{t_{13}} = 0.99*0 + 0.01*1 = 0.01$
3.3	t_7	Sequential	$CT_{f_{33}} = ct_{t_7} = 20$
3.4	t_8, t_9	AND-block	$CT_{f_{34}} = Max(ct_{t_8}, ct_{t_9}) = Max(5, 1) = 5$
3.5	t_{10}	Sequential	$CT_{f_{35}} = ct_{t_{10}} = 5$
Entire Process	t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{12}, t_{13}		$CT_{Process} = CT_{f_1} + CT_{f_2} + CT_{f_3} = 50 + 0.5 + 0.2*(1.5 + 10.1 + 0.5 + \frac{1.5+0.01}{1-0.01} + 20 + 5 + 5) = 59.225$

Table 22: The cycle time of the (as-is) process of rescheduling tickets

Fragment	Included Tasks	Type of Pattern	Cycle Time (minutes)
1	t_1, t_2	Sequential	$CT_{f_1} = ct_{t_1} + ct_{t_2} = 1 + 0.5 = 1.5$
2	t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}	XOR-block	$CT_{f_2} = (0.3*ct_{t_3} + 0.7*CT_{f_{21}}) = (0.3*1 + 0.7*CT_{f_{21}})$
2.1	t_4, t_5, t_6, t_7, t_8, t_9, t_{10}		$CT_{f_{21}} = ct_{f_{211}} + ct_{f_{212}} + ct_{f_{213}} + ct_{f_{214}}$
2.1.1	t_4, t_5	Rework	$CT_{f_{211}} = \frac{ct_{t_4} + ct_{t_5}}{1-r} = \frac{5+10}{1-0.8} = 75$
2.1.2	t_6, t_7, t_8	Sequential	$CT_{f_{212}} = ct_{t_6} + ct_{t_7} + ct_{t_8} = 1 + 15 + 5 = 21$
2.1.3	t_9	Sequential	$CT_{f_{213}} = ct_{t_9} = 1$
2.1.4	t_{10}	XOR-block	$CT_{f_{214}} = p_{no}*0 + p_{yes}*ct_{t_{10}} = 0.4*0 + 0.6*10 = 6$
Entire Process	t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}		$CT_{Process} = CT_{f_1} + CT_{f_2} = 1.5 + (0.3*1 + 0.7*(75 + 21 + 1 + 6)) = 1.5 + 0.3 + 0.7*103 = 73.9$

version of the same process are depicted in Figure 12, Figure 17, Figure 18 and Figure 19. We shall work out the transparent levels of these views using a formula presented previously in Subsection 6.1.5. The transparent levels of these views are presented in Table 26 (for the as-is version) and Table 27 (for the to-be version).

In the same way, we could list all possible views for the rescheduling process. However, we opt to skip this trivial elaboration to go straight ahead with the transparent levels of the rescheduling process, as presented in Table 28 (for the as-is version) and Table 29 (for the to-be version).

Table 23: The cycle time of the (to-be) process of rescheduling tickets

Fragment	Included Tasks	Type of Pattern	Cycle Time (minutes)
1	t_1, t_2	Sequential	$CT_{f_1} = ct_{t_1} + ct_{t_2} = 1 + 0.5 = 1.5$
2	t_3, t_4, t_5, t_6, t_7, t_8, t_9	XOR-block	$CT_{f_2} = (0.3 * ct_{t_3} + 0.7 * CT_{f_{21}}) = (0.3*1+0.7*CT_{f_{21}})$
2.1	t_4, t_5, t_6, t_7, t_8, t_9		$CT_{f_{21}} = ct_{f211} + ct_{f212}$
2.1.1	t_4, t_6, t_5	Rework	$CT_{f_{211}} = \frac{ct_{t_4}+ct_{t_6}+ct_{t_5}}{1-r} = \frac{5+1+10}{1-0.3} = \frac{16}{0.7}$
2.1.2	t_7, t_8, t_9	Sequential	$CT_{f_{212}} = ct_{t_7} + ct_{t_8} + ct_{t_9} = 15 + 5 + 1 = 21$
Entire Process	t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9		$CT_{Process} = CT_{f_1} + CT_{f_2} = 1.5+(0.3*1+0.7*(\frac{16}{0.7}+21)) = 1.5 + 0.3 + 30.7 = 32.5$

Table 24: The unit time and activity cost of process tasks – selling process

Task	Unit Time (minutes /transaction)	Activity Cost (USD/transaction)
Search flights (t_1)	30	2.4
Check ticket's status (t_2)	0.5	0.04
Notify customer (t_3)	0.5	0.04
Submit customer information (t_4)	10	0.8
Check eWallet balance (t_5)	0.5	0.04
Create payment instruction (t_6)	1	0.08
Pay the tickets (t_7)	20	1.6
Pay the Airlines (t_8)	5	0.4
Update eWallet information (t_9)	1	0.08
Send the tickets & result of payment (t_{10})	5	0.4
Refund customer (t_{11})	10	0.8
Top-up eWallet (t_{12})	5	0.4
Warn customer (t_{13})	1	0.08

Table 25: The unit time and activity cost of process tasks – rescheduling process

Task	Unit Time (minutes /transaction)	Activity Cost (USD/transaction)
Send a rescheduling request (t_1)	1	0.08
Check rescheduling condition (t_2)	0.5	0.04
Notify customer (t_3)	1	0.08
Make a list of rescheduling options (t_4)	5	0.4
Pick a rescheduling option or not (t_5)	10	0.8
Create payment instruction (t_6)	1	0.08
Make payment (t_7)	15	1.2
Pay the Airlines (t_8)	5	0.4
Send the tickets & result of payment (t_9)	1	0.08
Refund customer (t_{10})	10	0.8

Table 26: The transparent levels of the as-is process of selling tickets

Index	View	Business Process	Amount of Explicit Tasks	Transparent Level
1	Full view	Figure 4	13	100%
2	Customer service department's view	Figure 14	4	$\frac{4}{13} = 30.8\%$
3	Customer's view	Figure 15	5	$\frac{5}{13} = 38.4\%$
4	Finance department's view	Figure 16	4	$\frac{4}{13} = 30.8\%$

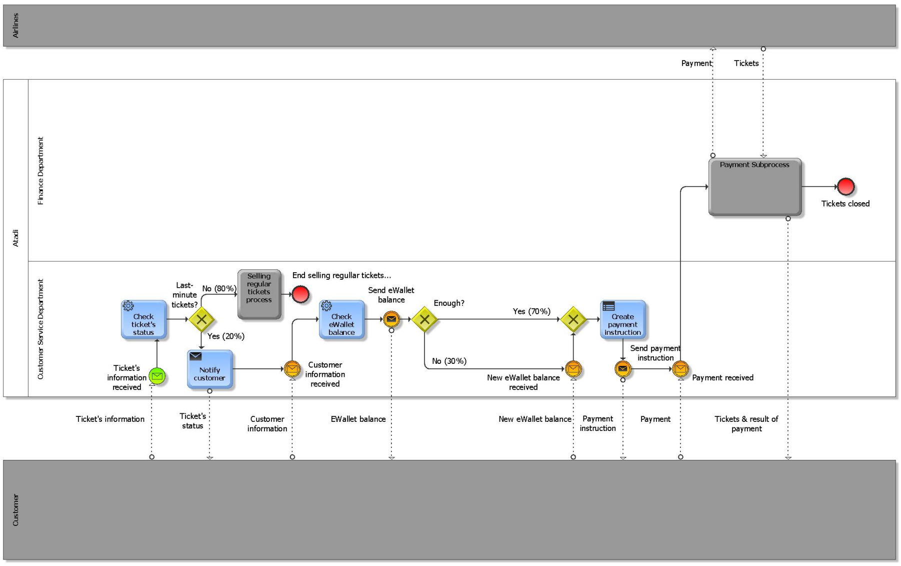


Fig. 14: Customer service department's view of the as-is process of selling tickets

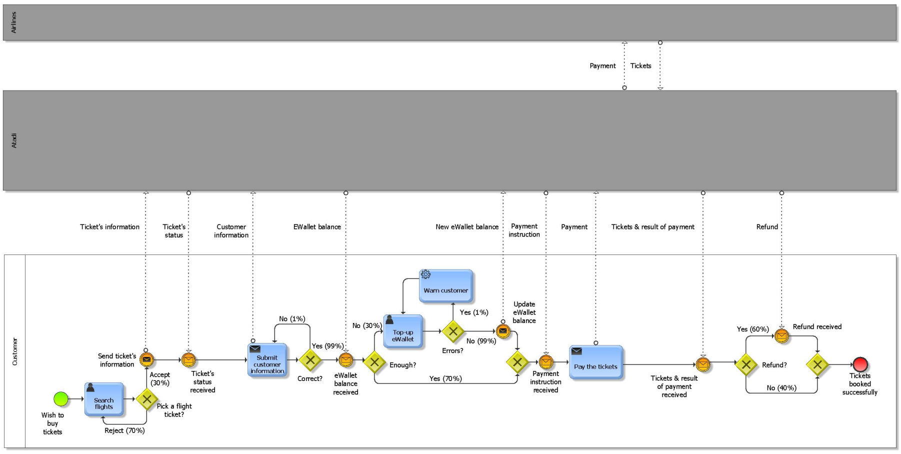


Fig. 15: Customer's view of the as-is process of selling tickets

Table 27: The transparent levels of the to-be process of selling tickets

Index	View	Business Process	Amount of Explicit Tasks	Transparent Level
1	Full view	Figure 12	12	100%
2	Customer service department's view	Figure 17	4	$\frac{4}{12} = 33.3\%$
3	Customer's view	Figure 18	5	$\frac{5}{12} = 41.7\%$
4	Finance department's view	Figure 19	3	$\frac{3}{12} = 25\%$

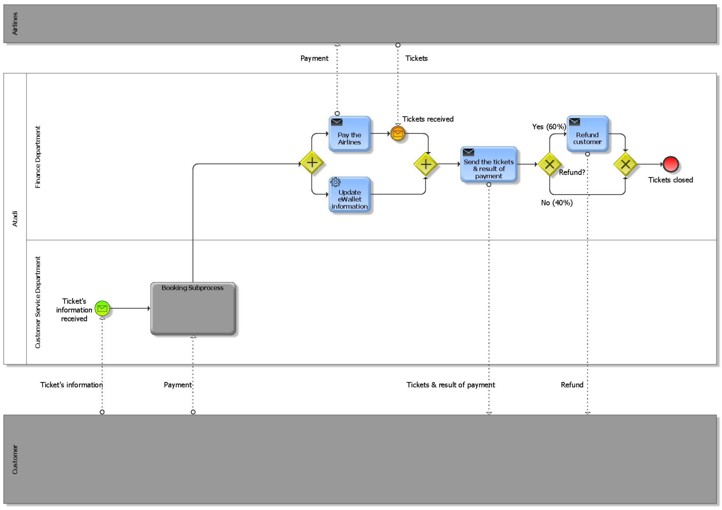


Fig. 16: Finance department's view of the as-is process of selling tickets

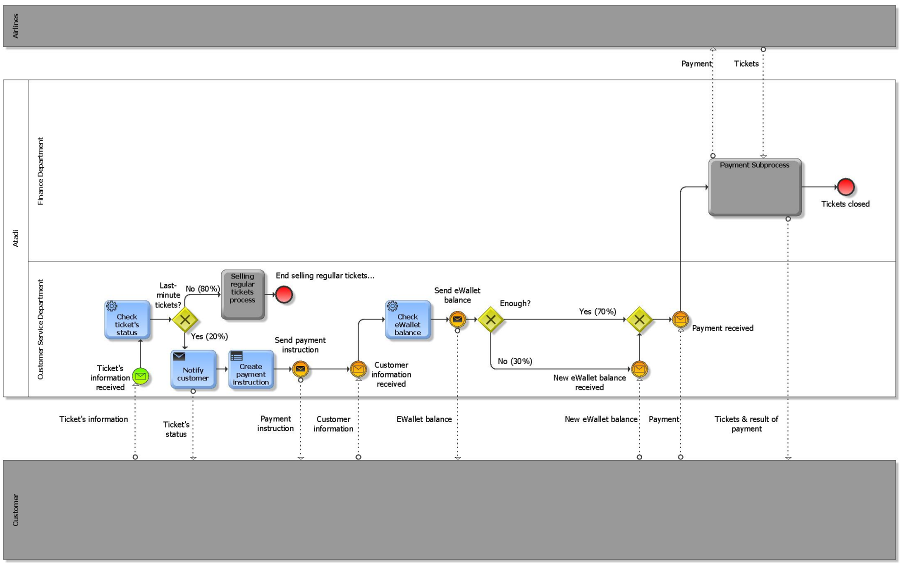


Fig. 17: Customer service department's view of the to-be process of selling tickets

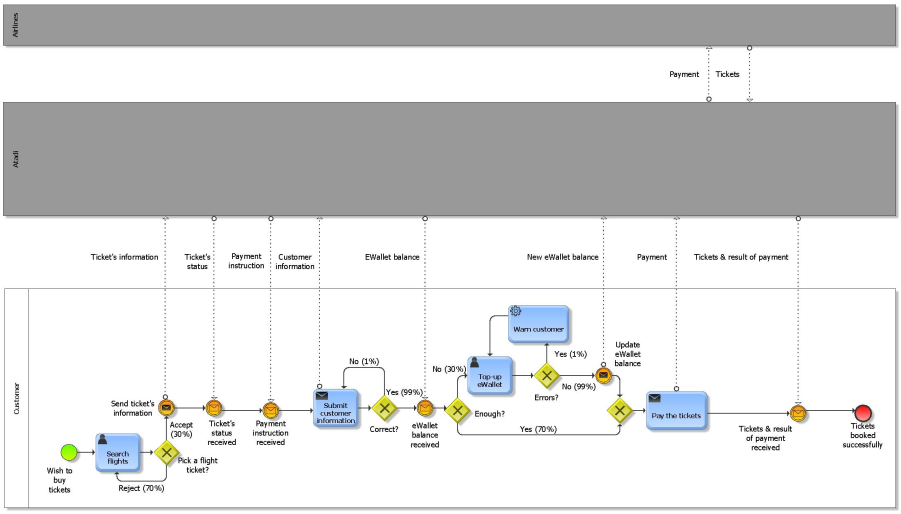


Fig. 18: Customer's view of the to-be process of selling tickets

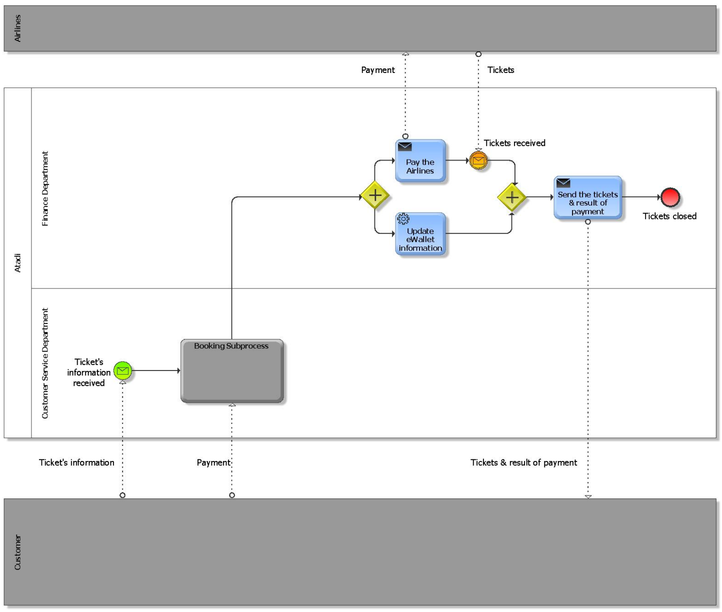


Fig. 19: Finance department's view of the to-be process of selling tickets

Table 28: The transparent levels of the as-is process of rescheduling tickets

Index	View	Transparent Tasks	Amount of Explicit Tasks	Transparent Level
1	Full view	t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}	10	100%
2	Customer service department's view	~~t_1~~, t_2, t_3, t_4, ~~t_5~~, t_6, ~~t_7~~, ~~t_8~~, ~~t_9~~, ~~t_{10}~~	4	$\frac{4}{10} = 40\%$
3	Customer's view	t_1, ~~t_2~~, ~~t_3~~, ~~t_4~~, t_5, ~~t_6~~, t_7, ~~t_8~~, ~~t_9~~, ~~t_{10}~~	3	$\frac{3}{10} = 30\%$
4	Finance department's view	~~t_1~~, ~~t_2~~, ~~t_3~~, ~~t_4~~, ~~t_5~~, ~~t_6~~, ~~t_7~~, t_8, t_9, t_{10}	3	$\frac{3}{10} = 30\%$

Table 29: The transparent levels of the to-be process of rescheduling tickets

Index	View	Transparent Tasks	Amount of Explicit Tasks	Transparent Level
1	Full view	t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9	9	100%
2	Customer service department's view	~~t_1~~, t_2, t_3, t_4, ~~t_5~~, t_6, ~~t_7~~, ~~t_8~~, ~~t_9~~	4	$\frac{4}{9} = 44.5\%$
3	Customer's view	t_1, ~~t_2~~, ~~t_3~~, ~~t_4~~, t_5, ~~t_6~~, t_7, ~~t_8~~, ~~t_9~~	3	$\frac{3}{9} = 33.3\%$
4	Finance department's view	~~t_1~~, ~~t_2~~, ~~t_3~~, ~~t_4~~, ~~t_5~~, ~~t_6~~, ~~t_7~~, t_8, t_9	2	$\frac{2}{9} = 22.2\%$