Tallinn University of Technology

Faculty of Information Technology

Department of Informatics

Chair of Information Systems

Master's thesis

Service Brokering Environment for an Airline

Author: Rein Nõukas, MA

Supervisor: Alexander Norta, PhD

2015

Author's declaration

I hereby declare that this thesis is based on my own work. All ideas, major views and data from different sources by other authors were only used as reference and/or for research purposes. The thesis has not been submitted for any degree or examination in any other university.

Date: 22/05/2015

Signature: *signed digitally*

Acknowledgements

I am very grateful for all the support and intellectual guidance that was given by my academic supervisor, Dr. Alexander Norta, before and during the thesis writing. His patience and empathic approach towards supervision made the whole process a journey of personal growth and mutual respect.

Annotation

In this thesis a collaboration-enabling environment is constructed using the aviation industry as an example. This environment accommodates and facilitates the cooperation of the service consumers, *resp.* airline companies, and external service partners at the business process level. The focus, therefore, is set on the transformation of the airline's organization to become digitalized and developing a supporting platform for service mediation between digitalized organizations of the collaboration environment.

The outcome of the thesis is the emergence of virtual enterprise model, situated in the aviation domain that comprises native components, controlled directly by an airline, and remote components that rely on the mutual dependency instead of control. This multi-party model displays the configurations that enable the intra- and inter-organizational coordination of business processes at software level.

The research in this thesis is backed up with the theoretical approaches covering the concepts of outsourcing, collaborations and Virtual Enterprises. These topics combined with business process modelling and cloud computing provide a basis for innovative business models for redefining the understanding of the B2B transaction configuration models.

As a consequence, it is possible to provide transformation heuristics for reasoning about the complexities of the intra- and inter-organizational transformation in the context of the digital maturation for B2B collaboration. The transformation heuristics provide an aggregated and abstract knowledge about the evolution of the business processes from human-operated workflows inside one organization to automated business transactions between parties of a business ecosystem.

This thesis is about the application of the concept of BPaaS in the context of the aviation industry as a means of proof-of-feasibility. As a consequence, the author proposes the implementation of the B2B collaboration mechanism to be located in the cloud by using the concept of BPaaS. This way, the collaborating partners inside the virtual enterprise – a cluster of autonomous entities with a common goal - can set up loose coupling between their respective business processes enabling automated and autonomous collaboration instances. The aim of loose coupling is to minimize dependencies between system components. This concept is deployed to deal with the requirements of scalability, flexibility and fault tolerance that surface in multi-party configurations.

Commercially, the transition to a cloud-based operations in the context of BPaaS, opens up the possibility to engage into one-to-X relationships. In this kind of relationships the outsourcing process is mediated by a technological solution, *e.g.* a service mediator that establishes terms and conditions per each outsourcing, *resp.* exchange of resources, instance. Furthermore, by dissolving the silo-based architecture of airlines, in the context of joining the separated clusters of the business processes into integrated workflows, the airline can become aware of all of its process states and instances in real time.

The thesis is written in English language and contains 144 pages of text, 6 chapters, 6 figures, 60 tables. Also there are included 8 appendices.

Abstract

The emergence of cheap commodity hardware and global internet coverage enables an extensive usage of cloud-based infrastructure and related services. The new technological advances and economic realities trigger enterprises to seek ways to harness the advantages the cloud-based operations could bring in order to retain and increase competitiveness.

For this purpose, Boeing has presented its view on airline's future - digitalization of the airline's operations in order to enable interconnectedness of operations, resources and services. Digital maturation of an airline enables, in turn, the emergence of a virtual enterprise - a novel way of configuring the execution of the value creating chains inside and between interconnected and -dependent organizations.

This thesis proposes the design for a virtual service brokering environment using the aviation industry as an example. The focus is set on the transformation of an organization (to become digitalized) and developing a supporting platform for service mediation between digitalized organizations. The virtual enterprise model is comprised of native components, controlled directly by an airline, and remote components, that rely on mutual dependency instead of control.

The model of service brokering environment presents the architectural constructs that cover both the collaboration initiation and coordination phases. The model spans across business process, application and infrastructure layers of a service consumer, service mediator and external service provider.

Abbreviations

ACARS - Aircraft Communication Addressing and Reporting System

AOC – Airline Operations Centre

API – Application Programming Interface

B2B – Business-to-Business

BPaaS – Business-Process-as-a-Service

BPDM - Business Process Definition Metamodel

BPMN - Business Process Management Notation

BPO – Business Process Outsourcing

CDN – Content delivery network

DNS – Domain name system

DOO - Day of operations

eSRA - eSourcing Reference Architecture

FMS - Flight Management System

ICT – Information & Communication Technologies

IT – Information Technology

KPI – Key Performance Indicators

PDA - personal digital assistant

SaaS – Software-as-a-Service

SLA – Service Level Agreement

SOA – Service-oriented Architecture

TCO – Total Cost of Ownership

TCT - Transaction Cost Theory

VE – Virtual Enterprise

VM – Virtual Machine

VPC - Virtual Private Cloud

VPN - Virtual Private Network

XML – Extended mark-up language

List of tables

List of figures

Table of Contents

1. Introduction

1.1 Thesis motivation

The aviation industry, at least the part that contains commercially self-dependent companies, is not immune to the trends of modern economics. For decades, the environment in which the companies had to work has been influenced by fast-paced globalization and dissolution of state-granted monopolies. This, in turn, has resulted in new emerging markets and competitors, also the customers of airlines have become increasingly quality-conscious.

The emergence of cheap commodity hardware (storage, servers etc.) and global internet coverage enables an extensive usage of cloud-based infrastructure and related services. The new technological advances and economic realities, f. ex. increasing cost of input resources, trigger enterprises to seek ways to harness the advantages the cloud-based operations could bring in order to retain and increase competitiveness. For this purpose, Boeing has presented its view on airline's future - digitalization of the airline's operations in order to enable interconnectedness of operations, resources and services. Digital maturation of an airline enables, in turn, the emergence of a virtual enterprise - a novel way of configuring the execution of the value creating chains inside and between interconnected and -dependent organizations.

During its life-cycle, each company has to decide whether and what kind of business processes should be conducted in-house or procured from external partners. While only a few marginal business activities were initially outsourced for the sole purpose of improving efficiency and controlling cost, in the late 90-ies, most organizations started to outsourcing entire core company functions, including in some instances core business activities (Campagnolo & Costa, 2006).

In order to make actual use (outsourcing is cheaper and/or with higher quality than producing in-house) of the outsourcing relationships any organization has to devote its resources and attention towards the governance of the outsourcing process. Meaningful governance requires organizations to become internally interconnected allowing the full awareness and control of the data and sequence flows underneath the business processes. With the assistance of complex information systems and platforms the Business-to-Business (B2B) interactions can become extremely streamlined and self-managed, allowing companies to take advantage of and be part of collaboration clusters.

The outsourcing of certain business processes, enabled by the virtual enterprise, is not only an economic decision driven by efficiency but a serious strategic decision driven by new business scenarios. Information and communication technologies (ICT) as a main factor enabling the realization of virtual enterprise allow moving from one-to-one relationships between two entities that regulate their transactions via a contract, including the terms and conditions of the exchange, towards a one-to-X relationship. In the latter the outsourcing process is mediated by a technological solution, labelled as a virtual marketplace, that establishes terms and conditions per each outsourcing (resp. exchange) instance.

This thesis project, therefore, makes an effort to design a collaboration-enabling environment using the aviation industry as an example. As is already mentioned, the focus is set on the transformation of an organization (to become digitalized) and developing a supporting platform for service mediation between digitalized organizations. The virtual enterprise model, therefore, is comprised of native components, controlled directly by an airline, and remote components, that rely on mutual dependency instead of control.

1.2 Theoretical background/Literature overview

In this paragraph a theoretical background and literature overview is given regarding the topic of the thesis project. Also a gap analysis is added in order to understand the current status of using cloud-based B2B solutions in aviation

The author provides a walkthrough that discloses the elements of the collaboration-enabling environment. At first, the meaning of outsourcing is defined using various sources. Secondly, an industry-backed concept of a vertically integrated airline is described. Based on the previous two sections, the nature of B2B collaboration environment is populated with theoretical frameworks. Finally, the collaboration-enabling environment is located in the wider business context by adopting the virtual enterprise paradigm.

1.2.1 Outsourcing

The collaboration between organizations, enacted by outsourcing, is in its simplistic form a desire to seek the most efficient production configurations. Some organizations take up the role of a consumer in order to procure certain goods and services from other organizations that, as producers, produce the aforementioned tangible or intangible assets. In parallel, the same consumer excel in other production areas by creating value there. The created value is exchanged for the goods and services procured from the specialized producers.

The outsourcing phenomenon has been often explained in the literature by means of the Transaction Cost Theory (TCT) (Coase, 1937) (Williamson, 1975) (Williamson, 1979). As is described by Rossignoli et al. (2007), TCT was first developed by Williamson to explain the inconsistency between economic theories and real business practices. Coase (1937), in his article *The Nature of the Firm*, had investigated the issue of company boundaries. Coase's contribution was then expanded by several authors and, with special reference to Information Technology (IT), was re-examined by Picot in 1991. In his work, Coase points out how the use of hierarchy represents an alternative to the action of the market's invisible hand in governing exchanges. According to Coase, the reason for utilizing the hierarchy is related to the need for organizational efficiency.

When the market marginal use costs are excessively high, the management's visible hand (Chandler, 1977) becomes more efficient in activities concerning the coordination of transactions. In the situations where the transaction costs for using outsourcing have become more expensive than those for producing in-house, it is reasonable to change the governance method. This is reasoned by the assumption that producing in-house is more efficient in terms of transaction costs (resp. resources spent on coordination) than in outsourcing as all the production phases are under the direct control of the consumer, thus ignoring any uncertainty that would be appearing during the various transaction phases.

Coase suggested that, on the one hand, enterprises and market are alternative, although complementary, governance methods of transactions and, on the other hand, the methods used to process information affect the comparative efficiency of organizational forms (Ciborra, 1993). Organizational design, following the transactional approach, is therefore connected with the choice of the most efficient form of transactions governance (Grandori, 1984).

As is described by Rossignoli et al. (2007), market and hierarchy represent two opposite ends of the same continuum, inside which different configurations of quasi-market and quasi-organization can be found. From an organizational design standpoint, the problem lies in the identification of an efficient boundary between interdependent organizations for the management of inter-organizational processes or, within the same organization, between different organizational units for the management of intra-organizational processes. The objective is to minimize coordination costs. The choice of the most efficient form of transactions governance therefore is connected with the form that contains both production and transaction costs.

Considering the above, it is extremely relevant to decide what kind of business operations can be executed inside the organization and which processes should be outsourced instead. Moreover, it is all about finding the appropriate balance that regulates the transactions within the hierarchical structures of an organization and, at the opposite end, the market-based structures. Based on this perspective, the purpose of an organization is to mediate the economic transactions between its members or with other organizations (Ulrich & Barney, 1984).

1.2.2 The Digital Airline Maturity Model

Boeing has compiled a digital airline maturity model that uses two perspectives – digital infrastructure and running the business. If the level of digital maturity is low then a majority of activities at the day of operations are conducted manually. Means of ICT, personal computers, digital assistants etc. are used only for data processing or storing and not for decision-making or data sharing. Data is kept in separate silos rather than in the integrated network of systems. Therefore, it is not usable for supporting the operations of an airline as data warehousing and business intelligence applications cannot be implemented.

On the contrary, if the level of digital maturity is mid-high[1] then the majority of activities, e.g., everything that does not require concrete human discretion, are conducted automatically or with the extensive assistance of digital systems. Data are shared among internal and external recipients and processed via the extensive usage of interlinked digital platforms and mediums.

According to Boeing (2014) airlines and their engineering and IT teams must take advantage of the increasing amount of data coming off of airplanes, using advanced analytics and airplane technology to take operational efficiency to the next level. The key to the digital airline is delivering secure, detailed operational and maintenance information to the people who need it most, when they need it most. That means that engineering will share data with IT, but also with the finance, accounting, operational and executive functions. The end result will be airlines

[1] This means having evolved beyond the low maturity level.

that can make the most informed, best possible decisions to maximize their efficiency, profitability and environmental performance.

As Boeing (*ibid*) states, the digital airline has three characteristics:

- Connected in real-time with passengers, maintenance, finance and operations groups, maintenance, repair and overhaul providers and the supply chain, enabling an ever-more-tightly linked airline ecosystem.
- Networked to their airplane assets through complex on-board and ground systems that generate increasingly richer volumes of data for mining by engineering and IT teams.
- Software- and data-driven.

In order to complement the Boeing's definition of the digital airline, it is relevant to define also the meaning of the ecosystem. Ecosystems (Moore, 2006), in the context of this thesis, refer to intentional communities of economic actors whose individual business activities share in some large measure the fate of the whole community.

1.2.3 Collaborations

By having a digitally mature airline that deploys dynamic workflows and the ability to share information in real-time among its networked operational units and external partners it is possible to advance further into optimizing the service acquisition processes. As airlines outsource certain parts of their operations, they are faced with the challenge of setting up efficient collaboration mechanisms between the consuming party and the service provider.

Based on Norta et al.'s (2014) journal paper about the architectural framework for managing dynamic inter-organizational business processes inter-organizational collaboration is defined as a B2B-composed flow of related activities that together create a customer value. This must happen in a dynamic way in that they are formed by the automatic integration of the sub-processes of the involved organizations. Thus, dynamic means that during the setup phase, collaborating organizations find each other by searching business-process marketplaces and match externally contractual-sphere processes of larger in-house processes. By enabling companies to connect their business processes the inter-organizational transactions are conducted in a faster and more cost-effective way with better quality of service.

The discussion in the thesis and the layout of the proposed a collaboration-enabling environment is aligned with Norta et al.'s (*ibid.*) perception on the characteristics of the B2B-collaboration environment according to which eSourcing[2] focuses on structurally harmonizing on an external layer (public views – R. N.) of intra-organizational business processes of a service consuming and one or many service providing organizations into a business collaboration. Important elements of eSourcing are the support of different visibility layers of corporate process details for the collaborating counterpart and flexible mechanisms for service monitoring and information exchange.

1.2.4 Virtual Enterprises

The final step in the configuration of the collaboration-enabling platform is to put it in the wider socio-technical context by adopting the paradigm of the virtual enterprises (VE). Virtual enterprises complement perfectly the nature of B2B-collaborations by showing its commercial potential and universal applicability. According to Camarinha et al. (2001), a virtual enterprise is a temporary alliance of enterprises that come together to share skills or core competencies and resources in order to better respond to business opportunities, and whose cooperation is supported by computer networks.

There are two main characteristics of this definition: cooperation and networking. Basically, the virtual enterprise is a cluster of cooperating enterprises. Certain pre-existing organizations that have shared business goals set up an interdependent and –operable network. This way, these organizations have created a uniform virtual organization without having actually launched any new legal entity or centralized body of control. Therefore, it can be claimed that virtual enterprises come into existence via the integration of skills and assets from donor organizations into a single business entity.

[2] A formal framework for specifying structurally harmonized inter-organizational business process collaborations (Norta & Eshuis, 2010)

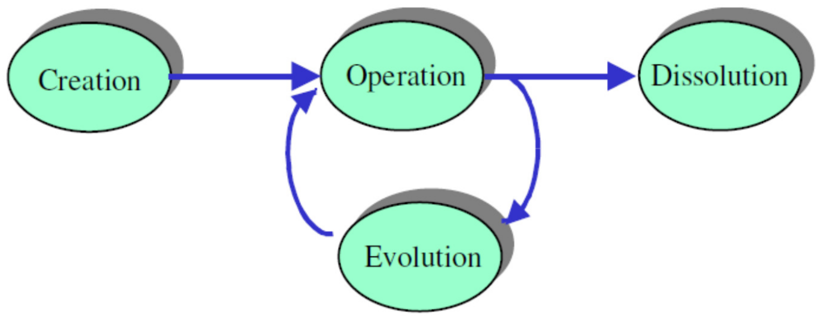


Figure 1. Life-cycle of a Virtual Enterprise. (ibid.)

A Virtual Enterprise evolves along various stages of its life-cycle (see Figure 1 above):

- *Creation*. This is the initial phase when the VE is created/configured and for which some of the major required functionalities are: partner search and selection, contract negotiation, definition of access rights and sharing level, join/leave procedures definition, infrastructure configuration, etc.
- *Operation*. This is the phase when the VE is performing its business process(es) in order to achieve its common goal(s), and which requires functionalities such as: basic secure data exchange mechanisms, information sharing and visibility rights support, orders management, distributed and dynamic planning and scheduling, distributed task management, high levels of task coordination, collaborative engineering support, etc.
- *Evolution* might be necessary during the operation of a VE when it is necessary to add and/or replace a partner, or change roles of partners. This need might be due to some exceptional event, such as temporary incapacity of a partner, changes in the business goal, etc. Functionalities similar to the ones specified for the creation phase are necessary to also be supported here.
- *Dissolution.* This is the phase when the VE finishes its business processes and dismantles itself. Two situations may be the cause for VE dissolution, either the successful achievement of all its goals, or by the decision of involved partners to stop the operation of the VE. The definition of liabilities for all involved partners is an important aspect that needs to be negotiated. For instance, the responsibility of a manufacturer more and more remains during the life cycle of the produced product till its disassembly and recycling. *(ibid.)*

Camarinha et al. (2001) continue by stating that an enterprise may play different roles within a VE during its life cycle. In other words, several kinds of actors can be found in and around a VE environment, acting as:

- *VE Coordinator.* The VE Coordinator is the regulator component of the VE-related activities. The coordinator is either a node specialized in coordination and added to the VE-network, or its role can be played by an already existing VE member. In addition to the VE coordination role, responsible for the global goal, other enterprises may assume the role of coordinators of sub-processes that might be decomposed and performed by a sub-consortium of enterprises.
- *Member Enterprise*. Enterprises with different skills and/or capacities participating in a VE constitute the member enterprise nodes.
- *Network Directory node.* One or more nodes in a network of enterprises may act as the directory nodes. Here the network refers to a general wide area network such as Internet to which a large number of enterprises have access or a closed community of enterprises that establish long-term cooperation plans (industry cluster). Various VEs may coexist in this network and clearly a node in the network may belong to several VEs.
- *Broker/initiator*. This is the role played for instance by a company (not necessarily the VE coordinator) that initiates / creates a VE, plans its business process, and searches for partners.

1.2.5. Current status of using cloud-based B2B-solutions in aviation

This section discusses some of the existing Web 2.0[3] and Service-oriented Architecture[4] (SOA) information systems that are currently in use in the aviation domain, including their shortcomings. On top of aviation-focused socializing networks, like HangarChat[5] and SkyChattr[6], there are some dedicated aviation software vendors and 3rd party developers (see Table 1) that provide their product suites also in a cloud-based Software-as-a-Service (SaaS) form.

[3] Web 2.0 is the current state of online technology as it compares to the early days of the Web, characterized by greater user interactivity and collaboration, more pervasive network connectivity and enhanced communication channels. (TechTarget, 2014)
[4] SOA is an architectural style whose goal is to achieve loose coupling among interacting software agents. A service is a unit of work carried out by a service provider to achieve desired end results for a service consumer. Both service provider and consumer are roles played by software agents on behalf of their owners. (He, 2014)
[5] Link to a site: http://myhangarchat.com/
[6] Link to a site: http://skychattr.com/

Aviation software vendor	Type of software delivery	Products (not exhaustive list)
Sabre Airline Solutions	On-site installation/SaaS	Sabre® AirCentre™ Enterprise Operations
Decisal Ltd	On-site installation/SaaS	SchedulAir
IBS Plc	On-site installation/SaaS	Custom-made solutions for airlines

Table 1. List of aviation software vendors. (Rein Nõukas)

However, none of these SaaS providers has not published a solution that would take SaaS beyond its boundaries. An airline using aforementioned SaaS offering is still limited to a predefined pool of business process management tools developed by the respective SaaS provider. This means that the respective airline in locked-in to the software vendor's product offering and has no options include toolsets from the other vendors into its SaaS-based management platform. Whereas, in Business-Process-as-a-Service (BPaaS) form there could be several SaaS providers' services bundled together into a single operations management platform. This operations management platform would contain respective toolsets to be used by an airline for running its vertically integrated business processes.

Without having a vertically[7] integrated business process management platform the airlines cannot dynamically expose their service acquisition needs (*resp.* public views) to the competing bidders (the external service providers). Instead, inflexible predefined sets of rules are needed to be agreed upon prior engaging into a collaboration (service outsourcing) relationship. Without being aware of each other's public process views both the service consumer and the external service provider are not able to enter into a targeted (automatic) coordination during the enactment of collaboration events. Also the execution of the evaluation of bids (finding the most optimal match to consumer's public view) by the consumer is not automatic, thus making the whole collaboration (initiation) effort dependent on the human component that, in turn, means inefficiently consumed time.

By deploying insufficient and impractical collaboration mechanisms (platform) current SaaS providers in the aviation domain are not enabling airlines to properly 'open themselves up' for partnering companies and thus, preventing the airlines focusing further on cost-cutting and gaining operational excellence.

By having detected a gap in the current service offering regarding the aviation software the author proceeds to discuss the applicable research methodology and propose the appropriate research questions.

[7] Encompassing all the airline's operations – R. N.

1.3. Research methodology and research questions

This chapter is about the chosen research methodology and research questions for this thesis.

At first the research methodology is described in order to understand its underlying logic. It is followed by the guidelines for the research approach that provide more detailed explanation of the application of the research methodology in this thesis. Lastly, a set of research questions are formulated based on the chosen research methodology.

Additionally, the components of the thesis structure are explained.

1.3.1 Design science framework

Given the nature of the researched topic, design-science research in the domain of information systems is followed as a research methodology. According to Hevner, March, Park, & Ram (2004) the design-science paradigm seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artefacts that are broadly defined as constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype systems).

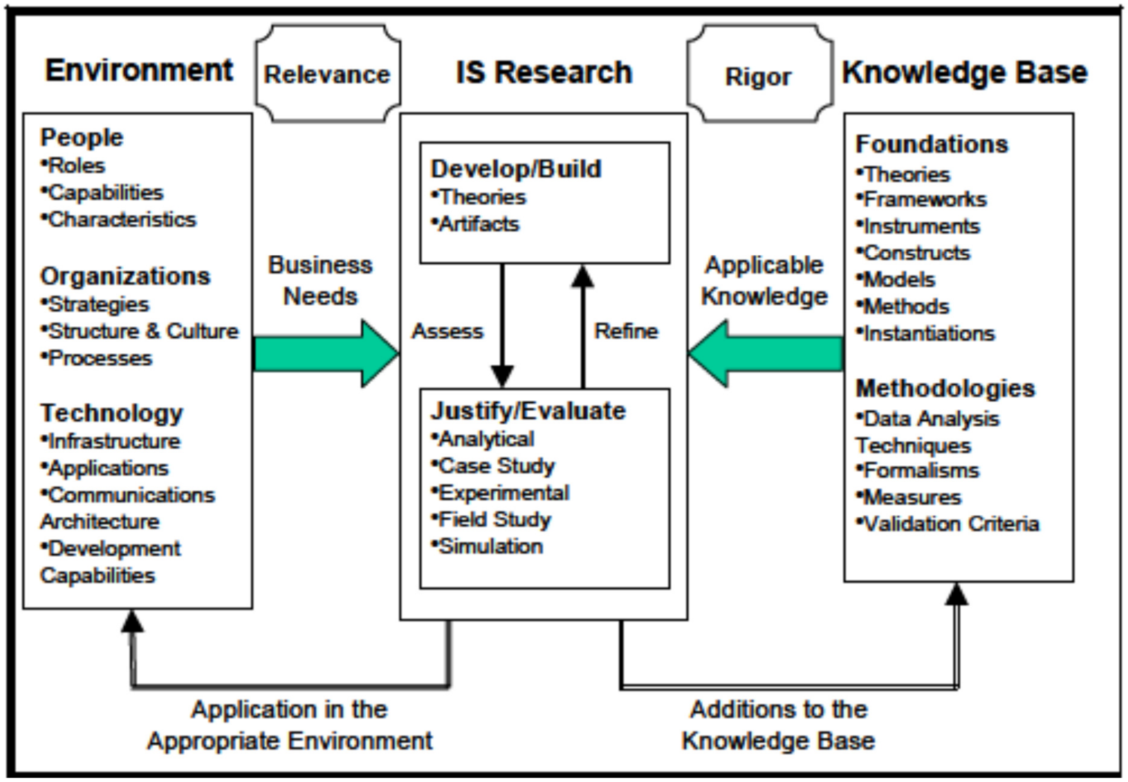


Figure 2. Design-science research framework for the domain of information systems. (Hevner, March, Park, & Ram, 2004)

In Figure 2, the essence of the information-systems research framework is depicted. Norta (2012) explains the contents of the Figure 2 as follows:

- To the left, the environment pillar defines the problem space in which phenomena of interest reside consisting of people, organizations, and technology. Design-science research achieves relevance by building artefacts that address the business needs evolving from the environment.
- To the right of Figure 1, the knowledge base delivers foundations and methodologies from and through which IS research is accomplished. Rigor is achieved by applying foundations in the develop/build-phase and methodologies during the 'justify/evaluate'-phase. The results of design-science research are assessed to the business need in an appropriate environment and contribute to the content of the knowledge base for further research and practice guidelines for conducting design-science research.

As the author aims to realize the Virtual Enterprise concept in a commercial context, it is relevant to differentiate routine design, or system building from design research. Hevner et al. (2004) continues, the difference lies in the nature of the problems and solutions. Routine design is the application of existing knowledge to organizational problems, such as constructing a financial- or marketing- information system using best practice artefacts (constructs, models, methods, and instantiations) existing in the knowledge base. On the other hand, design-science research addresses important unsolved problems in unique, or innovative ways or solved problems in more effective or efficient ways. The key differentiator between routine design and design research is the clear identification of a contribution to the archival knowledge base of foundations and methodologies.

Given the detected gap in the commercial product offering discussed in Chapter 1.2.5. *Current status of using cloud-based B2B solutions in aviation* the author is convinced of the novelty of this thesis' deliverables compared to the existing knowledge base.

1.3.2 Guidelines for the research methodology

For conducting design-science research that adheres to the design science framework, guidelines exist (Hevner, March, Park, & Ram, 2004) that are followed as a research approach in this thesis.

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Table 2. Design-science research guidelines. (Hevner, March, Park, & Ram, 2004)

Below are given the description of how the guidelines in Table 2 are followed in this thesis:

- *Design as an Artefact:*
    - The thesis deliverable (artefact) is a cloud-based collaboration-enabling model for an airline that realizes the Virtual Enterprise concept (see chapter 1.2.4) in a commercial context.
- *Problem Relevance:*
    - The current service offering is missing a BPaaS-solution that covers all operational aspects of an airline. Airlines adopt digitalized workflows but lack ways and means to become truly part of industry-specific ecosystems (resp. virtual enterprises). The general objective of this thesis is to demonstrate a way of how to adopt and apply the Virtual Enterprise concept in the aviation domain. This allows an airline to collaborate with its external service partners in a dynamic and automated way, resulting in a robust and scalable BPaaS-based service platform (see chapter 5.2).
- *Design Evaluation:*
    - The validity of the thesis deliverables (airline-related workflows) are evaluated by the domain experts from the aviation industry. Also the application of the concept of Virtual Enterprises (see chapter 1.2.4) and eSourcing (see chapter 1.2.3) has been verified by one of the active academic researchers (*resp.* Alex Norta, PhD) in this field. Simulations with real-life data have been executed in order to verify the performance ratings of the designed workflow models.
- *Research Contributions:*
    - The thesis contributes to the research made in the application of the concept of eSourcing Reference Architecture (eSRA) discussed by Norta, Grefen and Narendra in 'A reference architecture for managing dynamic inter-organizational business processes' (2014) in the commercial setting. Specifically, the author proposes the new business model, a BPaaS-based service ecosystem for aviation (see chapter 5.2) that reshapes the provision of software products and services to the airlines.
- *Research Rigor:*
    - The author uses several existing design-related (BPMN) frameworks and cloud-enabling technologies (messaging, databases etc.) in the guise of patterns in order to re-configure the business architecture of an airline and to compile a novel ecosystem for mediating collaborations in the aviation domain.

- *Design as a Search Process:*
  - During the development of the cloud-based collaboration-enabling model for an airline several domain experts from an aviation software vendor were utilized to validate the proposed workflow configurations. Also aviation-related source documentation[8] in the form of diagrams, tables and airline-specific technical assessments was extensively used in order to acquire the domain awareness and detect potential gaps in the existing process setups. In general, the aforementioned steps allowed the author to explore the mechanisms that enable actual governance of the seller-buyer relationship in the virtual marketplace (*resp.* governance of outsourcing). The term *governance of outsourcing* is described by IT Governance Institute (2005) as being: 'the set of responsibilities, roles, objectives, interfaces and controls required to anticipate change and manage the introduction, maintenance, performance, costs and control of third-party provided services. It is an active process that the client and service provider must adopt to provide a common, consistent and effective approach that identifies the necessary information, relationships, controls and exchanges among many stakeholders across both parties.'
- *Communication of Research:*
  - As this thesis is written at a master level, it is not an objective to be published for the academic or commercial audiences.

Based on literature on the design-science methodology Peffers et al. (2008) have compiled a design science process flow. The aim of the design science process flow is to design a methodology that would serve as a commonly accepted framework for carrying out research based on design science research principles outlined above.

[8] Due to the confidentiality agreement the referred documents have not been disclosed in this thesis.

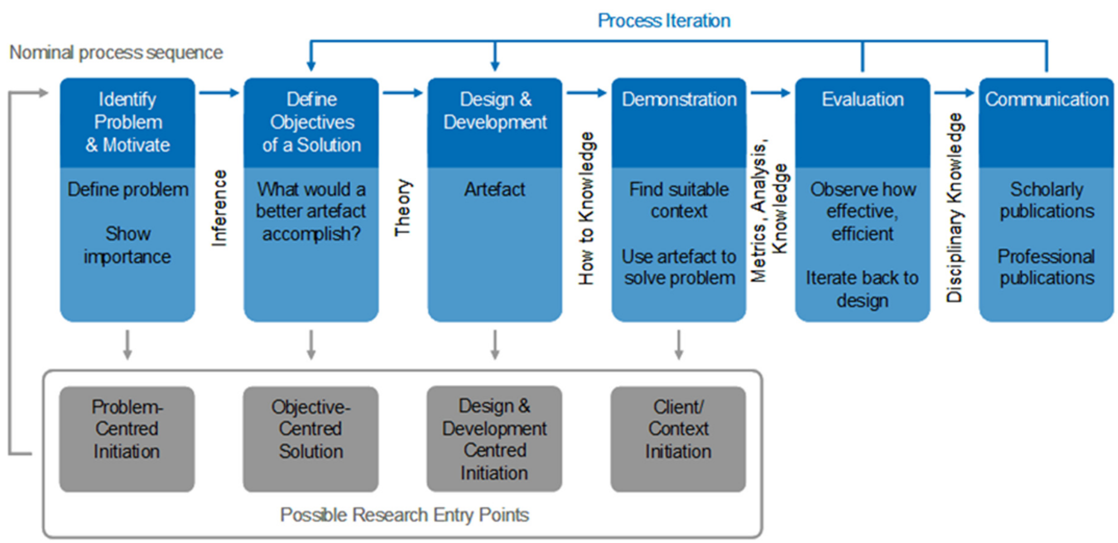


Figure 3. Design science research methodology: process model (Peffers, Tuunanen, Rothenberg, & Chatterjee, 2008)

The result of the synthesis is a process model consisting of six activities in a nominal sequence, which are graphically depicted in Figure 3 (see above). The process flow given in the Figure 3 correlates well with the design-science research guidelines in the Table 2.

The design-science research approach given above is expressed in a concrete form through a number of specific research questions (see chapter 1.3.4 *Research questions*) that result in answers for the research questions.

1.3.4 Research questions

Given the absence of proper service brokering environment for the aviation industry as is revealed in chapter 1.2.5. *Current status of using cloud-based B2B solutions in aviation*, the author proposes the following main research objective:

> *How to configure a cloud-based collaboration-enabling ecosystem for the aviation industry from an airline perspective?*

The abstract research question consists of two sequential topics based on their semantics:

- Preparation of the AS-IS state – focusing on the application of the BPMN 2.0 framework
- Emergence of the TO-BE state – focusing on the cloud and collaboration perspective

Preparation of the AS-IS state includes the following research question and two sub-questions:

- RQ 1: How to normalize and map airline's business processes?
    - rq 1.1: What are the re-built airline's business processes?
    - rq 1.2: What kind of methods can be used to refine the airline's business process model?

Emergence of the TO-BE state includes the following research question and two sub-questions:

- RQ 2: How to increase an airline's digital maturity level?
    - rq 2.1: What is the re-configured airline's digital infrastructure?
    - rq 2.2: What are the re-organized airline's business processes?

The aforementioned research questions are aimed at describing the transformation processes that the author undertakes in order to design a novel configuration of:

- the intra-organizational setup of an airline
- and inter-organizational relationships between entities populating the emergent ecosystem.

The focus is set on the business layer of the business architecture of the organizations in question. In parallel, limited efforts are made in order to describe the design patterns in the application layer and applicable technologies in the infrastructure layer to be used to enact the functionality at the business layer.

1.4 The description of the thesis structure

The remainder of this thesis is structured as follows.

Chapter 2 gives the background for the thesis by defining the state of the art in the cloud computing and business process management.

Chapter 3 presents the preparation of the AS-IS state of the airline's business processes by identifying and refining the airline's business processes.

Chapter 4 takes the results of the previous chapter and uses these for the emergence of the TO-BE state of the airline's digital architecture by introducing a B2B collaboration mechanism. The latter lays foundations for the abstract business, application and infrastructure layers of the participants in the virtual enterprise - an airline, service mediator and external partners.

Chapter 5 discusses the transformation heuristics detected in Chapters 3 and 4. These are relevant in order to understand the process of configuring the cloud-based collaboration-enabling ecosystem for the aviation industry from an airline perspective. Also an effort is made to present a novel business configuration in the context of BPaaS.

Chapter 6 draws conclusions based on the findings and discussion of the previous chapters. The research questions in this thesis are answered. Also, an outlook for further research is provided.

2. The state of the art

In this paragraph the author provides an overview of cloud computing by investigating the service models and cloud types. Cloud computing, in its simplest definition, enables to process and store data in the location-independent context. This way an enterprise can have a virtually unlimited computing capacity to run its operations.

Additionally, the basics of business process modelling and notation (BPMN) are presented in the form of choreography and orchestration. Both are part of the BPMN 2.0 framework. Choreography enables to detect and specify the pairs of peers, autonomous entities, and to understand their informational needs for becoming and staying aware of themselves and each other. Orchestration is more focused at the order of tasks which is maintained by the sequence and data flows. Both viewpoints are equally relevant as they enable to notice and analyze potential behavioral patterns and opportunities for workflow re-design and optimization without altering the underlying business logic.

Both cloud computing and BPMN concepts are relevant for understanding the research and development efforts that are presented in Chapters 4 and 5.

2.1 Cloud Computing

Cloud computing is an emergent set of technologies that is already changing the nature of methods for data-processing and thus, operations management in the commercial setting. Cloud computing is an actively researched topic both by commercial and academic organizations seeking for ways to harness the potential of the clouds. Many commercial giants like Google[9], Microsoft[10] and Amazon[11] offer cloud computing solutions of varying kind.

The goal of this section is to introduce cloud computing using several perspectives and viewpoints. At first, the general benefits and drawbacks of cloud computing are explained briefly. Next, the abstract cloud architecture and three common service models are introduced. After that, four different cloud types are presented.

[9] Google Cloud Platform. Link to a site: https://cloud.google.com/
[10] Microsoft Azure. Link to a site: http://azure.microsoft.com/en-us/
[11] Amazon Web Services. Link to a site: http://aws.amazon.com/

2.1.1 General benefits and drawbacks

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. (Mell & Grance, 2011)

The main benefits of the cloud computing are discussed by Zhang et al. (2010):

- **No up-front investment:** Cloud computing uses a pay-as-you-go pricing model. A service provider does not need to invest in the infrastructure to start gaining benefit from cloud computing. It simply rents resources from the cloud according to its own needs and pay for the usage.
- **Lowering operating cost:** Resources in a cloud environment can be rapidly allocated and de-allocated on demand. Hence, a service provider no longer needs to provision capacities according to the peak load. This provides huge savings since resources can be released to save on operating costs when service demand is low.
- **Highly scalable:** Infrastructure providers pool large amount of resources from data centres and make them easily accessible. A service provider can easily expand its service to large scales in order to handle rapid increase in service demands (e.g., flash-crowd effect).
- **Easy access:** Services hosted in the cloud are generally web-based. Therefore, they are easily accessible through a variety of devices with Internet connections. These devices not only include desktop and laptop computers, but also cell phones and personal digital assistants (PDAs).
- **Reducing business risks and maintenance expenses:** By outsourcing the service infrastructure to the clouds, a service provider shifts its business risks (such as hardware failures) to infrastructure providers, who often have better expertise and are better equipped for managing these risks. In addition, a service provider can cut down the hardware maintenance and the staff training costs.

There are also drawbacks and threats in using cloud computing (Duipmans, 2012):

- **Security:** Data is stored inside the cloud and accessible through the Internet. In several situations cloud users deal with confidential information that should be kept inside the cloud user's organization. In these situations cloud computing might not be a good solution, although there are solutions[12] with cloud computing in which data is stored inside the cloud user's organization but applications are hosted in the cloud. There are also technical solutions[13] for making data unintelligible for unauthorized people, for example, by using encryption algorithms.

- **Availability:** Clouds are accessible through the Internet. This gives cloud users the freedom to work with the services wherever they have an Internet connection. The downside is that when the Internet connection fails, for example, on the side of the cloud provider, cloud users are not able to access their services any more. This might lead to business failures, especially when the services are part of a business process.
- **Data transfer bottlenecks:** Users that use software systems might need to transfer large amounts of data in order to use the system. Data should be transported not only from the user to the system, but also to multiple systems in order to cooperate inside a company. Cloud computing providers do not only bill the computation and storage services, but also data transportation is measured and billed. For companies that deal with a lot of data, cloud computing may be expensive because of the data transportation costs. Another problem can be the time it takes to transfer data to the cloud. For example, suppose that a company needs to upload a huge amount of data in order to perform a complex computation, in which case the data transfer may take more time than the computation itself. In these situations it might be faster and cheaper to perform the computation inside the premises of the cloud user.

2.1.2 Cloud architecture

In general, the architecture of a cloud computing environment can be divided into 4 layers: the hardware/ datacentre layer, the infrastructure layer, the platform layer and the application layer, as shown in Figure 4 (see below).

The layers are described as follows (Zhang, Cheng, & Boutaba, 2010):

- **The hardware layer:** This layer is responsible for managing the physical resources of the cloud, including physical servers, routers, switches, power and cooling systems. In

[12] Refers to a hybrid cloud service that is a combination of a private cloud and a public cloud. It enables an organization to keep some of its operations in-house (private cloud) while also utilizing a cloud service from an outside provider for its other operations (public cloud). (Aerohive Networks Inc., 2013)
[13] CipherCloud Searchable Strong Encryption. Link to a site: http://www.ciphercloud.com/technologies/encryption/

practice, the hardware layer is typically implemented in data centres. A data centre usually contains thousands of servers that are organized in racks and interconnected through switches, routers or other fabrics. Typical issues at the hardware layer include hardware configuration, fault tolerance, traffic management, power and cooling resource management.

- **The infrastructure layer:** Also known as the virtualization layer, the infrastructure layer creates a pool of storage and computing resources by partitioning the physical resources using virtualization technologies such as Xen[14], KVM[15] and VMware[16]. The infrastructure layer is an essential component of cloud computing, since many key features, such as dynamic resource assignment, are only made available through virtualization technologies.
- **The platform layer:** Built on top of the infrastructure layer, the platform layer consists of operating systems and application frameworks. The purpose of the platform layer is to minimize the burden of deploying applications directly into Virtual Machine (VM) containers. For example, Google App Engine[17] operates at the platform layer to provide Application Programming Interface (API) support for implementing storage, database and business logic of typical web applications.
- **The application layer:** At the highest level of the hierarchy, the application layer consists of the actual cloud applications. Different from traditional applications, cloud applications can leverage the automatic-scaling feature to achieve better performance, availability and lower operating cost.

[14]Link to a site: http://www.citrix.com/products/xenserver/overview.html
[15] Link to a site: http://www.linux-kvm.org/page/Main_Page
[16] Link to a site: http://www.vmware.com/products/esxi-and-esx/overview.html
[17] Link to a site: https://cloud.google.com/appengine/docs/whatisgoogleappengine

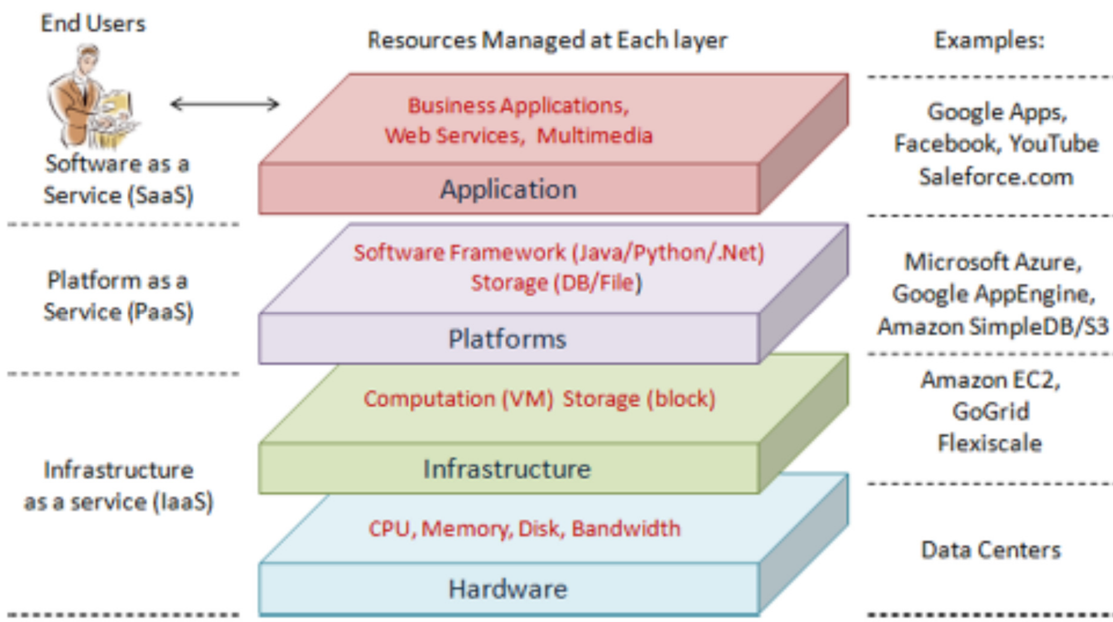


Figure 4. Cloud computing architecture. (ibid.)

Compared to traditional service hosting environments such as dedicated server farms, the architecture of cloud computing is modular. Each layer is loosely coupled with the layers above and below, allowing each layer to evolve separately. The architectural modularity allows cloud computing to support a wide range of application requirements while reducing management and maintenance overhead.

2.1.3 Service models

The National Institute of Standards and Technology (Mell & Grance, 2011) identifies three service models for cloud computing: Software-as-a-Service, Platform-as-a-Service and Infrastructure-as-a-Service (see also Figure 3 for visual description).

The models are described as follows:

- **Software as a Service (SaaS).** The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

- **Platform as a Service (PaaS).** The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.
- **Infrastructure as a Service (IaaS).** The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, like operating systems and database applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

Additionally, the IBM Cloud Computing Reference Architecture 2.0 (IBM, 2011) populates the fourth cloud-based service model - **Business-Process as a Service (BPaaS)** - as follows:

Business process services are any business process (horizontal or vertical) delivered through the cloud service model (multi-tenant, self-service provisioning, elastic scaling and usage metering or pricing) via the Internet with access via web centric interfaces and exploiting web-oriented cloud architecture.

2.1.4 Cloud types

There are many issues to consider when moving an enterprise application to the cloud environment. For example, some service providers are mostly interested in lowering operation cost, while others may prefer high reliability and security. Accordingly, there are different types of clouds, each with its own benefits and drawbacks (Zhang, Cheng, & Boutaba, 2010):

- **Public clouds:** A cloud in which service providers offer their resources as services to the general public. Public clouds offer several key benefits to service providers, including no initial capital investment on infrastructure and shifting of risks to infrastructure providers. However, public clouds lack fine-grained control over data, network and security settings, which hampers their effectiveness in many business scenarios.

- **Private clouds:** Also known as internal clouds, private clouds are designed for exclusive use by a single organization. A private cloud may be built and managed by the organization or by external providers. A private cloud offers the highest degree of control over performance, reliability and security. However, they are similar, by being located on-premise, to traditional proprietary server farms and do not provide benefits such as no up-front capital costs. According to Aerohive (2013), more customer control equals fewer security worries for private cloud service users. By moving traditional hardware-based old IT system over to the cloud, the customer is still able to enjoy the benefits of scalability, flexibility and higher productivity, but will be able to do so without sacrificing any of the accountability for data security that may sometimes be viewed with a public cloud service.
- **Hybrid clouds:** A hybrid cloud is a combination of public and private cloud that address the limitations (see in the paragraphs above) of each approach. In a hybrid cloud, part of the service infrastructure runs in private clouds while the remaining part runs in public clouds. Hybrid clouds provide tighter control and security over application data compared to public clouds, while still facilitating on-demand service expansion and contraction. According to Aerohive (2013), it enables an organization to keep some of its operations in-house (similar to a private cloud) while also utilizing a cloud service from an outside provider for its other operations (similar to a public cloud).

  In parallel, designing a hybrid cloud requires carefully determining the best split between public and private cloud components. As Breiter and Naik (2013) explain, the experience from both end user and administration point of view needs to be seamless in order to keep the hybrid cloud both functional and useable. This means heterogeneity across component clouds needs to be masked by homogeneous uniform portals, dashboards, and other management controls. It also means users and workloads from any component of the hybrid cloud are subject to the same policies, controls, and governance across the entire hybrid cloud wherever the services they are using are delivered from or workloads are processed.
- **Virtual Private Cloud:** An alternative solution to addressing the limitations of both public and private clouds is called Virtual Private Cloud (VPC). A VPC is an extra platform on top of public clouds. The main difference is that a VPC leverages virtual

private network (VPN) technology that allows service providers to design their own topology and security settings such as firewall rules. VPC virtualizes servers, applications and also the underlying communication network as well. Additionally, VPC provides seamless transition from a proprietary service infrastructure to a cloud-based infrastructure, owing to the virtualized network layer.

Having introduced the cloud computing, it is time to move forward and discuss the basics of the business process management. The latter constitutes the second pillar of the proposed collaboration-enabling environment.

2.2 Business process management

Business process management aims to identify the internal business processes of an organization, capture the detected processes in process models at varying abstraction levels, manage and optimize these processes by monitoring and reviewing them.

According to Weske (2007), business process management is based on the observation that each product that a company provides to the market is the outcome of a number of activities performed. Business processes are the key instrument for organizing these activities and to improve the understanding of their interrelationships. Information technology and information systems play an important role in business process management, because activities that a company performs are supported by information systems. Business process activities can be performed by a company's employees manually, or with the help of information systems (*resp.* semi-automatically). Business process activities can also be enacted automatically by information systems, without any human involvement. A company can reach its business goals in an efficient and effective manner only if people and other enterprise resources, such as information systems, are configured as collaborations. Business processes are an important concept to facilitating this effective collaboration.

Techtarget.com (Rouse, 2011) explains that business process management is a point of connection within a company between the line-of-business (*resp.* revenue generating activities – R. N.) and the IT department (*resp.* business IT analysis – R. N.). Business Process Management Notation (BPMN) was created to facilitate the communication between the development actors and the business owners. The Object Management Group (2011) confirms this deduction by stating that the primary goal of BPMN is to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes. Thus, BPMN creates a standardized bridge for the gap between the business process design and process implementation.

By having defined the relevance of the BPMN in understanding the business processes taking place in an organization the following subchapters describe the details of the BPMN using two viewpoints: choreography and orchestration.

2.2.1 Basics of choreography modelling

As the airline operations involve a constant exchange of information between various entities, the author models the first iteration of the AS-IS situation (see Appendix A) using choreography. This way, the complexity of having a number of activity diagrams in the initial AS-IS model, provided by the software vendor, is reduced to abstract workflows while retaining the core logic of relationships and dependencies.

Choreography modelling was incorporated to the BPMN framework by the Object Management Group (2011) in 2011 when the specification for BPMN 2.0[18] was released. A choreography is a type of process, but differs in purpose and behavior from a standard BPMN process. A standard process, or an orchestration process, defines the flow of activities of a specific business entity. In contrast, choreography formalizes the way business entities, labelled as participants, coordinate their interactions. The focus is not on orchestrations of the work performed within these participants, but rather on the exchange of information, in the form of messages, between these participants.

The content of the following discussion in this chapter is based on the aforementioned document.

[18] It represents the amalgamation of best practices within the business modeling community to define the notation and semantics of collaboration diagrams, process diagrams, and choreography diagrams. (OMG, 2011)

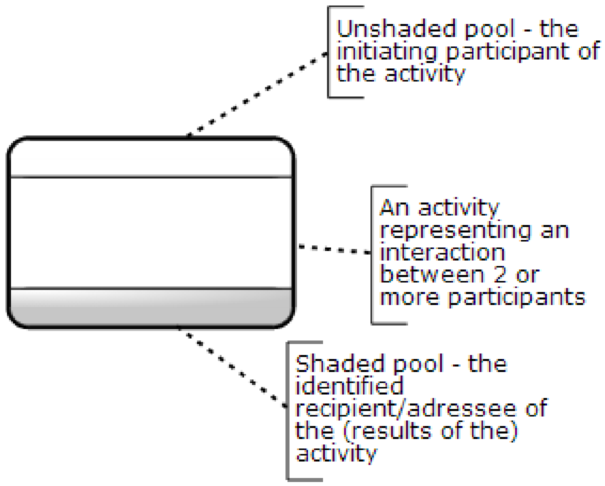


Figure 5. The depiction of the basic elements of choreography. (Rein Nõukas)

Above (see Figure 5) is given a visual depiction of the basic elements of choreography. A key to understanding choreography and how they are used in BPMN is their relationship to pools. A pool is the graphical representation of a participant in a collaboration. A collaboration depicts the interactions between two or more participants. While a standard process exists within a pool, a choreography exists between pools.

A choreography is a different kind of process. A choreography defines the sequence of interactions between participants. Thus, a choreography does not exist in a single pool—it is not the purview of a single participant. Each step in the choreography involves two or more participants (these steps are called choreography activities). Activities are interactions that represent a set (1 or more) of message exchanges, which involves two or more Participants. This means that the choreography, in BPMN terms, is defined outside of any particular pool.

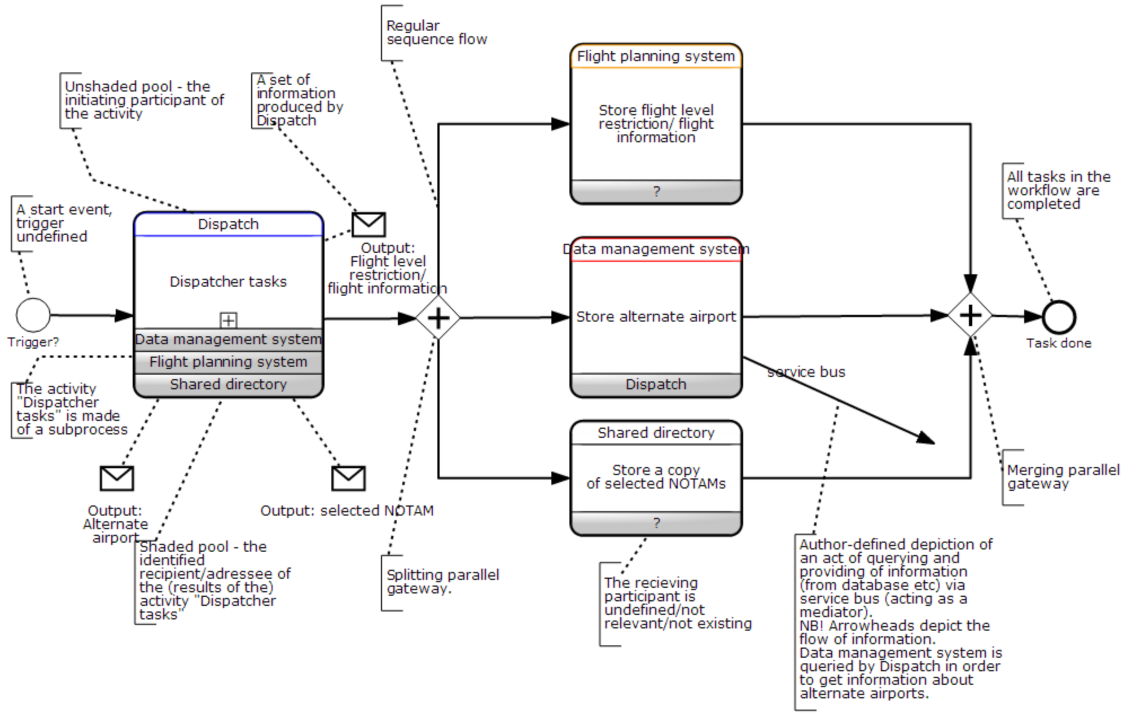


Figure 6. Dispatcher operation subprocess. (Rein Nõukas)

In Figure 6 the actual *Dispatcher operation subprocess* is used as a sample of the usage of BPMN 2.0 notation in representing choreography. The same rules apply to choreography models as to regular activity-based models.

Only major exception is the fact that neither data objects nor repositories are used in choreography. A choreography does not have a central control mechanism and there is no mechanism for maintaining any central process (choreography) data. Without centrally-provided data that is understood by all the participants conditional or assignment expressions, like if-then constructs, are impossible to maintain.

2.2.2 Orchestration versus choreography

Both orchestration and choreography are part of BPMN 2.0 framework and are subsets of a collaboration. In the Introduction to BPMN (White S. A., 2009) the collaborative process depicts the interactions between two or more business entities. The diagrams for these types of processes are generally from a global point of view. That is, they do not take the view of any particular participant, but show the interactions between the participants. The interactions are depicted as a sequence of activities and the message exchange patterns between the participants.

The activities for the collaboration participants can be considered the 'touch-points' between the participants; thus, the process defines the interactions that are visible to the public for each participant.

The author considers the definition above applying to both choreography and orchestration models. In turn, this means the both modelling perspectives are just representations of a single Business Process Definition Metamodel[19] attributable to airline at the day of operations.

Partially the definition of orchestration is covered already in the aforementioned definition of choreography (see Chapter 2.2.1 *Basics of choreography modelling*). As per the Object Management Group (2011) an orchestration process describes a sequence or flow of activities in an organization with the objective of carrying out work and achieving a common business goal. In BPMN, a process is depicted as a graph of flow elements, which are a set of activities, events, gateways, and sequence flows that define finite execution semantics. The flow elements are further elaborated in Chapter 2.2.3 *Specifics of orchestration notation.*

In order to populate the definitions of both orchestration and choreography with more practical descriptions the author presents some of ideas originating from the business analysis community:

- Orchestration is a term used to describe the traditional notion of managed or directed execution, where activities are carried out, with branching and synchronization of different threads. Choreography is a more abstract notion of process (for most people). It is used to describe the 'Interactions' of collaborating entities, each of which may have their own internal orchestration processes. (Miers, 2006)
- According to Ward-Dutton (2011), orchestration is useful if all actors, that take part of a process, are controlled by one entity. This entity, therefore, is able to dictate the flow of activities. This kind of configuration exists in cases of intra-organizational setups. Choreography is a way of specifying how two or more parties - none of which has any control over the other parties' processes, or perhaps any visibility of those processes - can coordinate their activities and processes to share information and value. Choreography becomes relevant when coordination across domains of control/visibility is required.

[19] The Business Process Definition Metamodel (BPDM) is a standard definition of concepts used to express business process models (a metamodel), adopted by the OMG (Object Management Group). Metamodels define concepts, relationships, and semantics for exchange of user models between different modeling tools. (Wikipedia, 2013)

Choreography can be considered to act, in a simple scenario, as like a network protocol. It dictates acceptable patterns of requests and responses between parties.

- Orchestration happens when a central coordinator takes care of deciding the execution sequence. Choreography happens when all participants are aware of their roles in the process, and are ready to perform the prescribed tasks upon getting the appropriate stimulus. In a web services environment, Orchestration is an executable workflow of invocations of services, and Choreography is a set of rules that every service complies with and exposes, so that others can interact with him. In BPM, one can apply the above concept where instead of services tasks and human/system executors are present. (Brambilla, 2011)

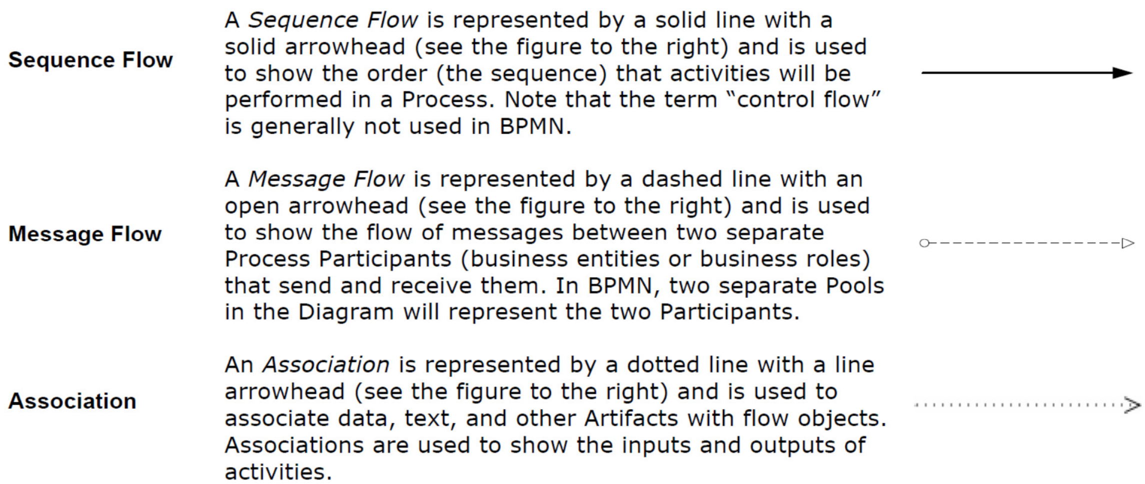
Due to the fairly emergent nature of the evolution of the business process modelling the definitions are yet to solidify therefore it is not possible to provide clear-cut meanings of both orchestration and choreography. As was expressed by the community members, orchestrations are used to configure processes inside an entity, choreography, in turn, is used to apply patterned coordination mechanisms at the inter-entity level. Open-ended definitions enable to utilize them with a relative flexibility that in turn contributes to the main goal, analysis and modelling of cross-organizational airline workflows, of this theses.

In the running case, using choreography enables to detect and specify the pairs of peers, autonomous entities, and to understand their informational needs for becoming and staying aware of themselves and each other. Orchestration is more focused at the order of tasks which is maintained by the sequence and data flows. Both viewpoints are equally relevant as it enables to notice and analyze potential behavioral patterns and opportunities for workflow re-design and optimization without altering the underlying business logic.

2.2.3 Specifics of orchestration notation

The second iteration of the normalized AS-IS model (see Chapter 3) and the TO-BE model (see Chapter 4) are compiled using the notation formalized in the specification for BPMN 2.0. As orchestration is basically a standard BPMN process then it is assumed the audience of this thesis is familiar to the BPMN notation. However, in order to maintain the ease of readability of the orchestration model some of the most relevant part of the notation are described in this chapter.

As the focus is set at the order of execution of activities in orchestration, data flows and workflows have to be depicted in an accurate and uniform way. For this purpose – to link the elements comprising of the data flows and workflows, connecting objects are deployed. They act as links between the flow objects[20] in a diagram to create the basic skeletal structure of a business process. The Figure 7 depicts three connecting objects that provide this function.



Sequence Flow	A *Sequence Flow* is represented by a solid line with a solid arrowhead (see the figure to the right) and is used to show the order (the sequence) that activities will be performed in a Process. Note that the term "control flow" is generally not used in BPMN.
Message Flow	A *Message Flow* is represented by a dashed line with an open arrowhead (see the figure to the right) and is used to show the flow of messages between two separate Process Participants (business entities or business roles) that send and receive them. In BPMN, two separate Pools in the Diagram will represent the two Participants.
Association	An *Association* is represented by a dotted line with a line arrowhead (see the figure to the right) and is used to associate data, text, and other Artifacts with flow objects. Associations are used to show the inputs and outputs of activities.

Figure 7. BPMN connecting objects. (White S. A., 2009)

Major emphasis has also been placed on the events whilst compiling orchestration model. There can be either start, intermediate or end events. Figure 8 depicts the full typology of notation used to express events.

[20] There are three types of Flow Objects: Event, Gateway and Activity. (White S. A., 2009)

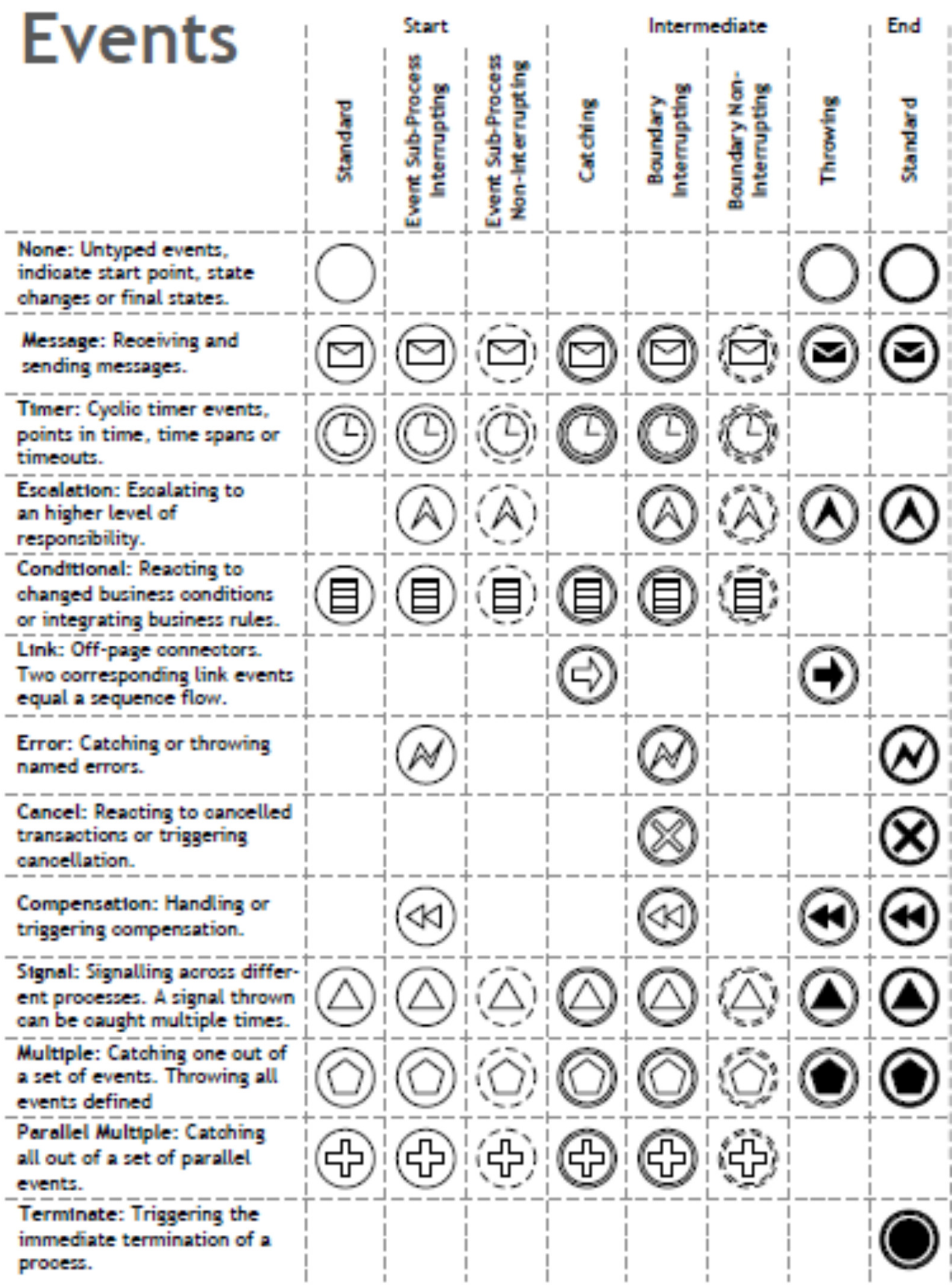


Figure 8. Typology of Events. (bpmb.de, 2011)

In Figure 9 it is shown a caption of the part of the orchestration model which is used as a sample of describing the sequence flow and data flow. A data object, a set of information, named as *External: Allocated slots* is used by task *Manage short-term schedule*. Upon completion of this task a message is sent to another peer (an autonomous entity named as *Fleet/tail planning system*) by which a task *Aircraft routing subprocess* is initiated thereafter. Once the task is completed the workflow is finalized by an end event.

Intermediate linking events allow to direct the sequence flow from one peer (entity) to another. Once a task *DOO schedule subprocess* has been completed by *Schedule planning* the throwing event labelled as *1.1* will direct the order of execution to catching event labelled as *1.1*. This event, in turn, launches a task *Tail assignment subprocess*.

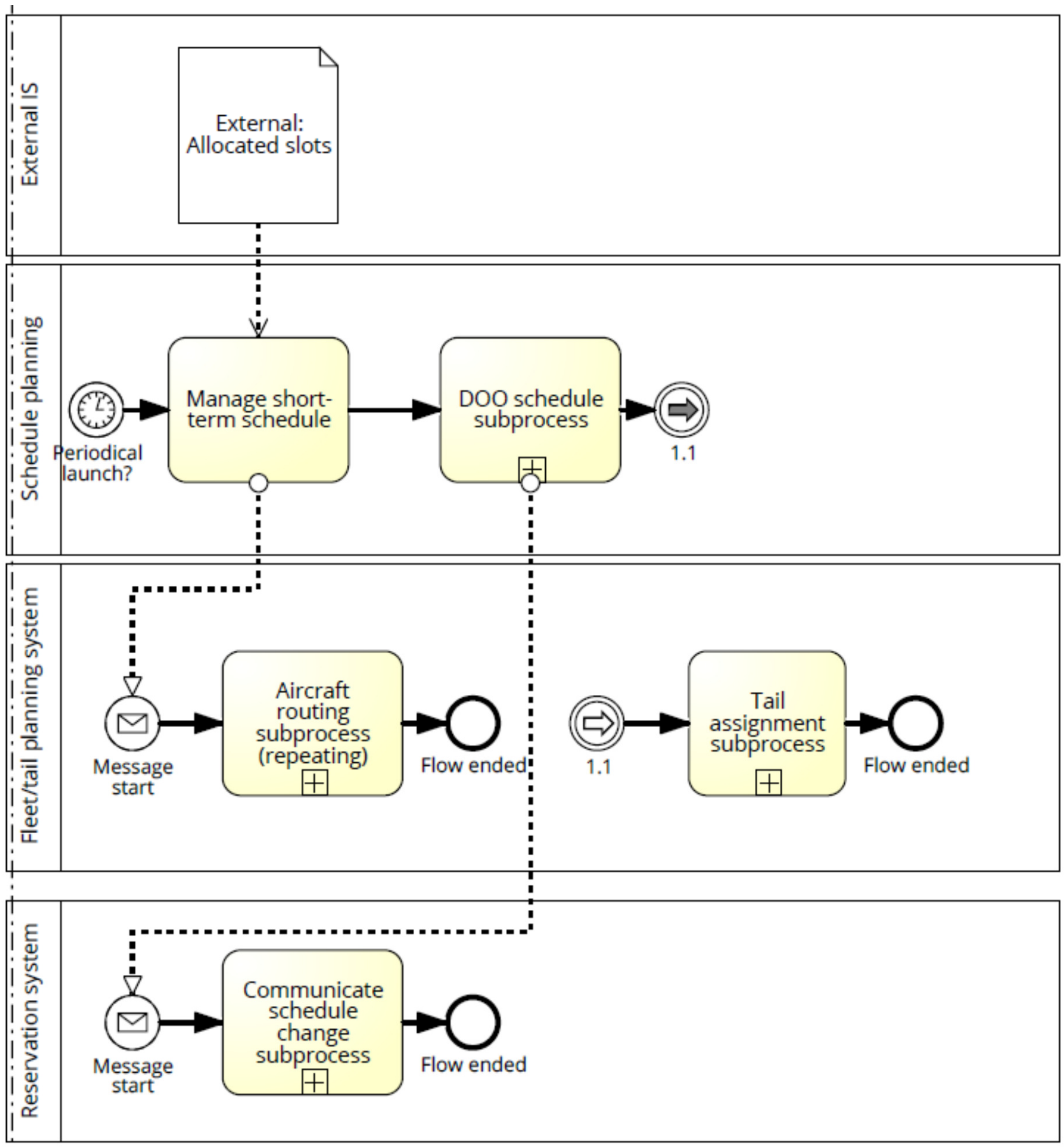


Figure 9. Excerpt from Orchestration model. (Rein Nõukas)

A task cannot be started if access to all instances of incoming data flow via data objects are not enabled. Simply saying, if a data object, a document in this thesis, is not 'ready' (*resp.* available) for use by the task then the task will wait for it to become available.

As per White and Miers (2008) data flow passes information in and out of the activity. If a sequence flow arrives at the task, it sets the state of the activity to ready. It is ready to begin, but cannot start until all of its inputs, in the form of documents, are available.

It could be stated that besides the regular sequence flow there exists a quasi-sequence flow enacted by the (un-)availability of the data objects, known as the data flow. In the case of inter-organizational, multi-entity collaborations the data flow seems to be even of higher importance than regular sequence flows because of the former's universal application - all entities can consume Data objects, whereas sequence flow is restricted to a single entity.

Having introduced properly the basics of the business process management and the cloud computing, it is time to move forward and reveal the application of the both concepts in the practical domain analysis. The analysis and design tasks are divided between Chapter 3 and Chapter 4.

3. Preparation of the AS-IS state

The first step in the process of configuring the cloud-based collaboration-enabling ecosystem for the aviation industry is identifying the AS-IS state of the theoretical airline operations. This is relevant because the aim of the thesis project is not to re-define the underlying logic of the airline operations, nor the way the operations are executed. Instead, the author makes an efforts to transform the organization of the operations in a way that it supports the virtual enterprise model.

Therefore, the goal of this chapter is to identify the native parts, from the airline's perspective, of the virtual enterprise model. This goal is achieved by following the given research question:

- RQ 1: How to normalize airline's business processes?

This research question is broken down into two sub-questions in order to distinguish the separate phases and deliverables of the research and development process:

  - rq 1.1: What are the re-built airline's business processes?
  - rq 1.2: What kind of methods can be used to refine the airline's business process model?

The first sub-question, located in Chapter 3.1, focuses on mapping the airline's business processes using the BPMN 2.0 framework in the form of choreography and orchestration. The second sub-question, located in Chapter 3.2, is about the refinement of the workflow model compiled in Chapter 3.1. Lastly, a summary and conclusions, located in Chapter 3.3, are provided in order to understand the nature and relevance of the deliverables produced during the enactment of the first main research question (*resp.* RQ 1).

3.1 Identification of airline's business processes

In Chapter 3.1 are explained the process of mapping and refining the business processes of an airline during the day of the operations. The aforementioned process is conducted by using choreography and orchestrations. The final deliverable of the process – a normalized AS-IS model - is described in detail in order to give a thorough understanding of the business processes populating the day of operations of an airline.

3.1.1. The utilization of choreography and orchestration

The identification of the airline's business processes (*resp.* the definition of the AS-IS state) takes place in iterative steps, of which the most notable are the usage of choreography and orchestration in order to provide a semantic and syntactic basis for the identified business processes of an airline.

The first phase of the identification of the airline's business processes is enacted by using choreography. The choreography-based AS-IS model is based on the preliminary business process model depicting the airline's operations compiled in Bizagi Process Modeler[21] by the domain experts and additional source materials. In order to maintain focus and avoid repetitiveness in this thesis, this phase is presented in Appendix A. The choreography-based AS-IS model, located in Appendix A, contains textual business process descriptions that are similar to ones created via orchestration in Chapter 3.1.2.

The main deliverable derived from the implementation of choreography – the initial day of operation (DOO) AS-IS model, located in Appendix B, representing the airline's business processes – is processed further by using the orchestration. The implementation of the orchestration, represented by the final, normalized DOO AS-IS model that see in Appendix C, is expanded at full extent in the sequel.

Due to the time and resource constraints the scope of the AS-IS model is limited to the following operational phases: short-term planning, DOO shift 1, DOO flight plan preparation, DOO flight plan calculation and DOO actual flight. This means that processes comprising of long term planning and after flight activities are out if the scope of this thesis.

3.1.2 Detailed view of the workflows in the normalized AS-IS model

In Chapter 3.1.2 are explained the details of each workflow of an airline operations covered in the normalized AS-IS model, also see in Appendix C. It is assumed the reader has familiarized oneself with the specifics of the notation used in the orchestration modelling presented in Chapter 2.2.3.

[21] Link to a site: www.bizagi.com

The screenshots containing the parts of the normalized AS-IS model are provided in the sequel. Also all diagrams are amended with relevant explanations covering the presented workflow's life-cycle.

3.1.2.1 Workflows located at Short-term planning phase

3.1.2.1.1 Aircraft routing

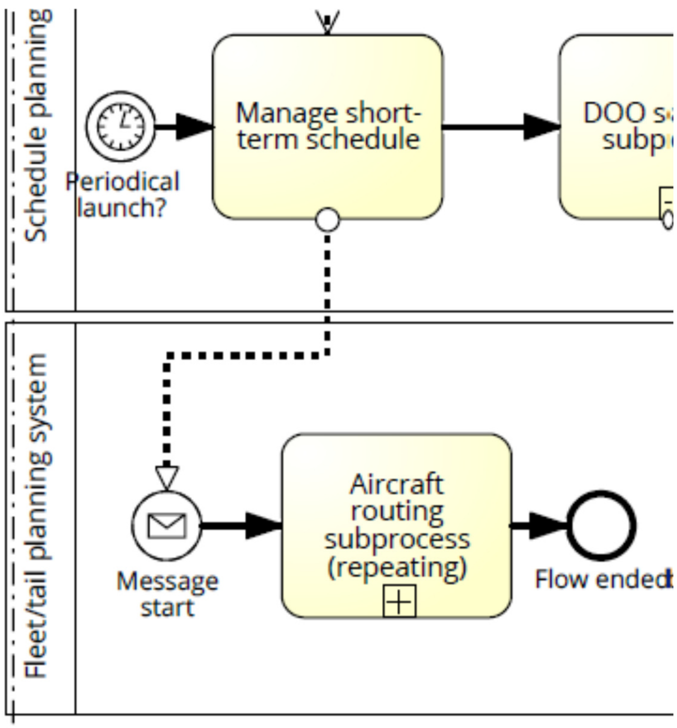


Figure 10. Aircraft routing workflow. (Rein Nõukas)

In Figure 10 the workflow named as *Aircraft routing* is condensed into a subprocess. As it is visible it is launched by a message sent by *Schedule planning* and providing some information to *Fleet/tail planning system* in the same way.

The workflow may be repeatable depending on circumstances undefined in this document.

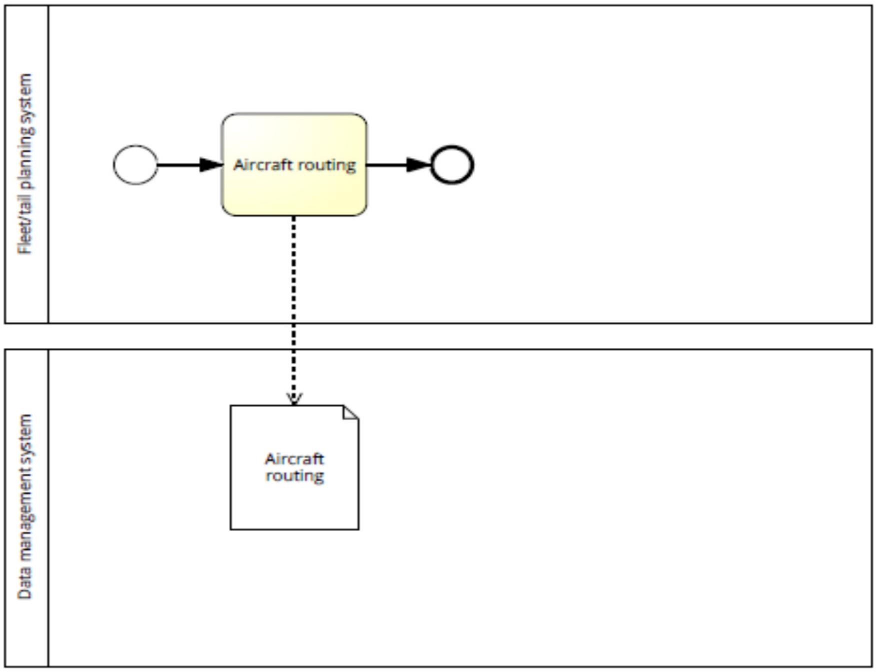


Figure 11. Aircraft routing subprocess, expanded. (Rein Nõukas)

In Figure 11 the contents of the *Aircraft routing subprocess* are visible. While the output document *Aircraft routing* is produced by *Fleet/tail planning system* it is stored in *Data management system*.

3.1.2.1.2 Communication of schedule changes

In Figure 12, the workflow named as *Communicate schedule change* is condensed into a subprocess. It is triggered by a message start initiated by *Schedule planning*.

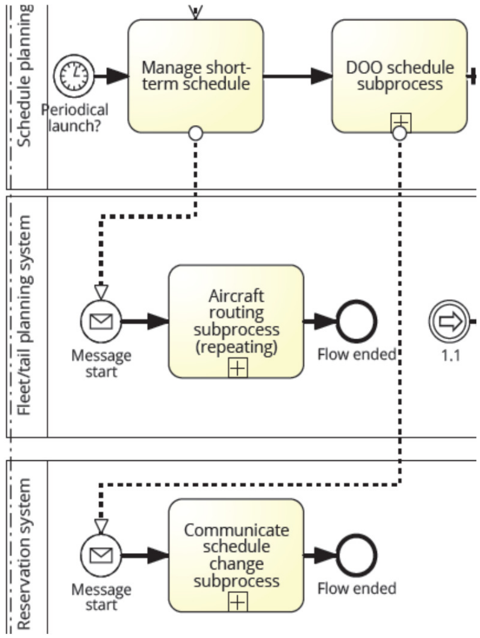


Figure 12. Communication of schedule changes workflow. (Rein Nõukas)

In Figure 13, *Communicate schedule change subprocess* is expanded. It consists of one task – *Communicate schedule changes.* As it is visible it uses (retrieves) a document *DOO schedule* in order to execute the task.

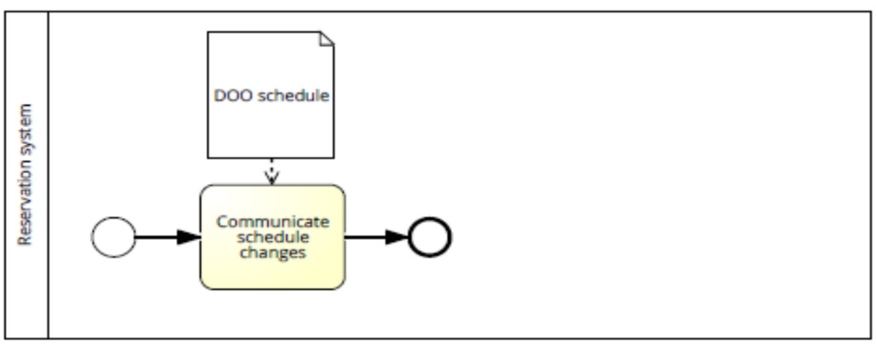


Figure 13. Communication of schedule changes subprocess, expanded. (Rein Nõukas)

3.1.2.1.3 Scheduling and tail assignment

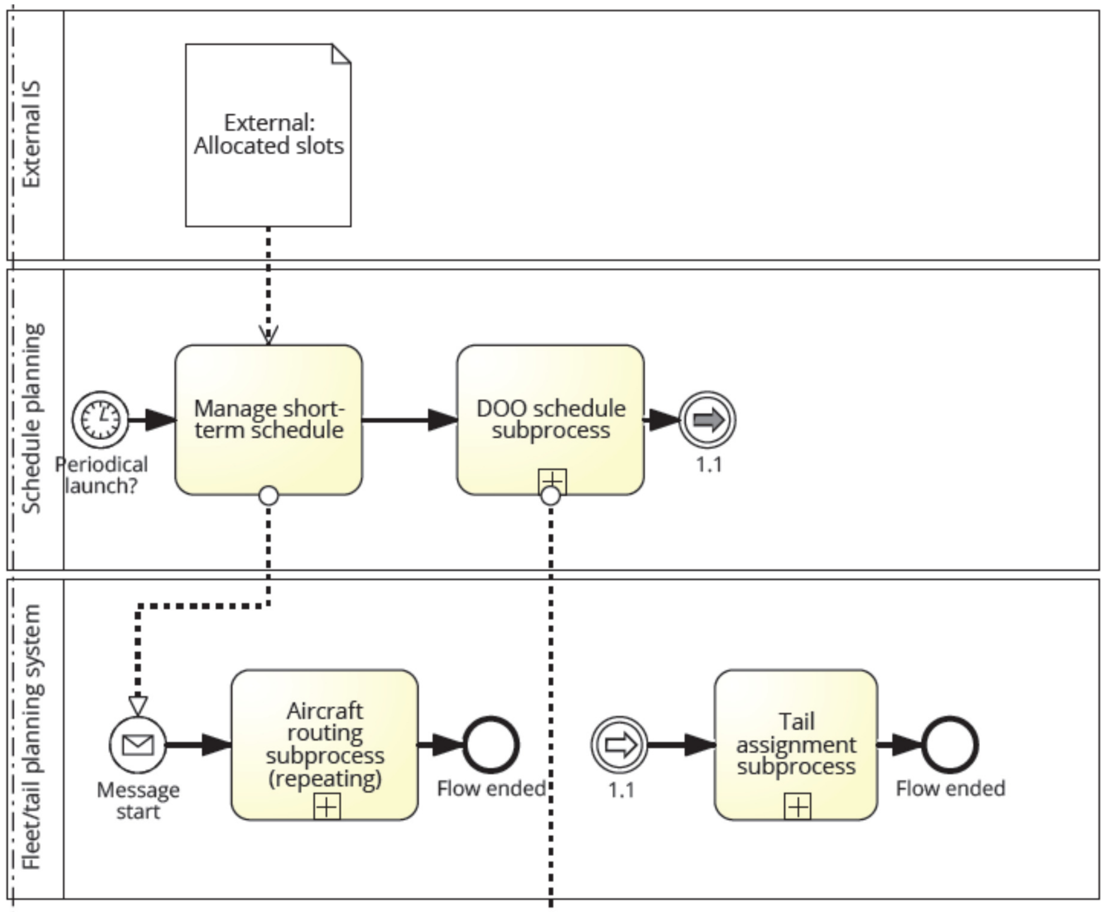


Figure 14. Scheduling and tail assignment workflow. (Rein Nõukas)

In Figure 14, the workflow named as *Scheduling and tail assignment* is displayed. The workflow is started periodically, exact launch time is not defined here. The externally created document named as *Allocated slots* has to be available in order to complete task *Manage short-term schedule*. Some information produced by this task are required by *Aircraft routing* workflow in order to initiate it.

The next step is *DOO schedule subprocess,* see Figure 15 below. It produces a document named as *DOO schedule* which is stored in four locations: *Operations control system, Schedule planning system, Data management system* and *Reservation system*. These locations serve as data sources respectively for *Crew management system, Data management system, Data management system* and *Reservation system.*

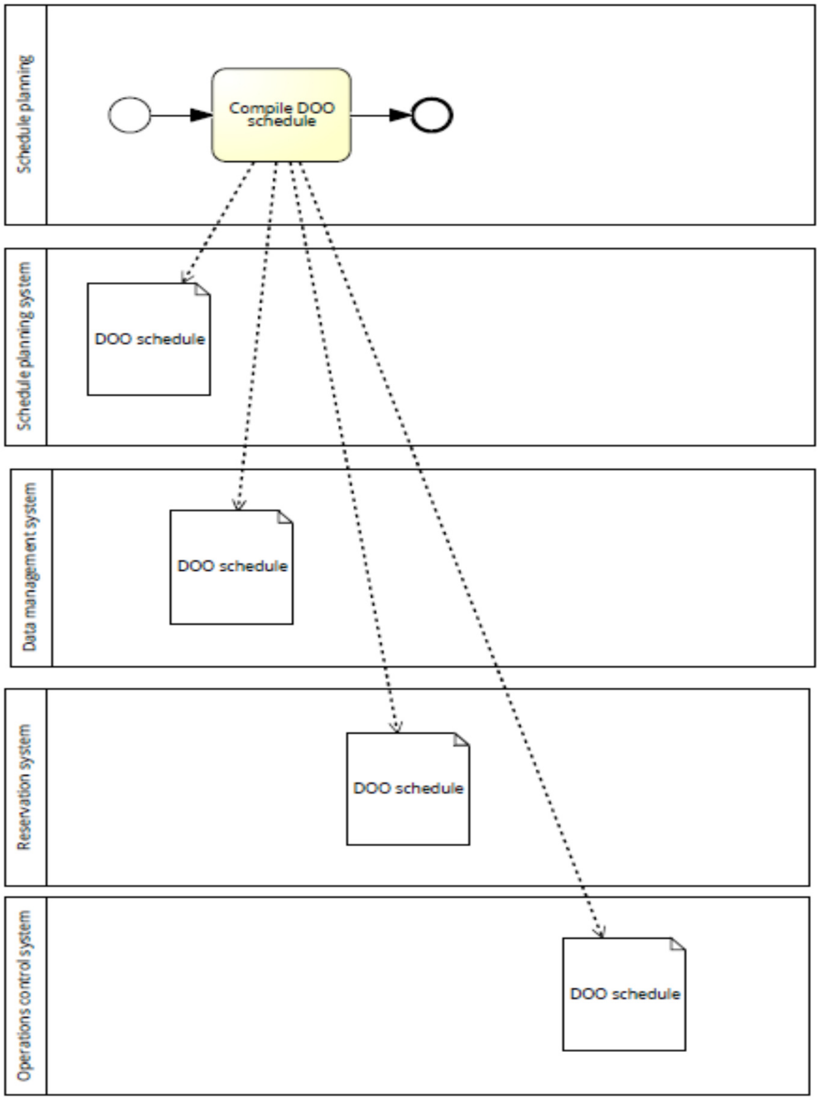


Figure 15. DOO schedule subprocess, expanded. (Rein Nõukas)

The final part of the workflow is the *Tail assignment subprocess*, see Figure 16 below. Externally created document *Fleet assignment*, located in *External IS*, and a document *Aircraft routing*, located at *Data management system*, are required in order to execute task *Create tail assignment*. The output of the task execution is a document *Tail assignment* that is stored in *Data management system*. Also the same system process the information given in the aforementioned document.

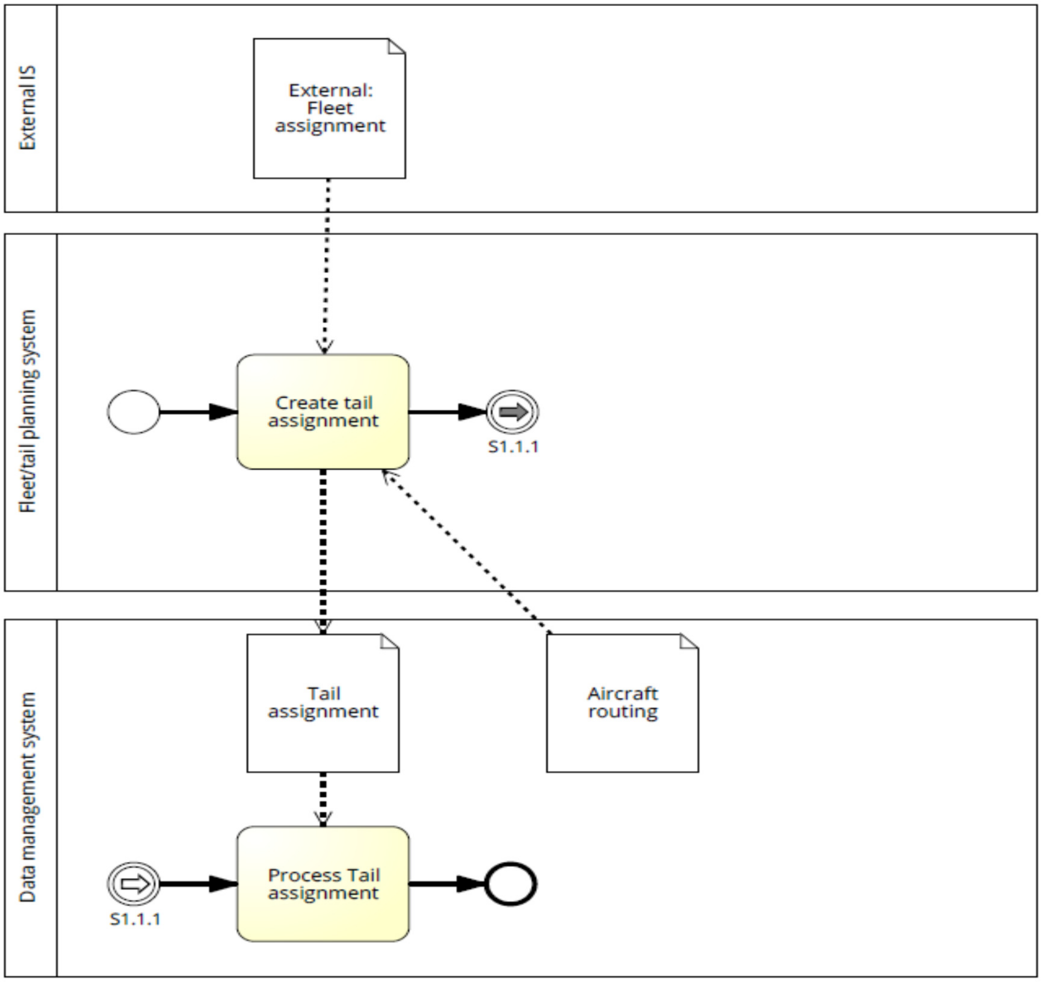
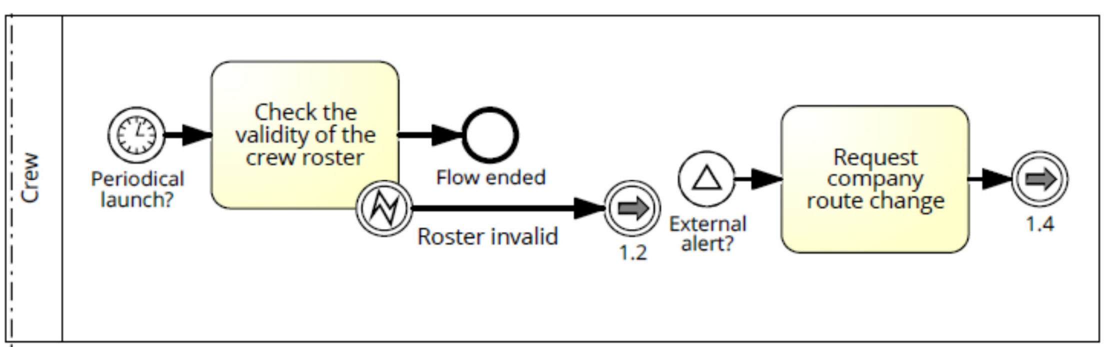


Figure 16. Tail assignment subprocess, expanded. (Rein Nõukas)

3.1.2.2 Workflows spanning Short-term planning and DOO shift_1 phases

3.1.2.2.1 Company route changing



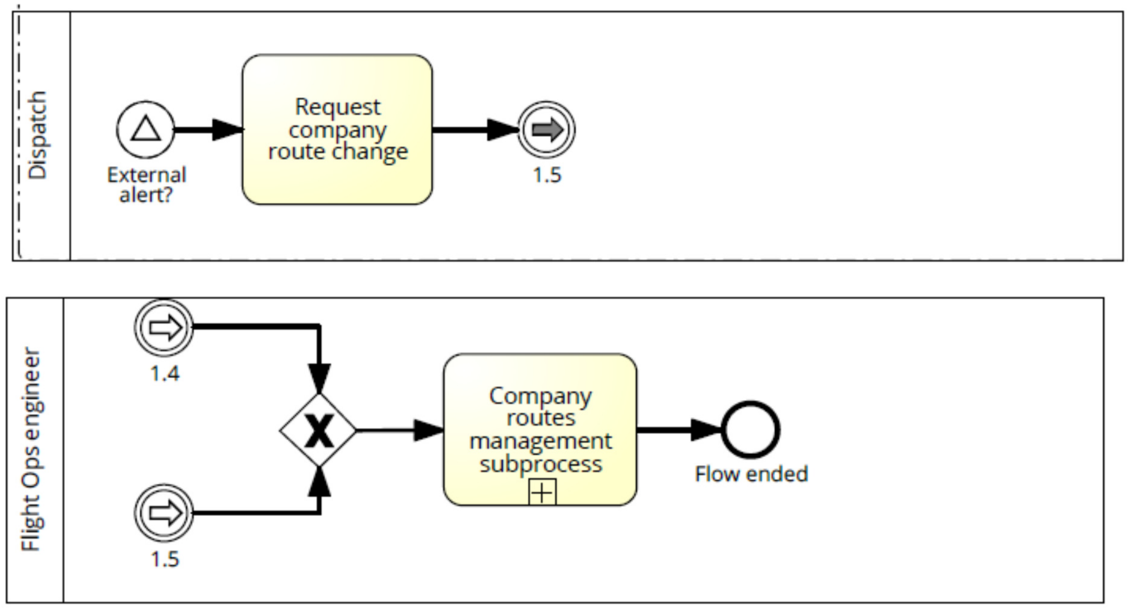


Figure 17. Company route changing workflow. (Rein Nõukas)

In Figure 17 it is visible that the workflow *Company routes management* can be initiated either by *Crew* or *Dispatch* or both of these units. Both initiation events depend on the incoming signal (*resp.* alert) event[22]. The actual process of implementing company route changes is initiated by *Flight Ops engineer*. The process itself is collapsed into a subprocess named as *Company routes management subprocess,* see Figure 18 below. This subprocess is executed entirely in phase *DOO shift_1*.

The aforementioned subprocess needs externally created document *Long-term schedule* in order to proceed to task *Validate runway analysis*. This tasks produces a document *Runway analysis* which is stored by *Flight planning system*. *Task Check payload*, operated by *Flight Ops engineer*, produces a document *New or updated company route*. This document is stored at *Data management system*.

[22] The exact nature of the event is not defined here.

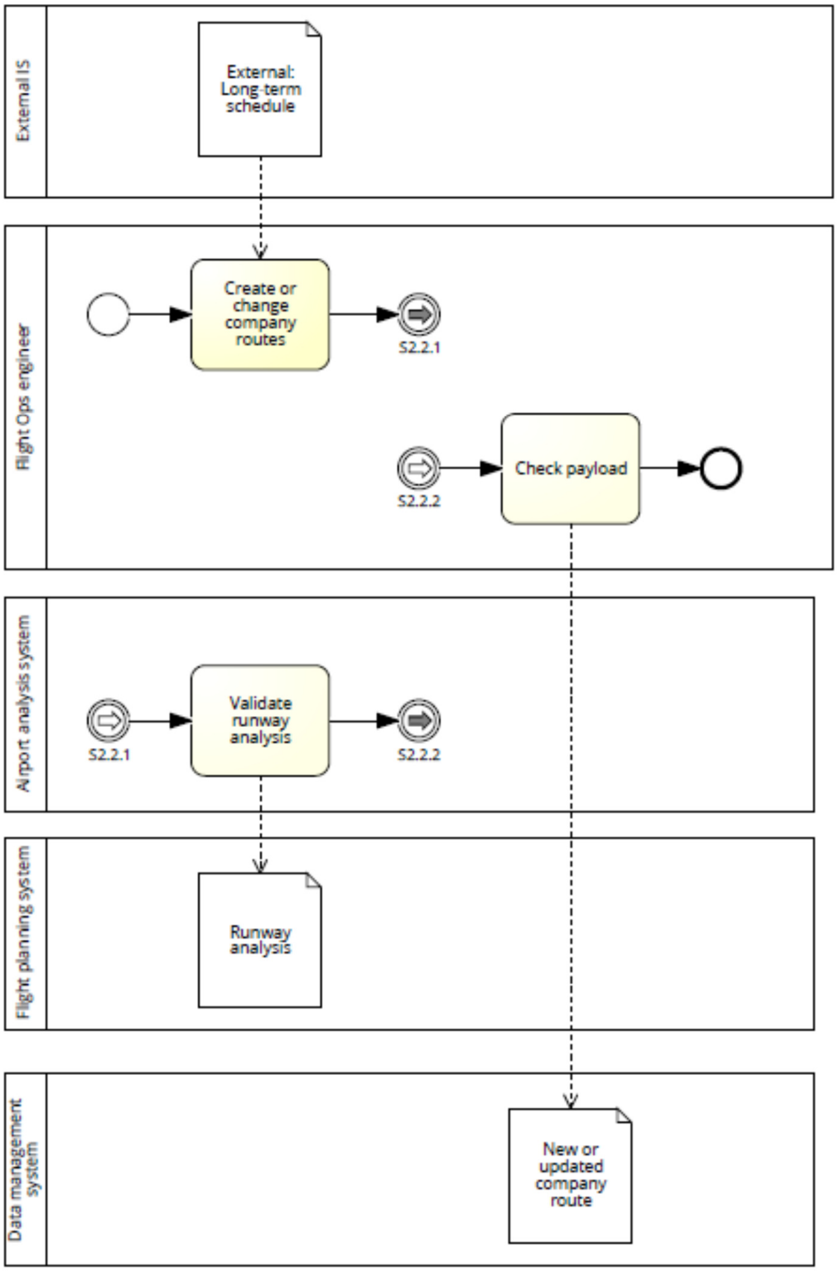


Figure 18. Company routes management subprocess, expanded. (Rein Nõukas)

3.1.2.3 Workflows spanning from phase Short-term planning to phase DOO flight plan preparation

3.1.2.3.1 Disruption management

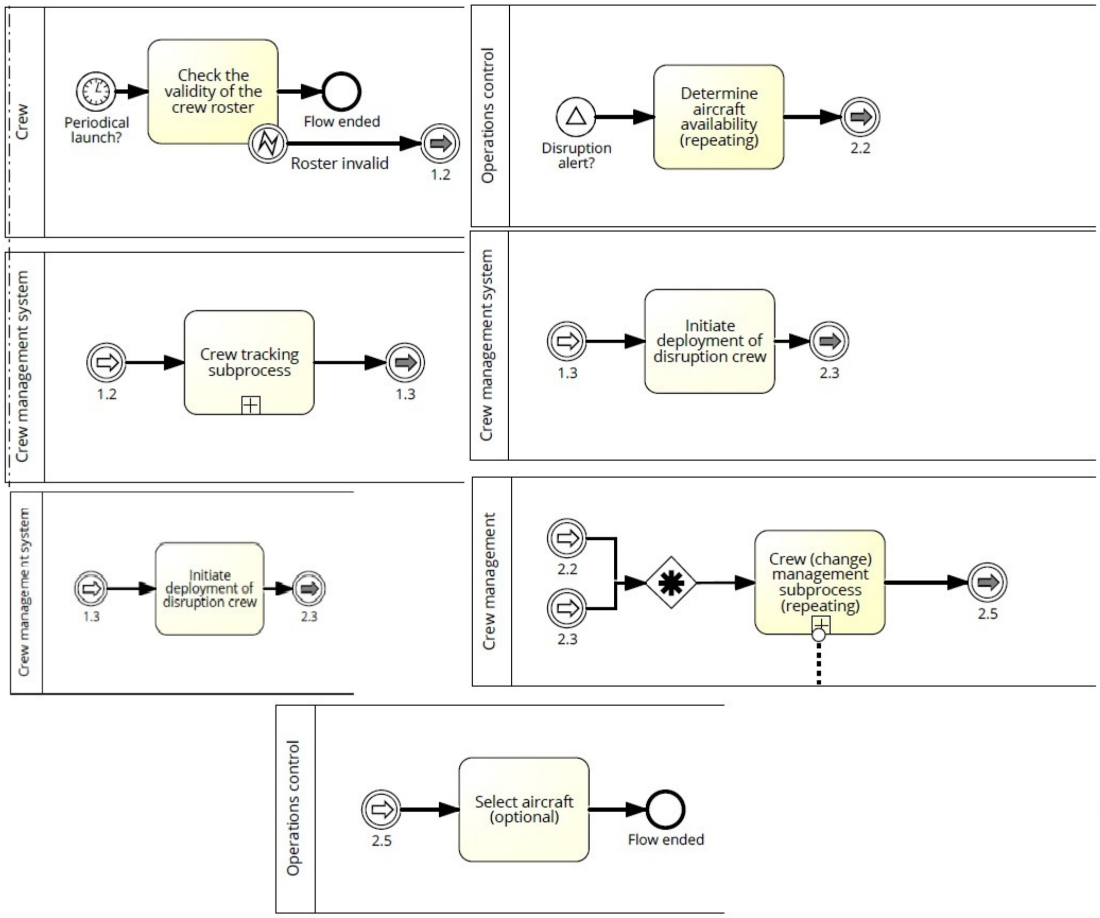


Figure 19. Disruption management workflow. (Rein Nõukas)

In Figure 19 the *Disruption management workflow* is displayed. The content covers the management of the disruptions, either created by (the absence of) crew or aircraft or both.

A periodical event that not defined here initiates the task *Check validity of the crew roster*. If an error occurs, for example the roster is invalid, a *Crew tracking subprocess*, see Figure 20 below, is launched. The execution of task *Crew tracking* requires access to documents *DOO schedule* and *Crew roster*.

In phase *DOO shift_1* the *Crew (change) management subprocess*, see figure 21 below, is initiated once either the task *Deployment of disruption crew* has been finalized by the *Crew management system* or if some disruptions with aircraft availability are detected. The initiation of the latter depends on a signal (alert) event not identified here.

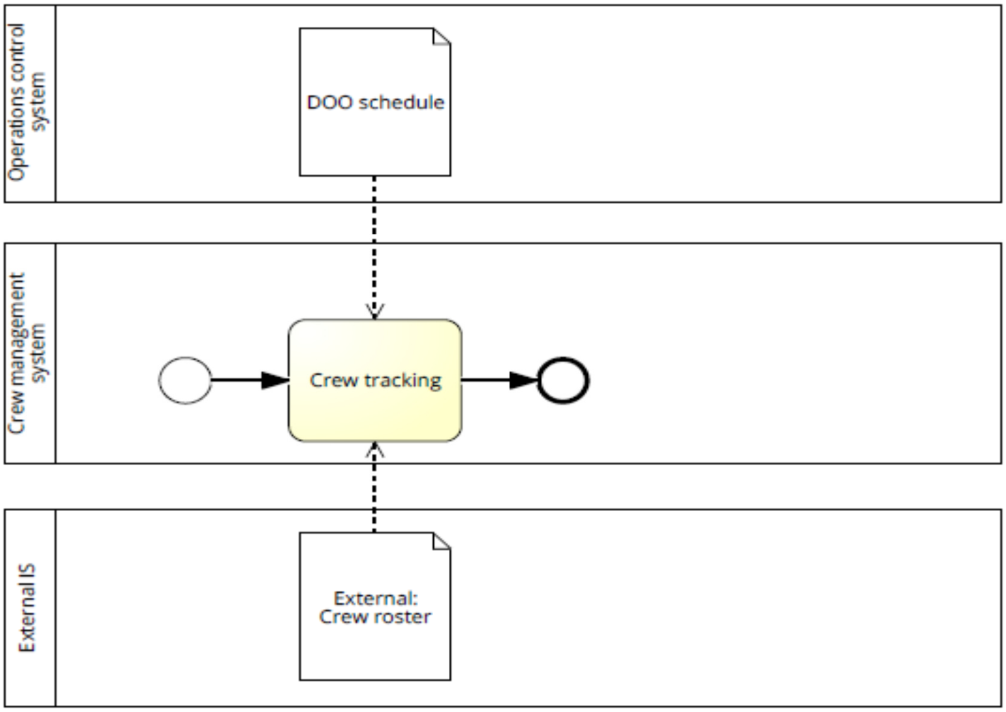


Figure 20. Crew tracking subprocess, expanded. (Rein Nõukas)

Figure 21 displays the *Crew (change) management subprocess*. In this subprocess task *Crew pairing and rostering* is executed one or several times until acceptable result is achieved. The result, a document *Crew change statement* will be stored by *Data management system*. The latter also serves as a data source for *Data management system* for processing the crew schedule.

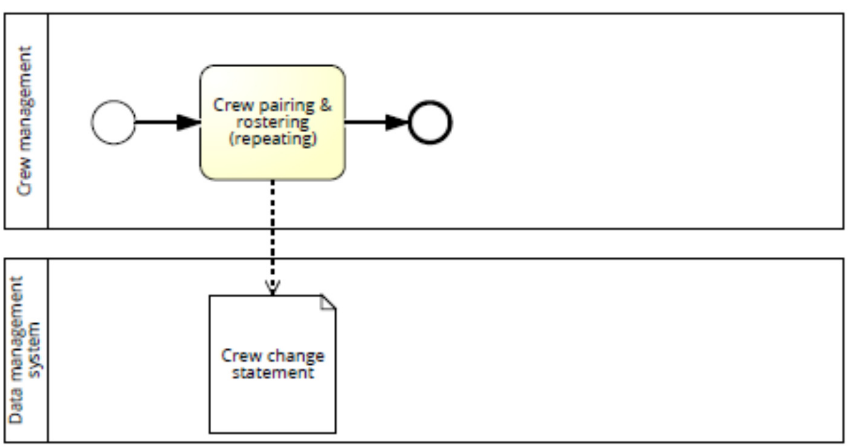


Figure 21. Crew (change) management subprocess, expanded. (Rein Nõukas)

The final step in this workflow is task *Select aircraft* executed *by Operations control*. This task becomes actual and valid only if there is a technical problem detected and the respective aircraft needs to be replaced by another one. This task is located at *DOO flight preparation*.

3.1.2.4 Workflows located at DOO shift_1 phase

3.1.2.4.1 Process Crew schedule workflow

In Figure 22 *Process Crew schedule* workflow is displayed. It is launched by the message event initiated by *Crew management*. In order to execute task *Process Crew schedule Data management systems* need access to two kinds of information: an externally created document *Crew roster* and a document *Crew change statement,* stored by *Data management system.*

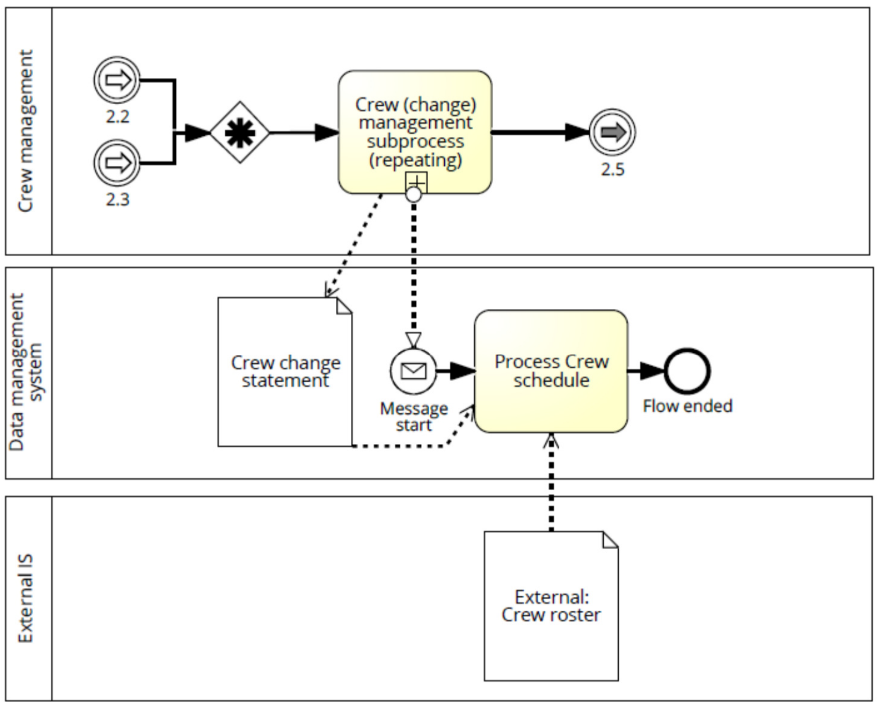


Figure 22. Process Crew schedule workflow. (Rein Nõukas)

3.1.2.4.2 Desk assignment compilation

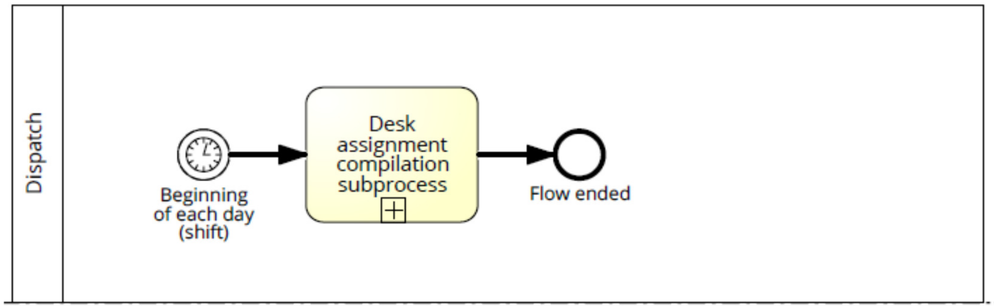


Figure 23. Desk assignment compilation workflow. (Rein Nõukas)

In Figure 23 *Desk assignment* compilation workflow is displayed. It is launched on a daily basis by *Dispatch*, probably in the beginning of a new shift.

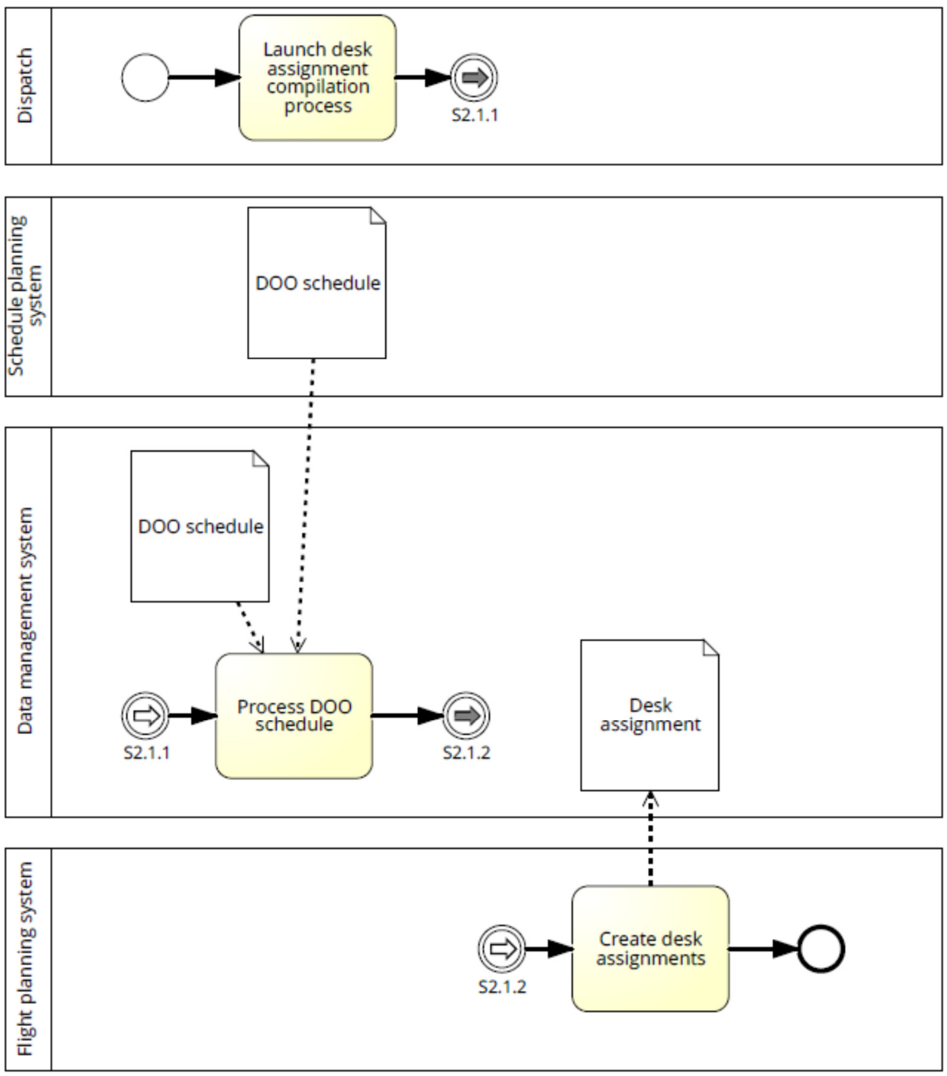


Figure 24. Desk assignment compilation subprocess, expanded. (Rein Nõukas)

Figure 24 explains the content of the *Desk assignment compilation* subprocess. In order to execute the task *Process DOO schedule Data management system* needs access to two instances of a document *DOO schedule*, stored respectively by *Data management system* and *Schedule planning system*. A document *Desk assignment* is created by *Flight planning system* and stored by *Data management system.* The latter serves as a data source for *Dispatch.*

3.1.2.5 Workflows located at DOO flight plan preparation phase

3.1.2.5.1 NOTAM

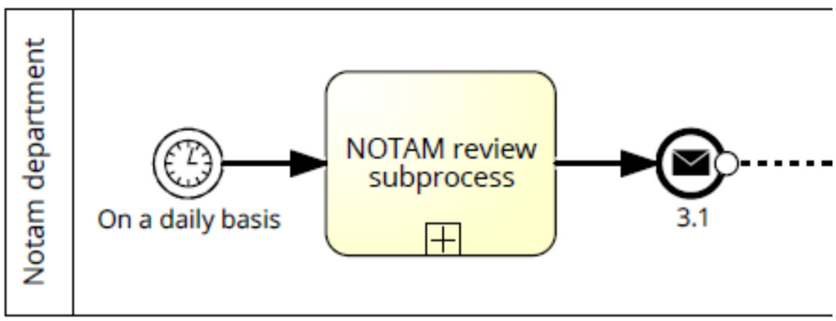


Figure 25. NOTAM workflow. (Rein Nõukas)

In Figure 25 *NOTAM workflow* is visualized. It is launched on a daily basis, exact time is not defined here. The workflow consists of one subprocess – *NOTAM subprocess* that is depicted in Figure 26.

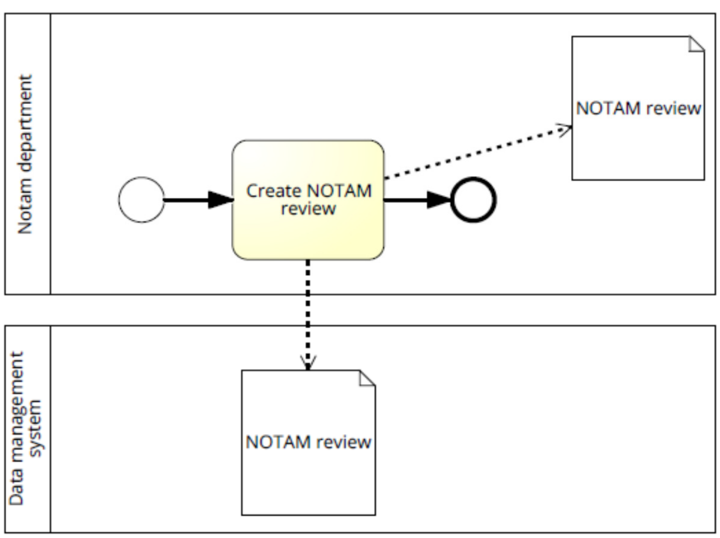


Figure 26. NOTAM subprocess, expanded. (Rein Nõukas)

Once a document *NOTAM review* has been produced by *NOTAM department* it is then stored in two locations: *NOTAM department* and *Data management system*. Both locations serve as data sources for *Flight Ops engineer* and *Dispatch* respectively.

3.1.2.5.2 Weather

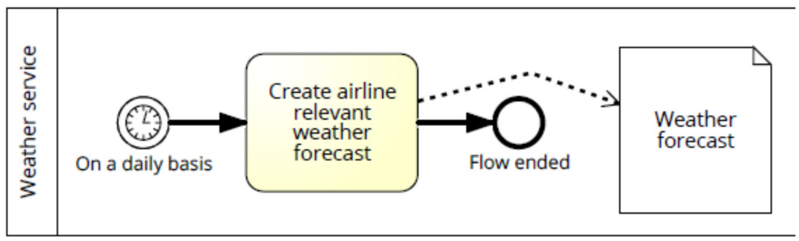


Figure 27. Weather workflow. (Rein Nõukas)

In Figure 27 *Weather workflow* is explained. It is launched on a daily basis, the exact time is not defined here. The workflow comprises a single task *Create airline relevant weather forecast*. The result of the execution of the task is a document *Weather forecast*, stored by *Weather service*, which is used by *Dispatch.*

3.1.2.5.3 MEL compilation

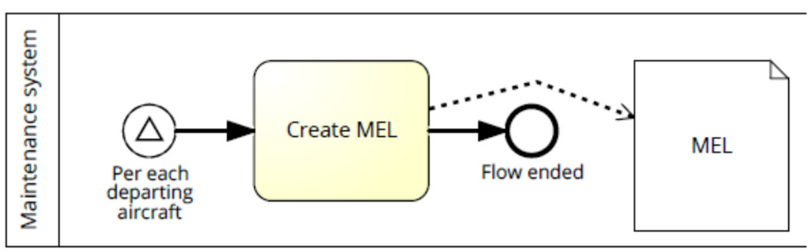


Figure 28. MEL[23] compilation workflow. (Rein Nõukas)

In Figure 27 *MEL compilation workflow* is explained. Workflow is initiated once per each departing aircraft, the exact source of signal (alert) is not defined here. The workflow constitutes of a single task *Create MEL.* The result of the execution of the task is a document *MEL* which is used by *Dispatch*.

[23] MEL is an abbrievation of term Minimum Equipment List.

3.1.2.5.4 ZFW calculation

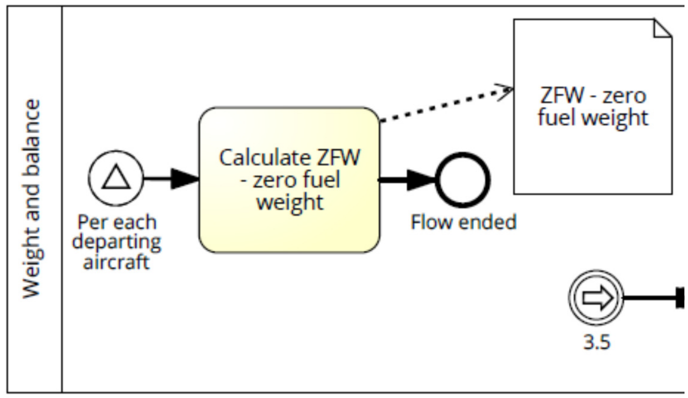


Figure 29. ZFW[24] calculation workflow. (Rein Nõukas)

In Figure 29 *ZFW calculation workflow* is depicted. The workflow is initiated once per each departing aircraft, the exact source of signal (alert) is not defined here. It consists of one tasks – *Calculate ZFW*. The task produces a document *ZFW – zero fuel weight* that is stored by *Weight and Balance*. The document is later on used by *Dispatch.*

3.1.2.5.5 Flight plan preparation

[24] ZFW is an abbrievation of term Zero Fuel Weight.

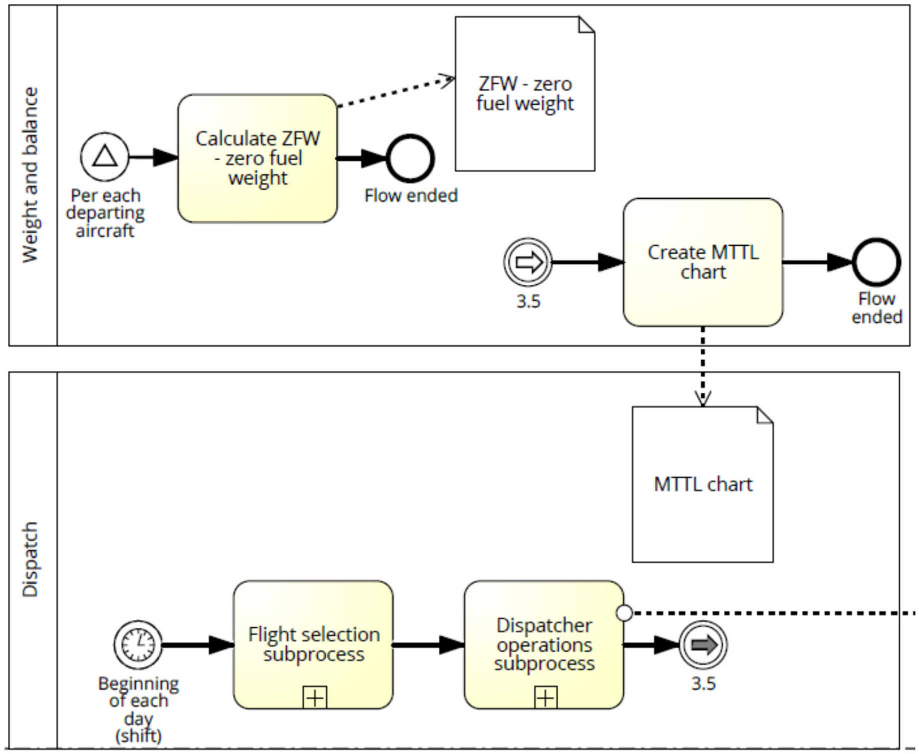


Figure 30. Flight plan preparation workflow. (Rein Nõukas)

Flight plan preparation workflow, displayed at Figure 30 is focused on the flight plan preparation. The workflow is initiated on a daily basis in the beginning of a shift (exact time is not defined here) by *Dispatch*.

In Figure 31 *Flight selection subprocess* is explained. *Dispatch* executes a task *Pull assignment flights into desk* during which a document *Desk assignment* is fetched from *Data management system*. Next, from *Weight and Maintenance* a document *ZFW* is fetched in order to execute a task *Select flight*. This task produces two documents: *Aircraft/crew change statement* and *Assigned flight*. These documents are stored as parallel tasks in *Data management system* and *Shared directory* respectively.

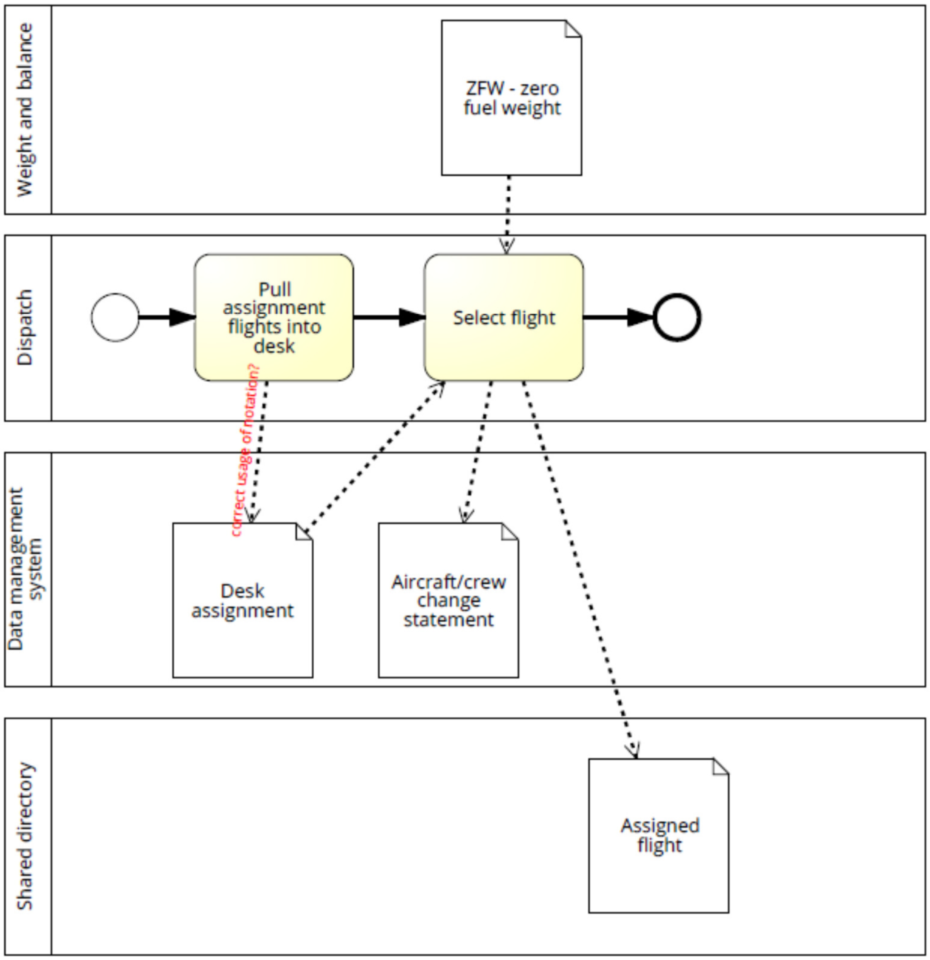


Figure 31. Flight selection subprocess, expanded. (Rein Nõukas)

The most populated subprocess, *Dispatcher operations subprocess* follows the *Flight selection*. The former has been organized into two layers: *Dispatcher operations layer*, see Figure 32 below, and its sublayer, *Dispatcher tasks*, see Figure 33 below. While the tasks' layer defines the tasks *Dispatch* unit has to execute in parallel, the operations' layer explains the data flow: inbound flow consists of documents *NOTAM review*, *Weather forecast* and *MEL*; outbound flow includes produced documents *Alternate airport*, *Flight level restriction/flight information* and *Selected NOTAM*.

The task *Store alternate airport* executed by *Data management system* is taking place in phase *Flight plan calculation.* The *data management system* serves as a data source to access a document *Alternate airport* via Service bus for *Dispatch*.

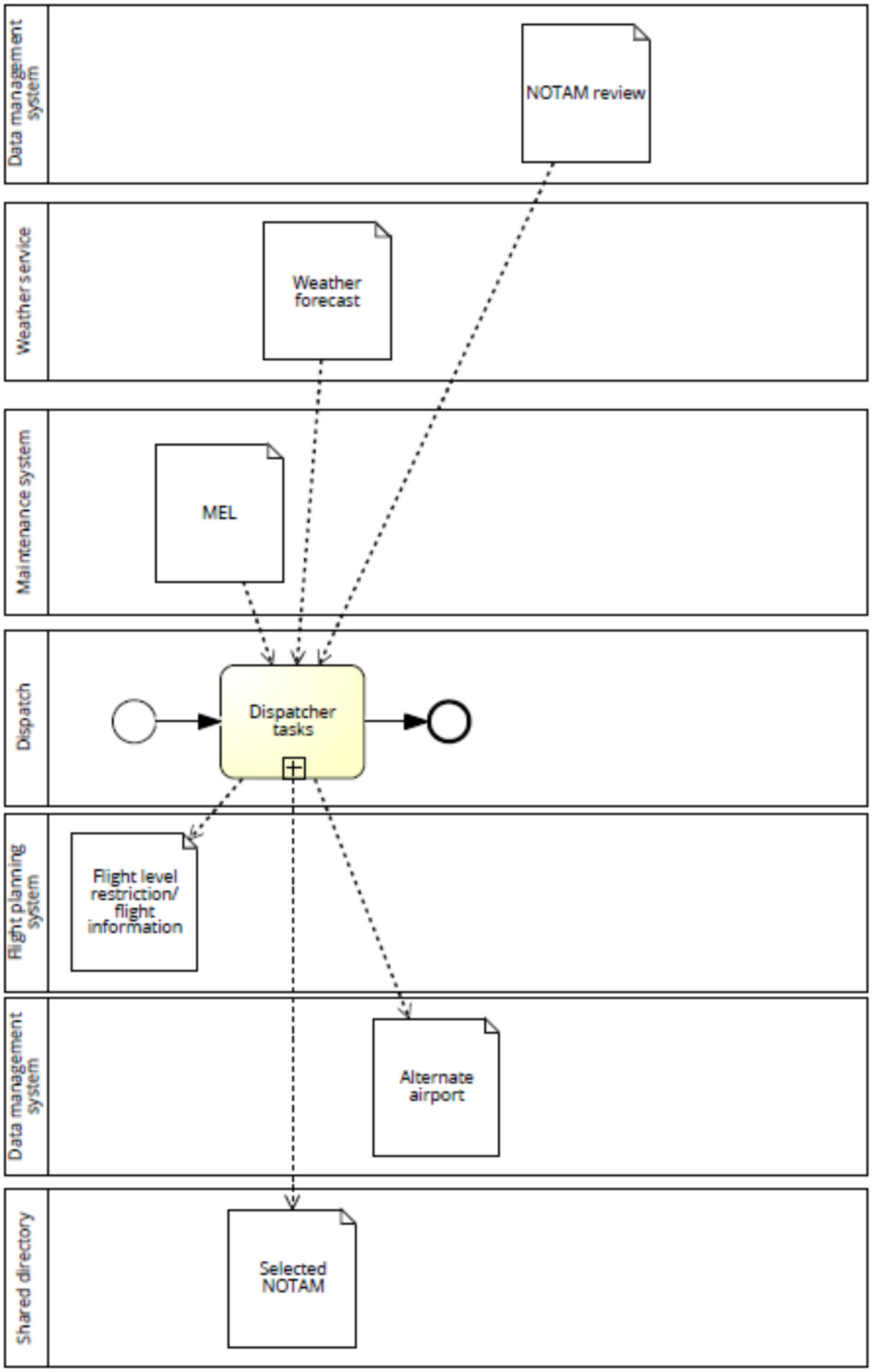


Figure 32. Dispatcher operations subprocess, expanded. (Rein Nõukas)

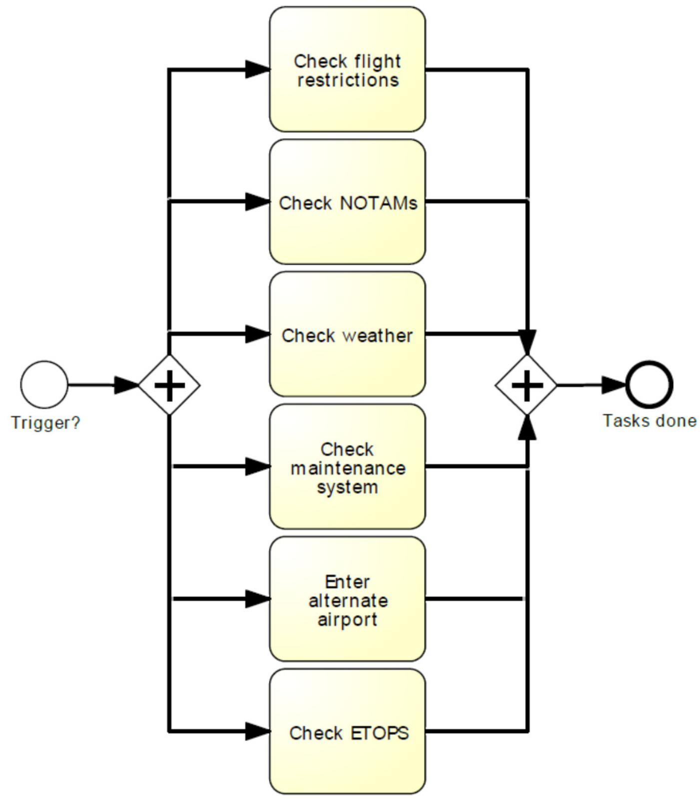


Figure 33. Dispatcher tasks, expanded – a sublayer of Dispatcher operations subprocess. (Rein Nõukas)

Final step in this workflow is a task *MTTL*[25] *chart compilation*. It is triggered by *Dispatch* that sends manually or via point2point connection data regarding the flight to organizational unit *Weight and balance* that is also called as performance system. The latter produces a document *MTTL chart* which is stored by *Dispatch* to be used for creating the finalized flight release and paperwork[26] for the flight in the phase *DOO actual flight*.

3.1.2.6 Workflows located at DOO flight plan calculation phase

3.1.2.6.1 Runway analysis

[25] MTTL is an abbreviation of term Module table take-off and landing charts.
[26] This is the case regarding AS-IS situation; TO-BE situation includes storing flight release and relevant documentation inside an Electronic Flight Folder to be used by (cockpit) crew.

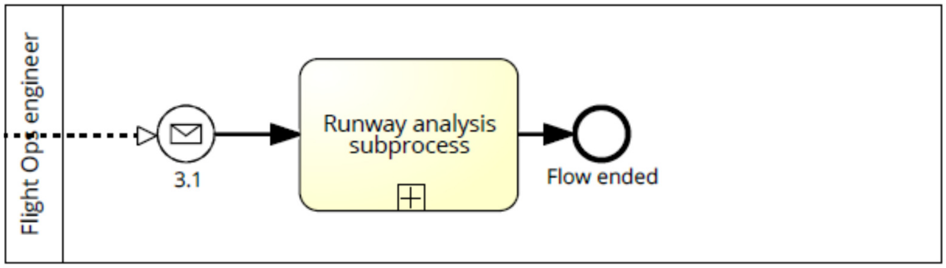


Figure 34. Runway analysis workflow. (Rein Nõukas)

In Figure 34 *Runway analysis workflow* is explained. It is triggered by the message event initiated by *NOTAM*[27] *department* – it sends a document *NOTAM review* via email to *Flight Ops engineer.*

The workflow constitutes of a single subprocess *Runway analysis*, see Figure 35 below. Before *Flight Ops engineer* can launch task *Perform runway analysis* it has to retrieve information, in the form of a document *NOTAM review*, from *NOTAM department* via email. The result of the execution of the *task Perform runway analysis* is a document *Runway analysis* which is stored by *Airport analysis system* and used later by *Dispatch.*

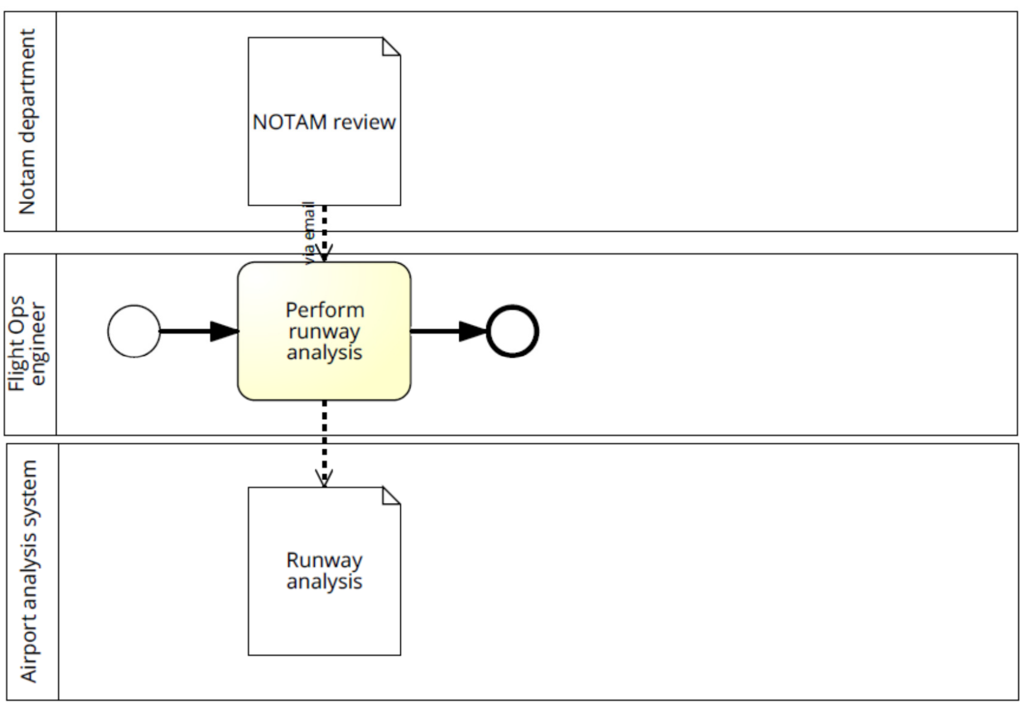


Figure 35. Runway analysis subprocess, expanded. (Rein Nõukas)

[27] NOTAM is an abbreviation of term Notice-to-Airmen.

3.1.2.6.2 Flight plan calculation & validation

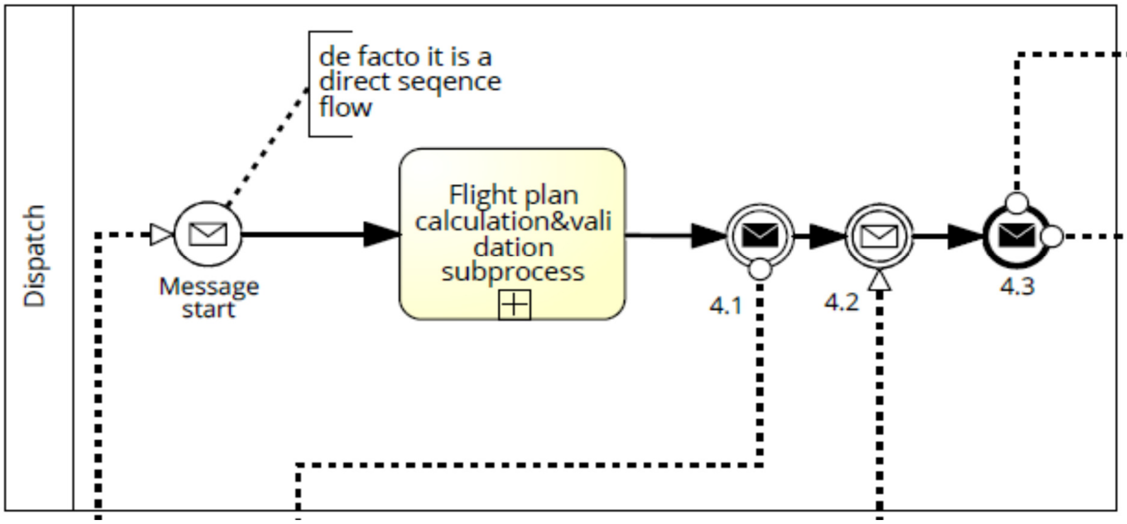


Figure 36. Flight plan calculation & validation workflow. (Rein Nõukas)

In Figure 36 *Flight plan calculation & validation* is visualized. The workflow is triggered by the message event initiated by *Dispatch*[28]. The workflow constitutes of a single subprocess *Flight plan calculation & validation*, see Figure 37 below. Before *Dispatch* can launch task *Calculate & validate flight plan* it has to access several documents:

- *Actual Payload* [29], estimates are calculated by *Dispatch*, final version is externally created by *Gate Agent*;
- *Flight level*[30], externally created via multiple iterations;
- *Alternate airport* stored by *Data management system*, accessed via Service bus;
- *Runway analysis* stored by *Airport analysis system,* accessed via Service bus.

The results of the execution of the task *Calculate & validate flight plan* are the following documents:

[28] *De facto* there is a sequence flow instead of a message event. Source: domain experts.

[29] Dispatch uses estimates based on male/female/crew defaults to calculate payload information as input. Final payload comes from the check-in process through the gate agent.
Often (especially in larger airlines) a dispatcher will be assisted by a load sheetner. For small airlines the dispatcher will do both. Load sheetners have the same license as a flight dispatcher. They must carefully plan the loading of the aircraft and do the weight and balance calculations for the aircraft. The outcome of this is the payload information. Source: domain experts.

[30] Flight level is a dynamic set of information based on an initial e.g. base season-dependent route (winter, summer, jet stream situation etc.). There is no concrete owner. Marketing might suggest a route, flight ops engineers calculate the base route including flight level and dispatch changes all flight levels again at the day of operation due to the actual situation. Source: domain experts.

- *Flight level*, stored by *Flight planning system*;
- *Flight plan,* stored by *Data management system*;
- *Fuel values,* queried by unit *Weight and balance*.

During the final step, *Flight planning system* conducts a task *Calculate & validate flight plan_2* once it has managed to access the document *Flight plan.* Once the workflow has completed the *Flight plan calculation & validation subprocess* it will engage into a coordination relationship with *Weight and balance* by using intermediate throwing and catching events in Figure 38. Once *Weight and balance* has completed its tasks the workflow can be finalized by the end message event triggering already another events in *DOO Actual flight* phase.

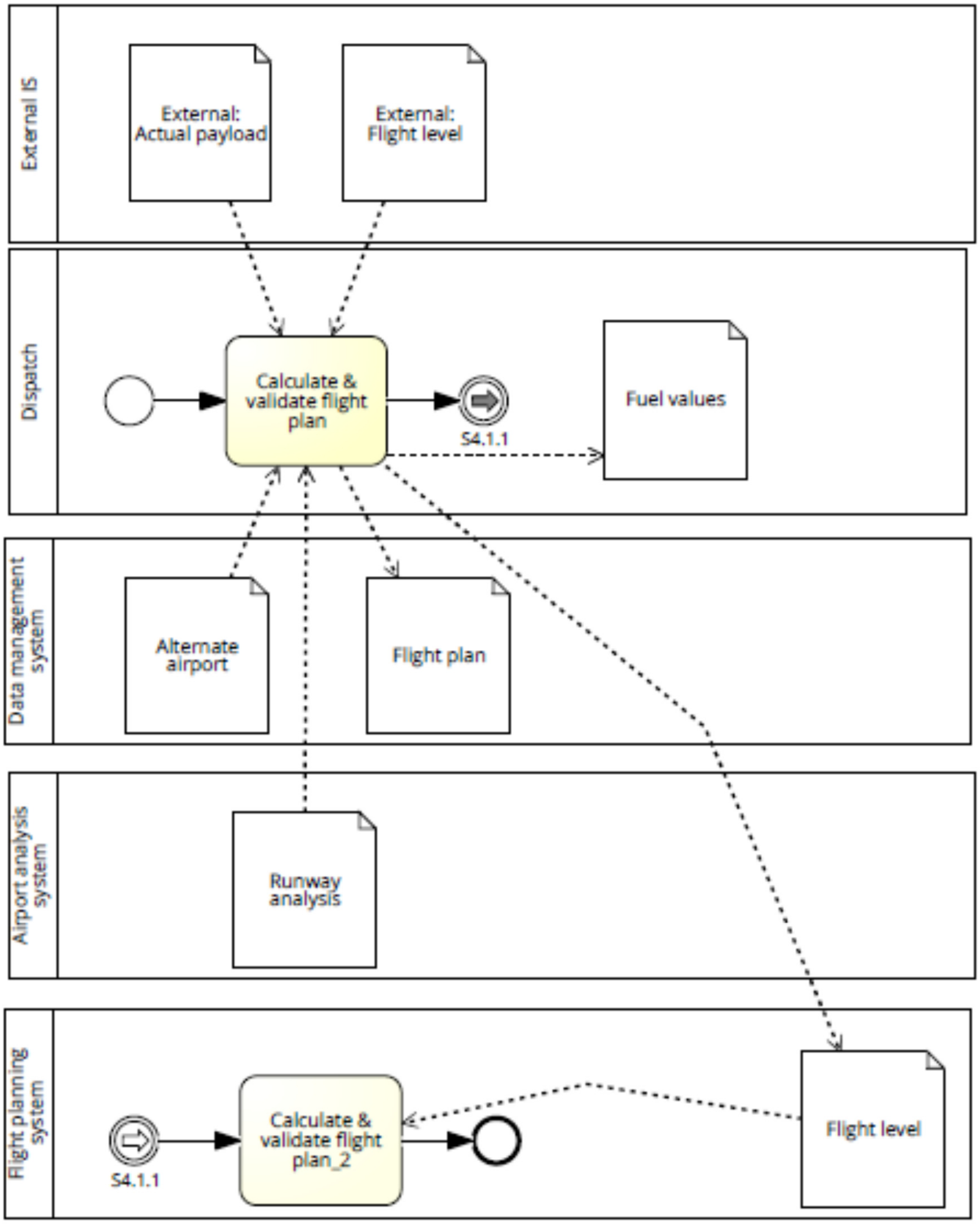


Figure 37. Flight plan calculation & validation subprocess, expanded. (Rein Nõukas)

3.1.2.6.3 Load sheet finalization

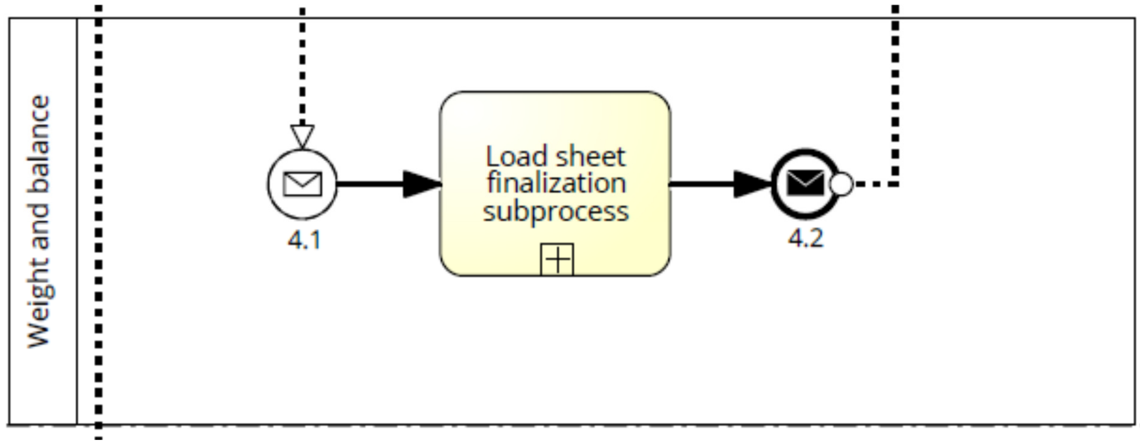


Figure 38. Load sheet finalization workflow. (Rein Nõukas)

In Figure 39 *Load sheet finalization workflow* is explained. The workflow is launched by a message event *4.1* initiated by *Dispatch* via email. The workflow comprises a single subprocess *Load sheet finalization*, see Figure 38 below. Before *Weight and balance* can launch task *Update Load sheet* it has to receive information, a document *Fuel values*, from *Dispatch* via email. The result of the execution of the task *Finalize weight & balance (Load sheet)* is a document *Load sheet*. This document is sent via ACARS to FMS which enables *Crew* to review it. After *Crew* has performed a task *Receive & confirm weight & balance (Load sheet)* a document *Load sheet_final* is created, it is stored by *Weight and balance*.

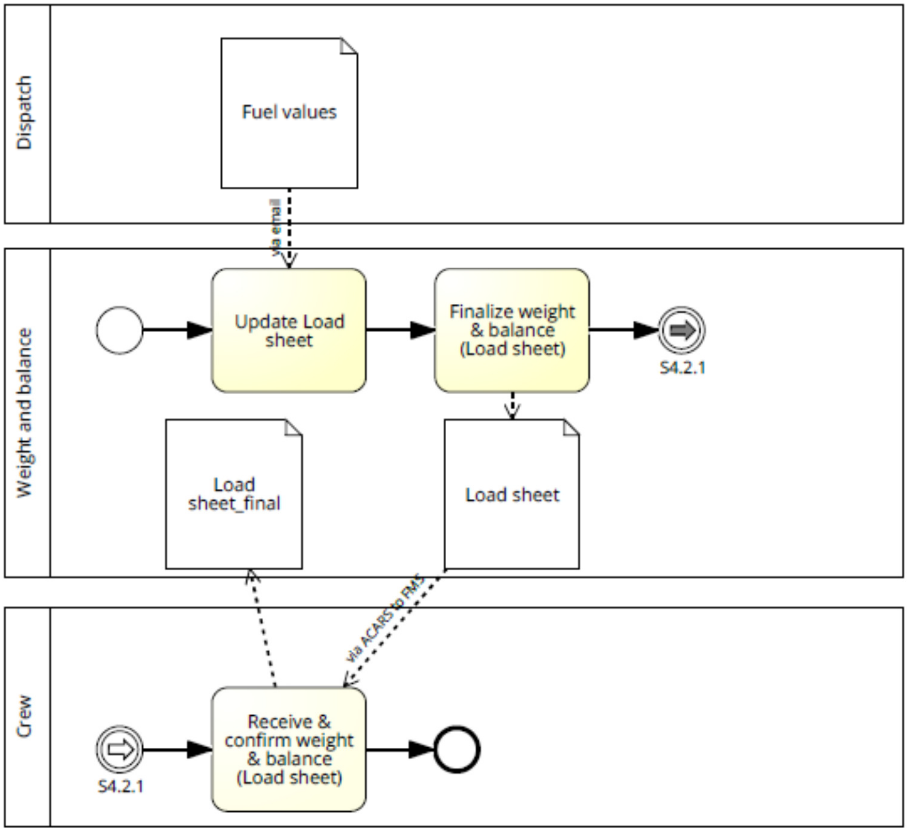


Figure 39. Load sheet finalization subprocess, expanded. (Rein Nõukas)

The workflow is finalized by the end message event *4.2* that informs the *Dispatch* about the right to proceed with its *Flight plan calculation & validation* workflow.

3.1.2.7 Workflows covering only phase DOO actual flight

3.1.2.7.1 ATC filing

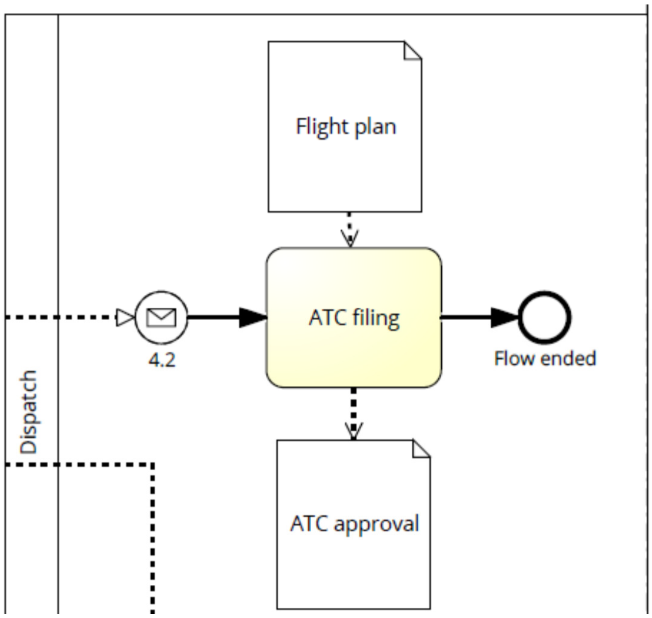


Figure 40. ATC filing workflow. (Rein Nõukas)

In Figure 40 *ATC*[31] *filing workflow* is explained. The workflow is launched by a message event initiated by *Dispatch.* The workflow constitutes of a single task *ATC filing* executed by *Dispatch*. The result of the execution of the task is a document *ATC approval*.

3.1.2.7.2 Crew briefing

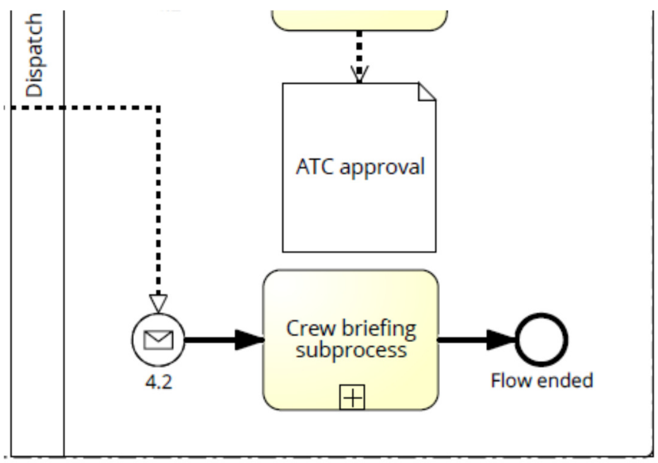


Figure 41. Crew briefing workflow. (Rein Nõukas)

[31] ATC is an abbreviation of term Air Traffic Control.

In Figure 41 *Crew briefing workflow* is displayed. The workflow is launched by a message event initiated by *Dispatch*. The workflow constitutes of a single subprocess *Crew briefing*, see Figure 42 below.

Before organizational unit *Dispatch* can launch task *Print flight plan* it has to retrieve a document named as *Flight plan* from *Data management system* via Service bus and access a document named as *MTTL chart* located at *Dispatch*' local data repository.

Once an artifact, a paper-back document, *Printed flight plan* has been created it will be manually stored and later taken to unit *Crew* for validation, review and acceptance. In parallel unit *Dispatch* creates a document *Briefing package* that will be stored as parallel tasks by *Crew* in a paper-back form, and by FMS[32] and *Data management* system in a digital form located at electronic flight folder.

If provided documents *Flight plan* and *Briefing package* include incorrect, or unacceptable direction, the unit *Crew* can process task *Validate & accept briefing package* in several iterations. After *Crew* has accepted the flight plan its acceptance, in the form of a document *Crew's acceptance*, will be sent to *Dispatch* and stored by this unit.

[32] FMS is an abbreviation of term Flight Management System.

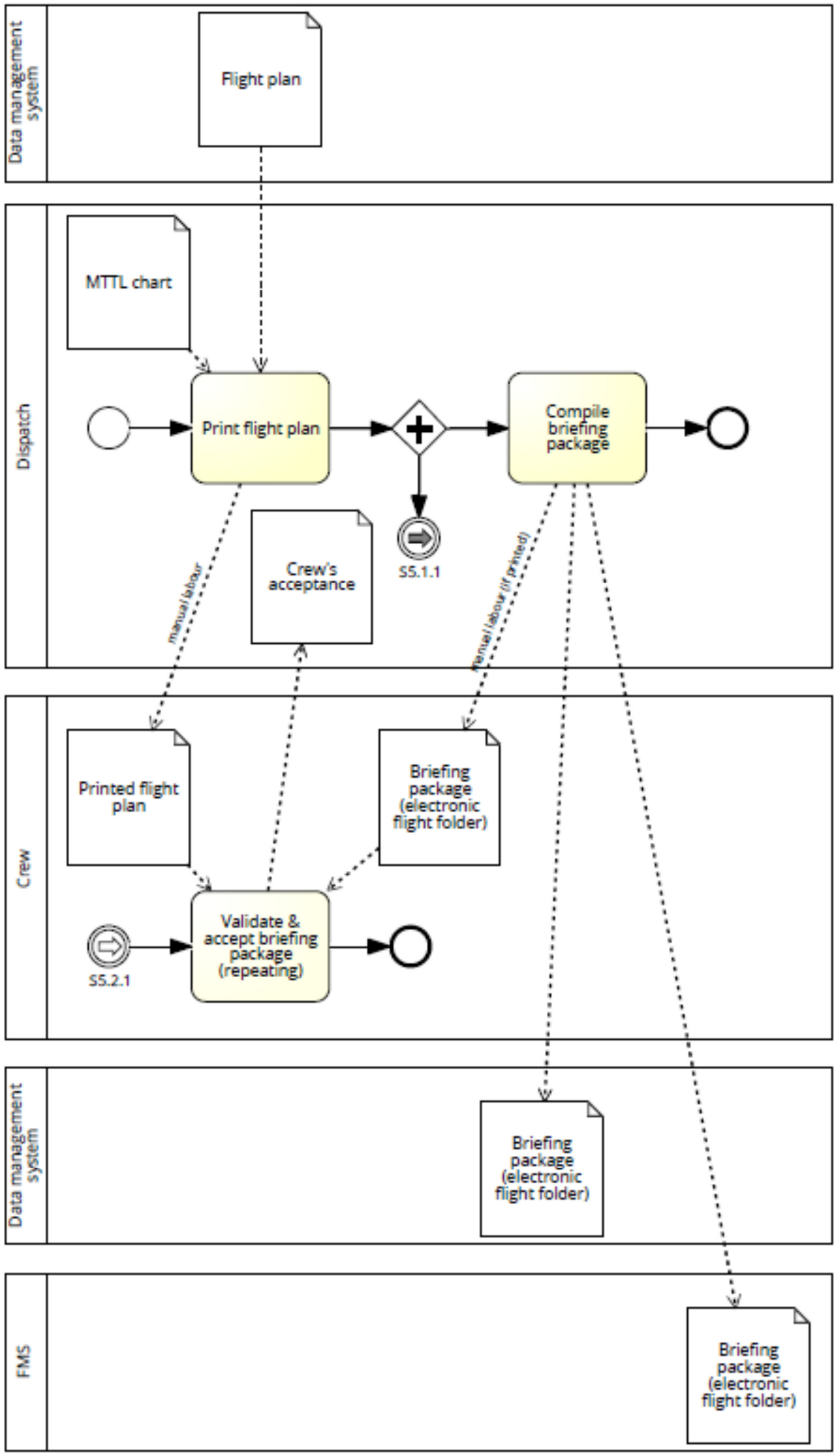


Figure 42. Crew briefing subprocess, expanded. (Rein Nõukas)

3.2 Refinement of an airline's business process model

The deliverable of Chapter 3.1, the normalized DOO AS-IS model, consists of constructs that are based on the BPMN 2.0 orchestration framework. The model covers the wide range of airline's business processes. As the modelling was based on the preliminary business process model depicting the airline's operations the normalized DOO AS-IS model cannot be considered as the final state of the airline's business process model.

In order to amend the existing model with currently unspotted and missing details the refinement of the model is executed. This way the model reflects better the real life situation and becomes self-contained at a higher degree. As there is no universally acknowledged airline business process model then the deliverable of the refinement process does not aim to acclaim this status also. The refined airline business process model is used, in turn, in this thesis to provide a basis for the architectural constructs in Chapter 4.

The refinement of the airline's business processes model is expanded at full extent in the following sequential subchapters of Chapter 3.2:

In Chapter 3.2.1 it is discussed the process of amending the normalized DOO AS-IS model with the DOO goal models that are compiled by Jekaterina Balashova (2014). Then the validation of the merged model with the assistance of the domain experts and source documentation is covered by the author in Chapter 3.2.2.

Based on the merged workflow model, the conceptual model, named as a value chain model, is created in Chapter 3.2.3 as it allows to detect a critical path and continue validation efforts in order to get as close to real life as possible. Finally, in Chapter 3.2.4 the refinement process is finalized by defining a clear scope for the business process model. This enables to increase the level of detail, and remove unnecessary parts, and to spot and describe vital characteristics of a workflow.

3.2.1 Amending DOO AS-IS model with DOO goal models

In this chapter the BPMN-based normalized DOO AS-IS model is amended with goal models and other types of models being part of the same set covering the same focus area – the day of operations of an airline. As goal models differ significantly, notation- and semantics-wise, from the workflow models then the author has discussed certain aspects needed to be accommodated during the matching process. Then, the actual matching process is described, highlighting the lack of time-related characteristics of goal models which, in turn, results in the omitting of data-flow. Also this may result in an inability to incorporate some content of the goal models into a workflow model.

3.2.1.1 Prior considerations and accepted limitations

In Chapter 3.2.1.1 are discussed the important aspects that were considered prior launching the activity of matching DOO goal models and normalized DOO AS-IS model.

The product of the matching process, a merged model, has to be a continuation of the normalized AS-IS model in order to carry a real value for further analysis. The non-exhaustive list of suggested aspects attributable to the merged model is as follows:

1. Visually, notation-wise the model should be similar to AS-IS model. Therefore the chosen modelling framework is BPMN 2.0 enacted in Signavio.
2. The semantics of the notation used in the model should be similar to AS-IS model. The generation of new, ad-hoc meanings attributed to symbols used in the model should be kept as minimal as possible.
3. The model has to fit to its purpose and goal, i.e., maintain its abstract nature while being detailed enough to allow capturing important changes compared to the AS-IS model.

3.2.1.2 About how the matching process is conducted

Role and goal models lack time-scale, i.e., roles ad goals don’t have time-based and -dependent relationships. Also these models don’t describe the flow of information between entities keeping the utilized data objects hidden and omitted.

The aspects described above mean that it is not possible to merge workflow models with goal and role models directly. However, the role and goal models are useful for spotting certain details which have not been covered by the workflow modelling. For example, some additional sets of activities or even workflows could be detected that complement the existing set up of workflows. This in turn allows to recognize additional data sets produced and consumed by the entities.

In parallel, it is relevant to state that the goal of the merging process is not a refinement of goal and role models – the process is aimed solely at enhancing the quality (*resp.* value) of the workflow model by adding previously unspotted, yet integral, workflow components.

3.2.1.3 Excluded goal models

The day of operations (DOO) provides the underlying temporal and spatial structure for the running case. In turn this means that all the content of goal models that are intended to be included to the workflow model should be redefined and reconfigured using those two parameters. This enables the actual integration of selected content from goal models and the existing components of workflow model.

Whereas it is possible to populate the spatial[33] parameter as both the goal models and workflow model share the same domain[34] then the problem arises from defining the time-based events. Workflows designed to represent processes at low abstraction level primarily depend on certain sequences in order to carry out intended activities (*resp.* changes of states). Goal models used in this project omit sequencing and focus solely on the relationships of the actors. The author admits that some other workflow modelling languages (like Event-driven Process Chains (Aalst, 1998)) utilize event-based processes that could be used for building a more abstract business process layers.

Given the aforementioned major differences between these modelling frameworks, certain content of a goal model is not possible to integrate into a workflow model. Or at least, the integration is not feasible if the actual meaning and semantics are to be preserved. In parallel, due to the high level of abstraction embedded in the semantics of a goal model the integration might not be successful.

[33] By spatial parameters the domain entities and their relationships are described.
[34] In the running case the domain is the civilian aviation (*resp.* airline company and the civilian airport).

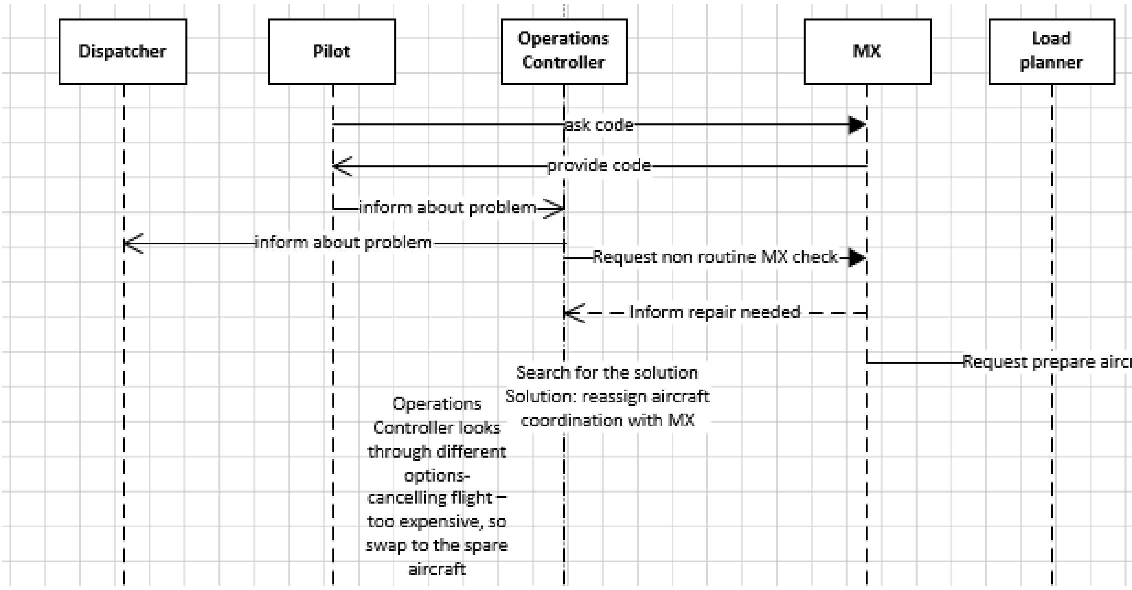


Figure 43. Part of the interaction model describing the relationships between roles. (Balashova, 2014)

An example of such an occasion is disruption management - see the excerpt of an interaction model depicting the handling of an aircraft malfunction in Figure 43. As disruptions may happen virtually in every subdomain and at any time then the goal model representing disruption management is not fitting to the existing DOO workflow model. The latter is heavily time- and space-dependent, mostly suitable for describing non-exceptional sequential activities. As alerts and the reactions to handle them have numerous varying configurations then the only solution is to use a highly abstract workflow, like business process viewpoint[35] in ArchiMate in order to depict the disruption management. While doing so still the actual meaning, semantics, is lost as the newly created dependencies, enacted by the relationships, between entities inside the merged model fail to present any sequential behaviour.

The result of the process of amending the normalized DOO AS-IS model with DOO goal models is named as the DOO TO-BE model, located in Appendix D. The reasons causing the evolution of this workflow model are based on the following aspects:

- The merged model doesn't represent anymore only the intentions of its original creator but it includes also the aviation industry's best practices

[35] The Business Process viewpoint is used to show the high-level structure and composition of one or more business processes. It describes the assignment of business processes to roles, which gives insight into the responsibilities of the associated actors. (Beauvoir, 2014)

- and the author's own line of reasoning on how to organize the logic of processes by including
- workflow parts and configurations not existing in previous (AS-IS) model.

3.2.2 Validation of the DOO TO-BE model

Domain experts were interviewed in order to validate the DOO TO-BE workflow model that represents the day of operations of an airline and includes both contents of the AS-IS model and the goal models. The validation process was conducted via verbal and written feedback sessions. The feedback is used to refine workflow constructions in order to understand fully the coordination requirements between entities or necessity to provide informational means to maintain situational awareness.

3.2.3. The value-chain model

By creating abstracted views of the workflows, it is possible to present these workflows in a way which allows to analyse their content without getting entangled with the details of used modelling notation. For example, abstracted views enable to detect critical dependencies and possible bottleneck situations. From the simulation perspective, an abstraction of a single workflow transforms it into a 'black box' – the functionality is observable, but the internal structures or transactions are hidden due to their irrelevance. The functionality is modified by the application of the attributable parameters[36].

The value-chain model represents a set of core processes on a high abstraction level (Signavio GmbH, 2014). This is the first step in creating a more refined overview of the underlying processes populating the DOO TO-BE model. For this purpose activities are extracted from the workflow model, see Table 3 below.

The content of the table is used to populate a value-chain model, or simply saying – a process map. Besides mapping the processes, the author has derived from the workflow model also the sequence and data flow parameters. These parameters are used to depict the control- and data-related dependencies between processes. For the sake of simplicity the sequence and data flow

[36] Time to finalize process, resource costs, probability of exceptions etc.

parameters are not given in the table as these can be derived from the DOO TO-BE model (see Appendix D) at any point.

Short-term planning	DOO shift_1	DOO flight plan preparation	DOO flight plan calculation	DOO actual flight
Short-term maintenance scheduling	_Crew_tracking_subprocess	_NOTAM_subprocess	_Runway_analysis_subprocess	ATC approval request
_Aircraft_routing_subprocess	_Desk_assignment_compilation_subprocess	Weather forecast	Coordination of signoffs (pre-flight services)	_Crew_briefing_subprocess
_DOO_schedule_subprocess	_Crew_(change)_management_subprocess	Compilation of MEL/CDL	_Flight_plan_calculation&validation_subprocess	Flight following
_Tail_assignment_subprocess	_Company_routes_management_subprocess	Coordination of signoffs (after-flight services)	_Load_plan_finalization_subprocess	_Flight_performance_monitoring_&_reporting_subprocess
_Communicate_schedule_change_subprocess	Disruption management	Passenger check-in	Passenger management (boarding)	Pre-flight check-up (crew)
_Crew_schedule_adjustment_subprocess		Passenger deboarding	Cargo loading	
		_Load_plan_compilation_subprocess	Refueling&restocking	
		Flight_selection_subprocess	Planned maintenance	
		_Dispatcher_operations_subprocess	Preflight check-up (MX)	
		_Cargo_information_processing_subprocess		
		Cargo offloading		
		Aircraft cleaning		

Table 3. Extracted processes from DOO TO-BE model, input for _Process_Map_merged_model. (Rein Nõukas)

The goal is to map the activities as processes and also their dependencies in the value chain model, see in Appendix E. A novel way is implemented in depicting the dependencies in order to enable ease of understanding of the sequence and data flow between the components inside the value chain model. Figure 44 depicts an excerpt of the aforementioned value chain model.

The connectors connecting processes by using only vertical lines determine the sequence flow. If the preceding process also produces data objects that are to be consumed by the following process then sequence flow also depicts the data flow. The data flow is depicted using a connector that has both horizontal and vertical lines.

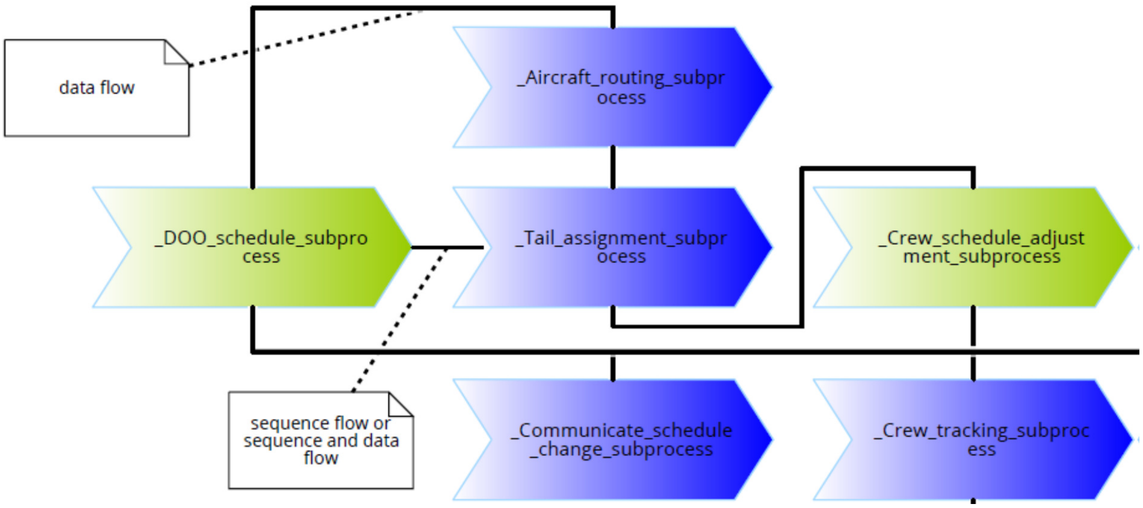


Figure 44. Depicting data and sequence flows in a value-chain model. (Rein Nõukas)

The data and sequence flows placed in the context of a value-chain model allow already to get a comprehensive overview of the workflow model without becoming entangled in the details of the business-process notation. At the same time, by including the depictions of sequence and data flows, still the desired level of detail is preserved as the core dependencies existing between processes have been revealed.

Another layer of information in the process mapping is contained in the colour patterns used for defining the nature of decision-makers. If the square depicting a process is blue, then the decision-making is done primarily by a software agent. If the square is green – a human-based agent is the main decision-maker. If the square is yellow, an unclear, or a very balanced mixture or human- and software-based agents take part in the decision-making. By defining the nature of the decision-makers it becomes visible which processes are more automatable[37] and which depend on the human factor. Also this gives an input to preparing the simulation environment as the human-based agents should embody a considerably varied behaviour compared to software-based agents and therefore provide less expected and more unexpected reactions to

[37] The author means by automation the complete removal of human intervention from the lifecycle of the process, also the start-, intermediate- and end-events should be determined and executed by software-based agents.

scenarios. This means in turn optimized efforts in defining the sets of parameters for running simulations.

By knowing the dependencies between the processes the door has been opened for conceptual designing and modelling the business layer within the enterprise architecture, also simulations could be implemented based on the process maps.

3.2.4. Defining the scope for the business process model

The value-chain model is reviewed by domain experts, and the processes, irrelevant for conducting the day of operations, are removed from further analysis. In order to put sole focus on the day of operations, the *short-term planning* phase is discarded completely. Also processes that are not directly related to pre-flight activities, like flight planning, aircraft terminal-servicing and take-off activities, are discarded. This means *company routes management subprocess* and *desk assignment compilation subprocess*, along with *planned maintenance, flight following* and *flight performance monitoring* activities are removed from the model. All data resources created during that discarded *short-term planning* phase and other discarded activities are assumed to be existing in their native locations (*resp.* information systems).

Another major change affects the phase *DOO shift_1*. As it comprises mostly of the activities forming the disruption management then a separate workflow model, located in Appendix G, is compiled. The occurrence of a disruption is very unpredictable – basically during any point of time throughout the day of operations, some kind of event can cause disruption management to be initiated. Therefore, by nature the disruption management is not suitable to be a part of a workflow model describing a flow of control from processes that need to be completed first to the processes that need to be completed last.

By removing non-essential processes from the DOO TO-BE workflow model a new workflow model, located in Appendix F and depicted in Figures 46-48 is compiled. It represents the core processes executed during the day of operations focusing solely on the following phases:

- *DOO flight preparation*
- *DOO flight calculation*
- *DOO actual flight*

By having a concrete focus set in place, the newly formed workflow model is refined further with the help from domain experts and by using source documentation. As a result, the workflow model is divided into the following domains:

Domain	Agents populating the domain
Flight preparation	NOTAM specialist, Weather specialist, Maintenance controller, Flight ops engineer, Dispatch, Weight & balance (load planner)
Turnaround	Ground operations, Passenger management, Gate agent, Crew management system
Take-off	Dispatch, Ground operations, Crew

Table 4. Workflow domains and respective agents. (Rein Nõukas)

It is necessary to put stronger attention at the activities related to turnaround (aircraft terminal-servicing) as this domain affects directly the execution of the whole workflow. While during the flight preparation and take-off mainly data is being transformed by agents and transmitted from one agent to another, then the turnaround is all about manipulating physical assets like aircraft, fuel, passengers and cargo. This means that turnaround is a very resource-intensive domain, and it requires precise coordination between the agents populating it. The coordination is vital because compared to data processing manipulating physical assets cannot be scaled up or put on hold without serious operational and financial penalties.

Another remarkable change introduced by the higher level of detail is the significant reduction of the number of subprocesses used in the workflow model. The overall number of subprocesses drops from 19 to five.

In conjunction with the two factors described above it is possible to amend the workflow model with a much higher degree of data-dependencies representing the coordination mechanisms taking place on a day of operations. Due to the variety of information systems deployed by the airlines the coordination mechanisms are kept as generic as possible in order to avoid limited applicability.

3.2.4.1. Final DOO TO-BE workflow model

In this chapter the final DOO TO-BE workflow model, located in Appendix F, is described in a more detailed way. Each of the subsections of this chapter – flight preparation, turnaround and take-off - contains one phase of the workflow model.

Prior proceeding to the workflow model the notation used in the modelling of the workflow should be familiarized. Figure 45 contains all flow elements used in the workflow model and their respective meanings in the workflow model. The notation is compliant with a set of rules defined by BPMN 2.0 framework (OMG, 2011). The modelling tool is Signavio Process Editor[38].

Note: All agents are placed in separate pools as author assumes these are independent entities.

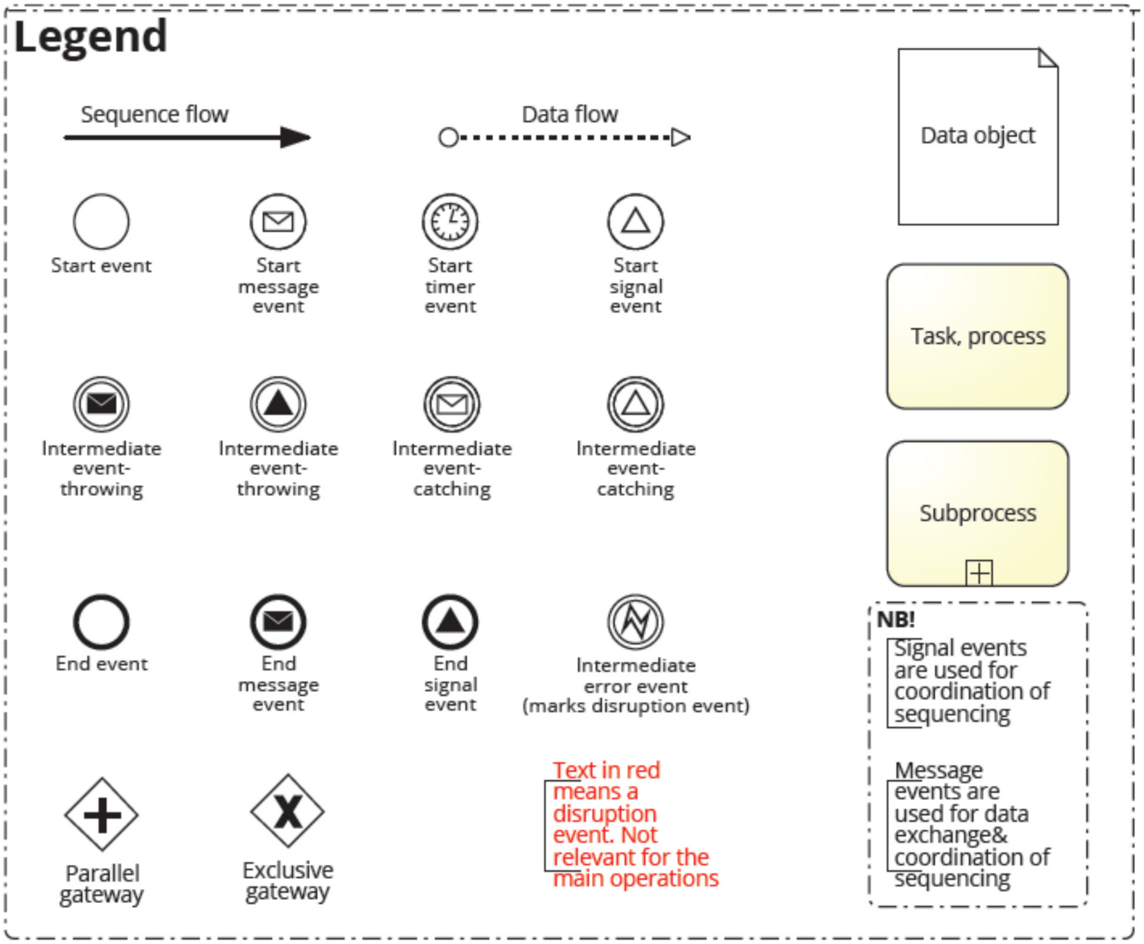


Figure 45. Description of the notation used in the DOO TO-BE workflow model. (Rein Nõukas)

Flight preparation

[38] Link to a site: http://www.signavio.com/.

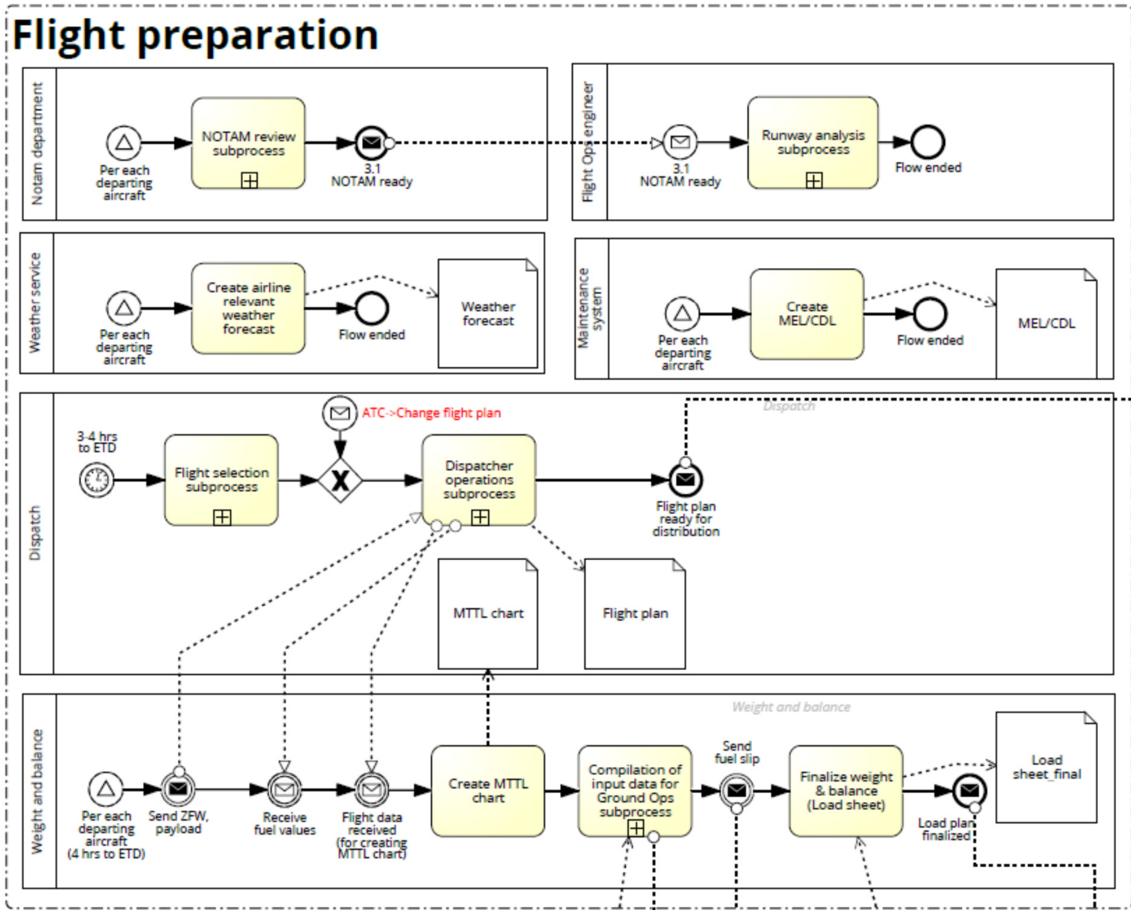


Figure 46. Flight preparation as one of the phases of the day of operation. (Rein Nõukas)

Flight preparation, see Figure 46 above, is the 1st phase of the day of operations. It is initiated 3-4 hours before the flight commences.

The most core activity here is the compilation of a *flight plan* per each departing aircraft by the *Dispatcher*. Prior executing this activity four other agents - *NOTAM specialist, Weather specialist, Maintenance controller, Flight ops engineer* - have to perform their activities and provide the results in the form of respective data sets to the *Dispatcher*.

Also *Weight and balance*, enacted by *Load planner*, has to provide *Dispatcher* certain information, in the form of *zero fuel weight, payload*, in order to receive in return *flight data* and *fuel values*. *Load planner* prepares work orders, documents *fuel slip* and *cargo assignment*, for *Ground Operations.* Based on the *Passenger Information* provided by *Passenger Management* a *load sheet* is compiled and sent to *Ground Operations*.

Turnaround

Turnaround, see Figure 47 below, is the 2nd phase of the day of operations. It starts with passenger-related activities – *passenger check-in* and *luggage check-in*. Once the aircraft has landed then the *Gate agent* informs *Ground operations* and *Passenger management* about the start of their respective activities.

Once the passengers have deplaned *cabin* and *galley servicing* and *refuelling* can start. Again, once these services have been completed new set of passengers can be boarded. *Cargo and luggage offloading and loading* is not dependent on the passengers. Instead, *Ground operations* needs to get information about the *cargo and luggage assignments* from the *Load planner* and *Passenger management* respectively.

Once both *Ground operations* and *Passenger management* have informed *Gate agent* about the completion of their activities it allows the *Crew* to proceed with the actual *take-off*.

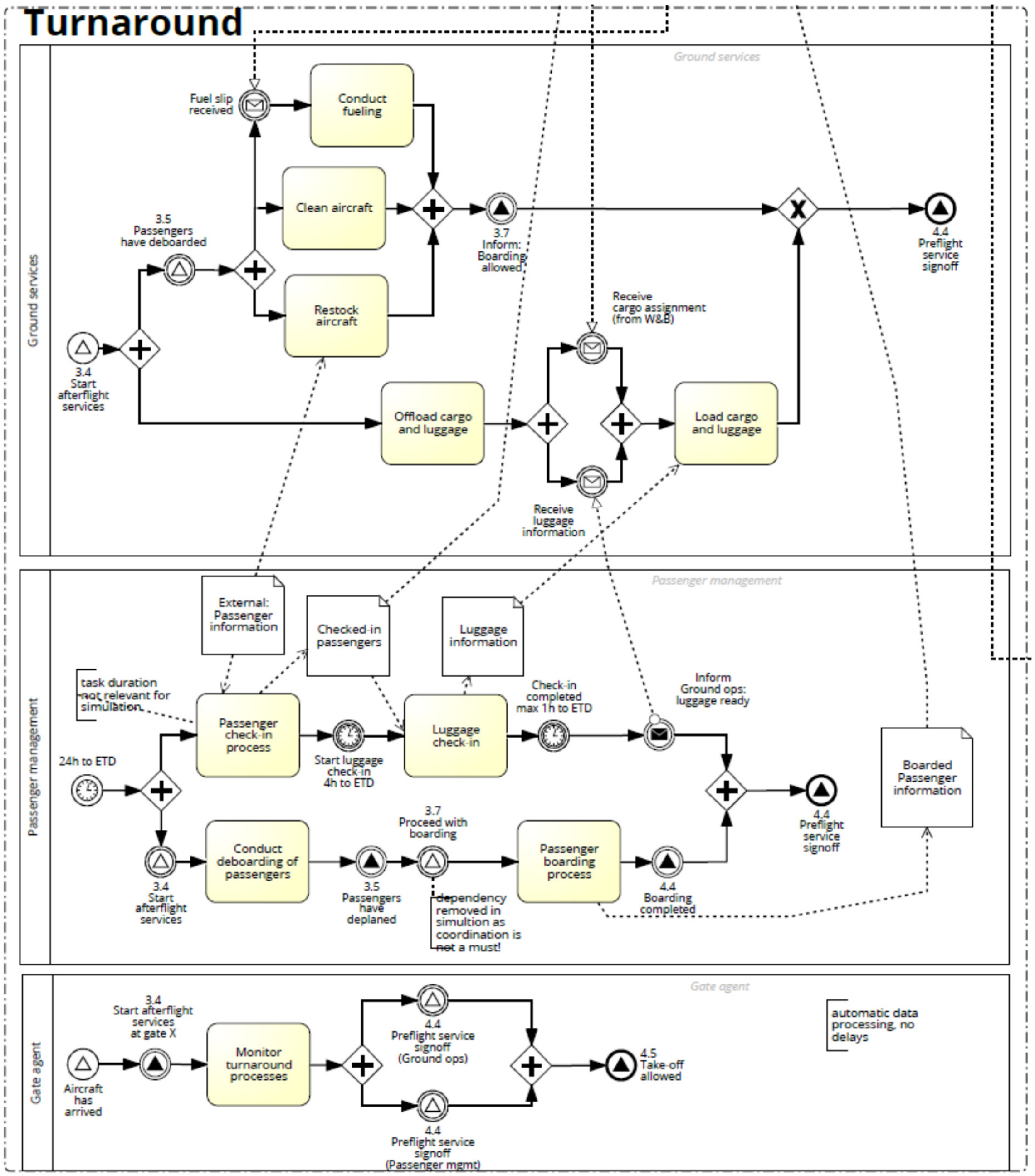


Figure 47. Turnaround as one of the phases of the day of operation. (Rein Nõukas)

Take-off

Take-off, see Figure 48 below, is the 3rd phase of the day of operations. It starts usually approximately 2 hours before estimated time of departure when *Dispatcher* executes *ATC filing* by sending a document *flight plan* for approval. Around 1 hour before estimated time of departure *Dispatcher* compiles a *briefing package* by using *flight plan* and *MTTL chart* and sends it the *Crew* assigned for the flight.

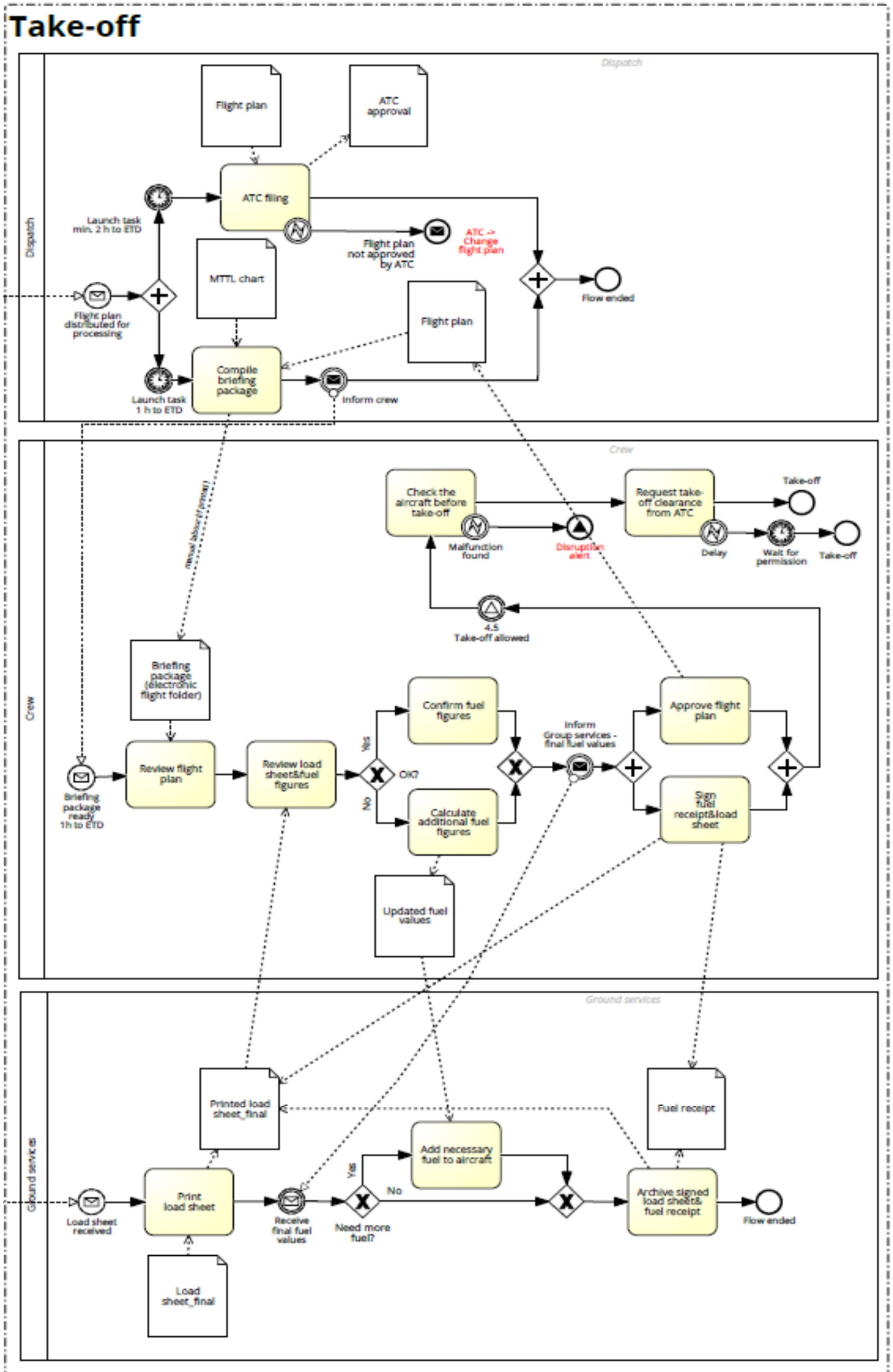


Figure 48. Take-off as one of the phases of the day of operation. (Rein Nõukas)

The *Crew* reviews the *flight plan* along with *load plan* handed over by the *Ground operations*. If the *Crew* is not approving the *fuel figures*, e.g. the amount of pumped fuel, then the *Ground operations* receives an order, in the form of a document *Updated fuel values* to conduct a task *Add necessary fuel to aircraft*.

Once *flight plan has been approved* and *fuel receipt and load plan have been signed* by the *Crew* then it is time to proceed with the *aircraft inspection*. If the *Crew* is satisfied with the condition of the aircraft *ATC* is asked for a *take-off clearance*. Based on the airport traffic volumes the *Crew* is either permitted to proceed with *take-off* immediately or it has to wait for some pre-determined time before it is allowed to enter the runway and *take off*.

3.3 Conclusion

In this chapter the AS-IS state of the business processes were identified and refined. This was done by analysing and re-modelling a day of operations of an airline.

During this chapter, the author analysed and decomposed the initial workflow model in order to propose a normalized, BPMN 2.0 compliant and amended workflow model using choreography and orchestration frameworks via Signavio. The workflow model was refined according to the best practises originating from aviation industry and respective domain experts. Also, based on the airline goal models, additional workflow constructs that were missing from the initial workflow model were added to the business process model. Lastly, a final DOO TO-BE model was compiled that contains a higher level of detail and is solely focused on the day of operations.

In the course of this chapter, BPMN 2.0 frameworks were used at varying levels of abstraction. The business process modelling was applied for describing operations of an airline at the day of operations at wide spatial and temporal scale. The aforementioned modelling provided foundation for designing workflow constructs at varying level of abstraction, from simple activities to clusters of inter-related and –dependent subprocesses. As airline operations are rather data-heavy then, instead of having directly coupled and linked processes to pass the control onwards in a workflow, a data-flow assumes the function of a quasi-sequence flow enabling effortless scalability of operations and modular design.

It is worth mentioning the primary source for the modelling of orchestrations was the choreography-based model. This way the conformity of both modelling perspectives is preserved. Additional benefit is the potential discovery of uncharted operational areas. By modelling airline's business processes at the day of operations using both choreography and orchestration it is possible to depict clearly the direct and indirect relationships and dependencies between participating cross-organizational entities, also control flows, in the form of data flows and workflows, can be easily visualized while maintaining readability and universal applicability of a diagram. Doing so is especially useful in domains which are populated by a large number of peer entities resulting in high complexity.

For the preparation of the AS-IS state and the intra-organizational business processes of an airline the following research question and two sub-questions were presented:

- RQ 1: How to normalize and map airline's business processes?
    - rq 1.1: What are the re-built airline's business processes?
    - rq 1.2: What kind of methods can be used to refine the airline's business process model?

By normalizing the airline's business processes in the form of a BPMN-based workflow model, see the final AS-IS model in Appendix C, the rq 1.1 has been answered. A full and complete understanding has been achieved with regards to the composition of the airline's business processes.

The airline's business process model, in the form of the normalized AS-IS model, was refined using the domain experts, goal models and abstract value chain model – the rq 1.2 has been answered. The refinement process provided a relevant level of detail to the airline workflows, that is, in turn, vital for organizing an airline's business processes in the real-life situations.

By amending the workflow model with the additional workflow parts from the goal models provided the first iteration of the refined workflow model that is located in Appendix D. This model enabled the emergence of the second iteration of the workflow model - the final TO-BE model, located in Appendix F - that encompasses the best practises from the industry and optimization suggestions from the domain experts. The final TO-BE model, in turn, serves the larger purpose of laying the foundations for the fulfilment of the second main research question, see RQ 2 in Chapter 1.3.4 *Research questions*, by enabling the compilation of enterprise business architecture of an airline and the evolvement of the airline's digital maturity level. The second main research question is further elaborated in the following Chapter 4.

4. Emergence of the TO-BE state

In Chapter 3 the AS-IS state of the airline operations was discussed, also it was populated with the normalized business processes in the form of the final TO-BE model. By having a solid understanding of the native components (*resp.* airline's workflow constructs), from an airline's perspective, that form part of the virtual enterprise model, it is possible to move forward to the next phase.

The next step in the process of configuring the cloud-based collaboration-enabling ecosystem for the aviation industry is the digitalization of the airline operations and providing a layout for the collaboration environment. Both activities are relevant because these form the basis of the TO-BE state, a situation that describes the structure and components of the virtual enterprise model in the aviation domain.

Therefore, the goal of this chapter is to identify the remote parts, from the airline's perspective, of the virtual enterprise model that enable the enactment of the mutual dependency between the collaborating parties instead of direct control. This goal is achieved by following the given research question:

- RQ 2: How to increase an airline's digital maturity level?

In Chapter *1.2.2 The Digital Airline Maturity Model* the airline's digital maturity is explained as the ability to utilize the means of ICT to run the business operations and to support the decision-making processes.

This research question is broken down into two sub-questions in order to distinguish the separate phases as deliverables of the research and development process:

  - rq 2.1: What are the re-organized airline's business processes?
  - rq 2.2: What is the re-configured airline's digital architecture?

Due to the lack of resources and time the author has limited the research efforts according to the perspectives as follows:

    - rq 2.1: Limitation: only collaboration perspective
    - rq 2.2: Limitation: only cloud computing perspective

The first sub-question, located in Chapter 4.1, focuses on developing the collaboration mechanism using several iterative cycles that address the business layer of the airline's enterprise architecture. Next, the second sub-question, located in Chapter 4.2, provides an abstracted view of the airline's business layer based on the deliverables of the previous chapter. Also the application and infrastructure layers have been composed in order to provide a robust overview of the components of the airline's digital architecture. Finally, a summary and conclusions in Chapter 4.3, are provided in order to understand the nature and relevance of the deliverables produced during the enactment of the second main research question (*resp.* RQ 2).

4.1 Reorganization of an airline's business processes

The discussion in Chapter 4.1 is based on the deliverable of Chapter 3 - the final DOO TO-BE workflow model, see in Appendix F. It encompasses the best practises from the industry and optimization suggestions from the domain experts. This model enables the compilation of enterprise business architecture of an airline and the evolvement of the airline's digital maturity level. The same model is considered as the basis for the AS-IS state in the context of Chapter 4.

The reorganization of the airline's business processes takes place in several iterative steps in order to achieve a desired level of abstraction of the airline's business layer. Before it is possible to start with the reorganization tasks, the understanding of the mutual dependencies of the workflow components have to be achieved. For this purpose, the final DOO TO-BE model is decomposed into atomic components that are listed in Appendix H. This list provides a verified source for the business process modelling in the sequential subchapters.

The rest of Chapter 4.1 is structured as follows. At first, it is discussed why the cloud computing, and consequently – collaborations, cannot be used in the AS-IS state of the airline's business processes. Secondly, a solution is provided in the form of the TO-BE model that enables the configuration of a cloud-based collaboration platform encompassing also the airline's business processes.

4.1.1 Defining the problem areas in the AS-IS state

In the given chapter the author detects why it is not possible to utilize cloud computing and collaborations within the architectural setup provided by the AS-IS state. As business process modelling is addressing the matters of sequence flow and data flow then the current BPMN 2.0 compliant workflow model given in Chapter 3.2.4.1. *Final DOO TO-BE workflow model* is not suitable for the further analysis. This is reasoned by the model's lack of necessary level of abstraction as it focuses on the details of the execution of each business process. Instead, the new requirement is to get an overview of the business processes without getting 'inside' of their execution mechanisms.

Additionally, during the following analysis efforts the author focuses only on the turnaround phase of the day of operations. This is based on the argument that a vast majority of the business processes in this phase are not considered as core operational competencies of an airline due to being carried out by logistics-oriented roles. These roles, enacted by the external outsourcing partners, take care of the relocation of the physical assets, like passengers, cargo, galley supplies, garbage and fuel. These services are universal in their nature as they do not depend on the airline's flight routes or operational setup. Therefore, it is reasonable to expect these business processes to be outsourced by an airline. In parallel, flight preparation and take-off, for being carried out by data processing-focused roles, are populated with processes that are mostly, if not all, part of airline's core operational competencies.

Based on the aforementioned line of reasoning, the turnaround phase is the most likely to have collaborations with the external outsourcing partners. The limitation set for the rq 2.1 states that the airline's business processes are re-organized only from the collaboration perspective. Therefore the author has modelled only the AS-IS turnaround phase, located in Figure 71 below, in a more abstract level using the ArchiMate's business layer, namely the business process co-operation viewpoint. The business process co-operation viewpoint, according to the definition given in ArchiMate (Beauvoir, 2014), is used to show the relations of one or more business processes with each other and/or with their environment. It can both be used to create a high-level design of business processes within their context and to provide an operational manager responsible for one or more such processes with insight into their dependencies.

Instead of workflows, compared to the BMPN 2.0 modelling, the focus is set on describing the business services and their relationships with other business process components. As events are not relevant in collaborations, this abstraction level is devoid of temporal parameters. This means, there are no concrete event-driven points of control in the new model. In other aspects, the description of business processes taking place in the turnaround phase is similar to the description given in Chapter 3.2.4.1. *Final DOO TO-BE workflow model.*

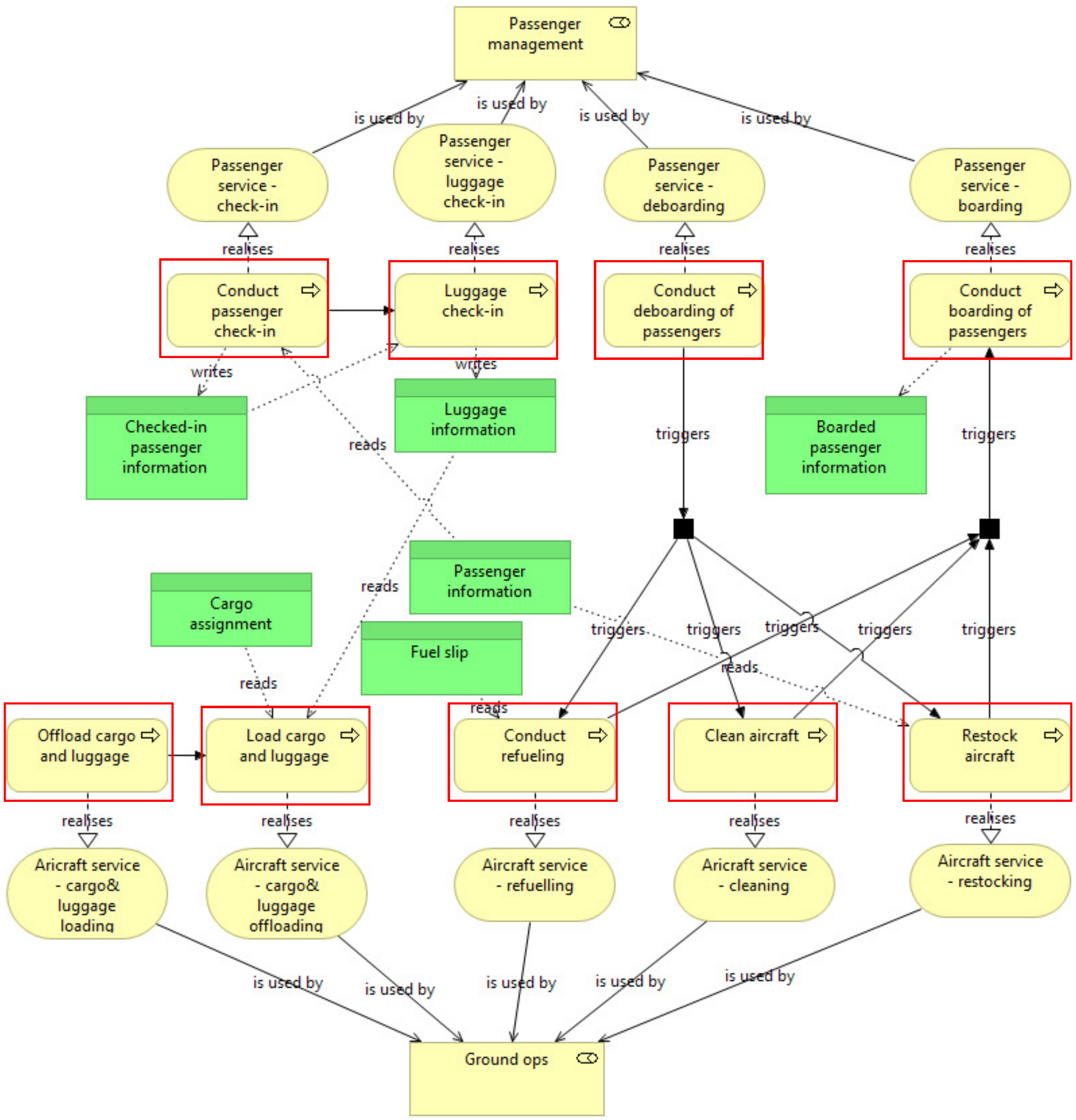


Figure 49. AS-IS turnaround phase of the day of operations, a business process co-operation viewpoint. (Rein Nõukas)

The AS-IS turnaround phase of the day of operations is populated by two business roles: *Passenger management* and *Ground operations.* Both roles use a number of business services in order to execute their respective operational responsibilities. Each business service, in turn, is realized by a respective business process. The business processes, in Figure 49, are marked with red-lined rectangles. Each business process creates by *writing* or consumes by *reading* certain business objects, coloured in green. The business objects represent data objects that are not visualized in this model. The data objects contain data that is necessary for the business processes to conduct their respective activities. Once a business process has completed its activities it *triggers* another business process. The multiple-join or -split gateways are marked as black rectangles in this figure. Gateways enable to maintain expected sequential behaviour of the business processes.

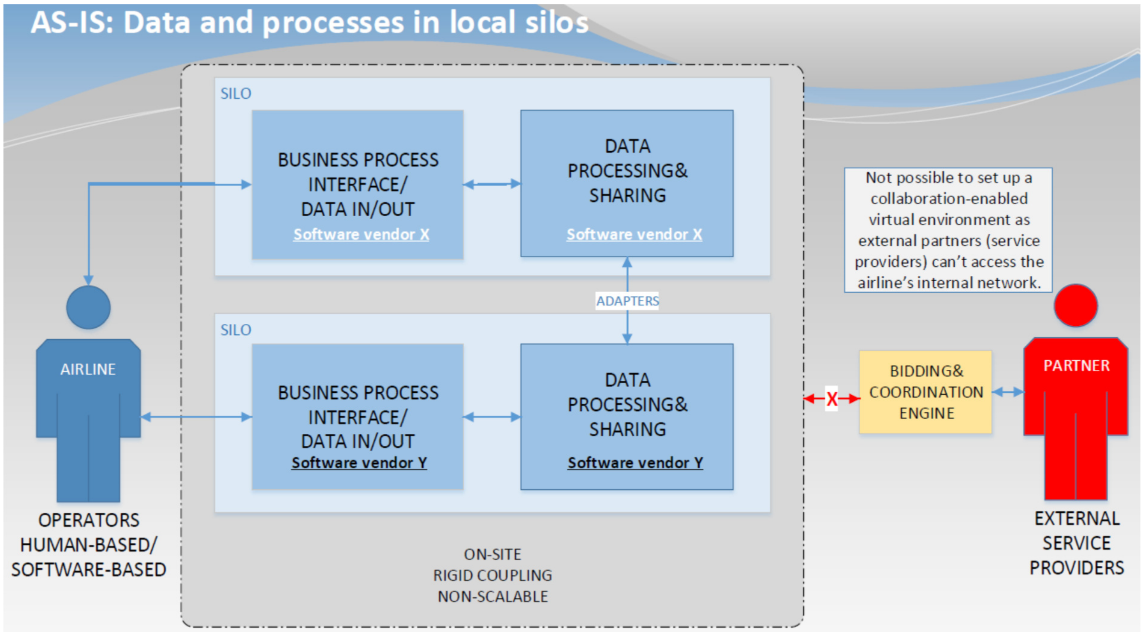


Figure 50. Conceptualization of the AS-IS model. (Rein Nõukas)

Figure 49 provided an abstracted view on how the business processes, data processing and data sharing take place in the airline's virtual domain. In order to understand if the inter-organizational collaborations could be enabled by the current configuration of the airline's information system, a very conceptual AS-IS model is compiled. The model is located in Figure 50 above. This model is relevant because it allows to achieve the required level of abstraction that allows to spot the restrictions of the current (AS-IS) abstract architecture of an airline with low digital maturity level. As was defined in Chapter 1.2.2 *The Digital Airline Maturity Model* the digital maturity is assessed by the setup of the digital infrastructure and business processes. The assessment of the setup of the aforementioned perspectives is given in sequel.

Figure 50 is compiled from the potential collaboration and cloud-computing viewpoints. In order to describe the AS-IS model by using the aforementioned viewpoints it is relevant to decompose the meaning of collaborations. Collaborations take place between a service consumer (*Airline*) and service providers (*External service providers*). The service consumer publishes service acquisition intents in a way that these become available for the service providers. If the service acquisition intent fits to the service provider's service portfolio and the necessary resources are available for the required period of time, the bid is presented in a way that it is accessible for the service consumer. The service consumer selects the most optimal bid and declares the outcome of the service acquisition instance.

In order to have properly executed collaboration, the service provider needs to be able to access information that describes the details of the consumer's public view – the part of the workflow that the *Airline* is willing to outsource, accompanied with concrete qualitative and quantitative service delivery parameters. Only information containing no business secrets or confidential information is added to the public view (Norta & Eshuis, 2010). The information flow between *Airline* and *External service provider* happens in both directions as the consumer needs to coordinate the sequence flow and data flow at certain check-points throughout the collaboration period. The service provider informs the consumer about the service execution results and deliverables achieved at some pre-defined synchronization events.

Figure 50 provides the opportunity to assess the airline's digital maturity level:

- Digital infrastructure perspective:
    - In Figure 50 the airline's abstract digital infrastructure is described. Each *Silo* contains a set of tools used for business processes and data processing, data sharing is done either via an enterprise service bus, if processes are originating from the same product family, or by utilizing custom-built adapters[39], if processes are not originating from the same product family, linking directly different silos. The *Silos* are located on-site at an airline's data centres.
- Business process perspective:
    - Additionally, as is given in the Figure 49, each business process, marked with red-lined rectangle, of an airline is kept as a single block. This way it is not possible to disclose the public view to the external service providers without the risk of revealing an enterprise's confidential information.

By having this kind of non-cloud setup for a digital infrastructure, as shown in Figure 50, and undivided business process views, as shown in Figure 49, it is not possible to enable collaborations in the time-critical and data-heavy conditions. This happens due to the inadequate access to the service consumer's data stream, also the consumer itself lacks the complete and real-time, vertically integrated understanding of the status of its business

[39] Not visualized in the Figure 72.

processes. The nature of the airline business demands very precise service provision in order to avoid negative effects on running the operations, *e.g.* delay of one take-off may result in a domino effect spanning throughout several days of flight schedules, and an airline's financial and operational performance in general.

4.1.2 Solving the problems - the TO-BE state

In the current chapter the author proposes the solutions for the shortcomings of the AS-IS model that were detected in Chapter 4.1.1. The AS-IS model, in its conceptualized form located in Figure 50, describes the architecture of an airline with low digital maturity. By having a silo-based digital infrastructure the airline is not able to create cloud-enabling conditions suitable for external partners to dynamically access former's repositories containing information about the public views of outsource-able services and business processes.

The discussion about the proposed TO-BE model is separated into three segments. At first, the author describes the conceptual model that gives a top-down view towards the collaboration-enabling digital architecture of an airline. Next, the collaboration-enabling TO-BE state of the airline's turnaround phase during the day of operations is reviewed. The third part is about the collaboration mechanism that is designed at the business layer. The proposed collaboration mechanism enables airline and its external service providers to execute streamlined and dynamic collaborations.

4.1.2.1 The conceptual TO-BE model

In order to enable collaborations, an airline has to transform its silo-based digital architecture into a cloud-based form. For this purpose, a TO-BE model, located in Figure 51 below, has been constructed. This model visually depicts a cloud-based collaboration-enabling concept that provides means for an airline to acquire a state of high digital maturity.

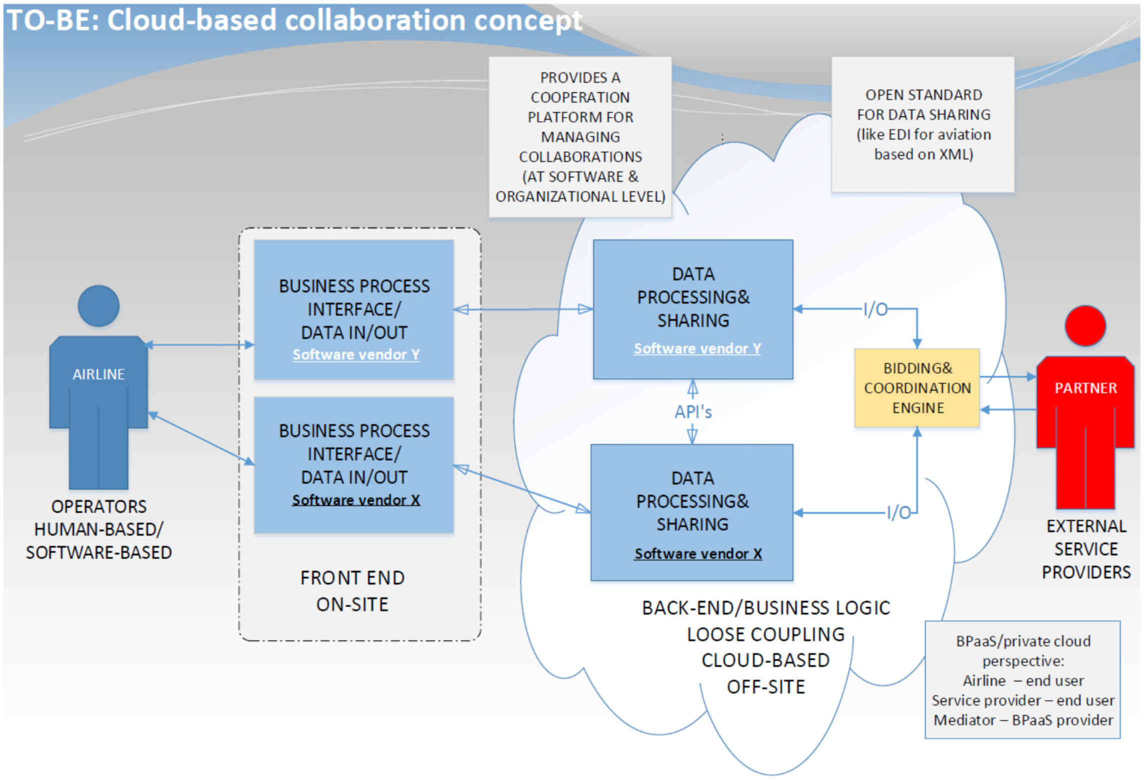


Figure 51. Conceptualization of the TO-BE model. (Rein Nõukas)

The most visible difference, compared to the AS-IS model in Figure 50, is the dissolution of silos. The back-end, underlying business logic, data processing and data sharing using API's, is located in a cloud computing environment that is hosted by a third-party service provider. Cloud computing (Mell & Grance, 2011) enables on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Compared to the back-end, the front-end, interfacing with the web-based tools operating business processes, remains on-site, at the premises of an airline.

By moving its data processing and data sharing systems at a cloud computing environment *Airline* has created a suitable cooperation platform for managing collaborations with the *External service provider* at a software level, specifically at business process level. Besides the data repositories of *Airline,* the cloud computing environment hosts the bidding and coordination engine that serves as the mediator between *Airline's* service acquisition intents and *External service provider's* bids. This way all the interaction between consumer and service provider can be automatized in order to streamline the coordination of the service acquisition and service provision.

As an airline's goal is to 'purchase' the most optimal[40] service from the pool of bidding service providers then it has to literally open itself up and create public views of its business processes that are about to be outsourced. The public views serve as sets of information describing how the outsource-able part of the business process has to be conducted. Depending on the nature of the business processes the limitations and requirements set in the public views can be binding to a service provider at varying extent. The service providers can, then, adjust to the consumer's expectations and make their bids by exposing their own public views matching the targeted consumer's public views every time an airline indicates its need for service acquisition. Norta (2015) discusses in his conference paper the novel concept of smart contracting that forms the basis for agile business-network collaborations. Similarly to this thesis he describes the usage of service-oriented cloud computing in a loosely-coupled collaboration lifecycle.

4.1.2.2 The TO-BE state of the business layer

In order to enable inter-organizational collaborations as depicted in Figure 51, an airline has to divide each of its outsource-able business processes into internal and public parts (*resp.* views). The private view contains airline's unique or, at least, domain-specific knowledge of how to run a certain operation. Also, it contains parts of the business process that is not outsourced.

The public view, in turn, contains airline's expectations regarding the ways the outsourced part of business process has to be conducted. And, if necessary, there are defined the coordination and synchronization events during the execution of the business process. The extraction of public elements from the abstract business process has been elaborated by Norta and Eshuis (2010).

Norta and Eshuis (*ibid.*) describe possible projection relations that may exist between a process view and a conceptual process. Each projection relation is realized using a combination of the projection rules. There are in total six (6) combinations, describing the range of the combinations of the projection rules. On the one hand, there is a black-box projection that aggregates (or hides) all the activities that are located inside service consumer's internal process into a single observable activity. At the opposite side there is a white-box projection, according to which the public view is identical with the service consumer's internal process and no abstraction is used. The description of the combinations that are located between the

[40] Completion-, quality- and cost-wise.

aforementioned extremes is omitted as the projection rules and their application to setting up collaboration-enabling environment is out of the scope of this thesis.

To sum it up, the public view has to reveal the airline's pre-set requirements for service provision, providing quantitative and qualitative parameters for external service providers.

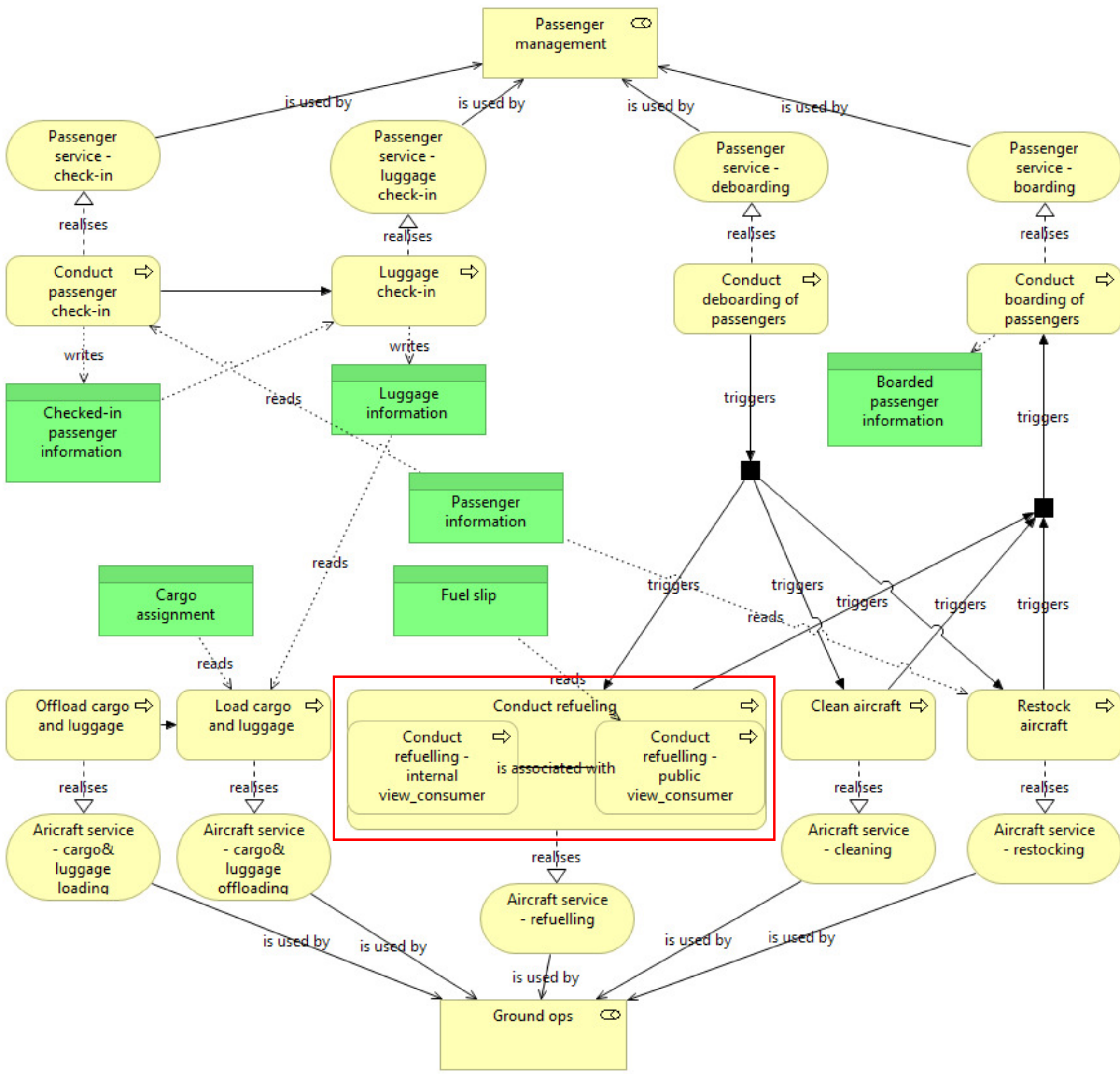


Figure 52. TO-BE turnaround phase of the day of operations, a business process co-operation viewpoint. (Rein Nõukas)

Figure 52 depicts the division of a business process into an internal and public view. As all the depicted business processes in the turnaround phase are treated as eligible for outsourcing then, for the sake of simplicity and ease of comprehension, the author focuses on the business process '*Conduct refuelling'* in this model. Both views - '*Conduct refuelling - internal view_consumer'* and '*Conduct refuelling - public view_consumer*' - are still part of a single process unit '*Conduct refuelling'* by being its sub-processes. However, these views or sub-processes are treated differently, only public view has the access to business object '*Fuel slip*'. This kind of differentiation is reasoned by the argument that the '*Fuel slip*' contains a set of concrete non-confidential work orders, a set of data, derived from the operational situation an airline must execute. Also, this data is vital for the external service provider in order to assess the parameters of the expected service provision and, in turn, prepare the respective bid to be assessed by the service consumer, the airline.

4.1.2.3 The proposed collaboration mechanism

By having separated the internal and public views into separate sub-processes, the airline has re-organized its business processes for outsourcing. Outsourcing is an orchestrated transaction, a collaboration between the consumer and service provider(s). The following discussion focuses on the author's attempt to design and reason the possible collaboration mechanism.

The attention is reserved mostly for designing and describing the business layer of the collaboration mechanism. The application and infrastructure layer are subject of further research, thus not covered at full extent in this thesis.

Each collaboration instance has two major phases:

- Bidding, located in Figure 53 below
- Coordination, located in Figure 54 below

In the TO-BE state, see Figure 53, there is a special application called 'C*ollaboration mediation platform*' that is enacted by an application service '*Collaboration mediation service_mediator*'. The platform is located in a cloud computing environment that is accessible to both collaboration counterparts according to Figure 51 – *Ground ops,* being a part of *Airline operations*, and *External service provider.* Both parties have designated business functions that operate respectively *outsourcing*- and *insourcing*-related activities.

Another application component '*Airline operation platform*' is enacted by the application service '*Collaboration coordination service_consumer*' that is a subset of another application service '*Airline operations service*'. The latter in general and the former in particular form the basis of the airline's capability to interact with the collaboration mediation platform by using designated business processes. The application layer and the infrastructure layer designated for the external service provider are not described here as it lies beyond the scope of this thesis.

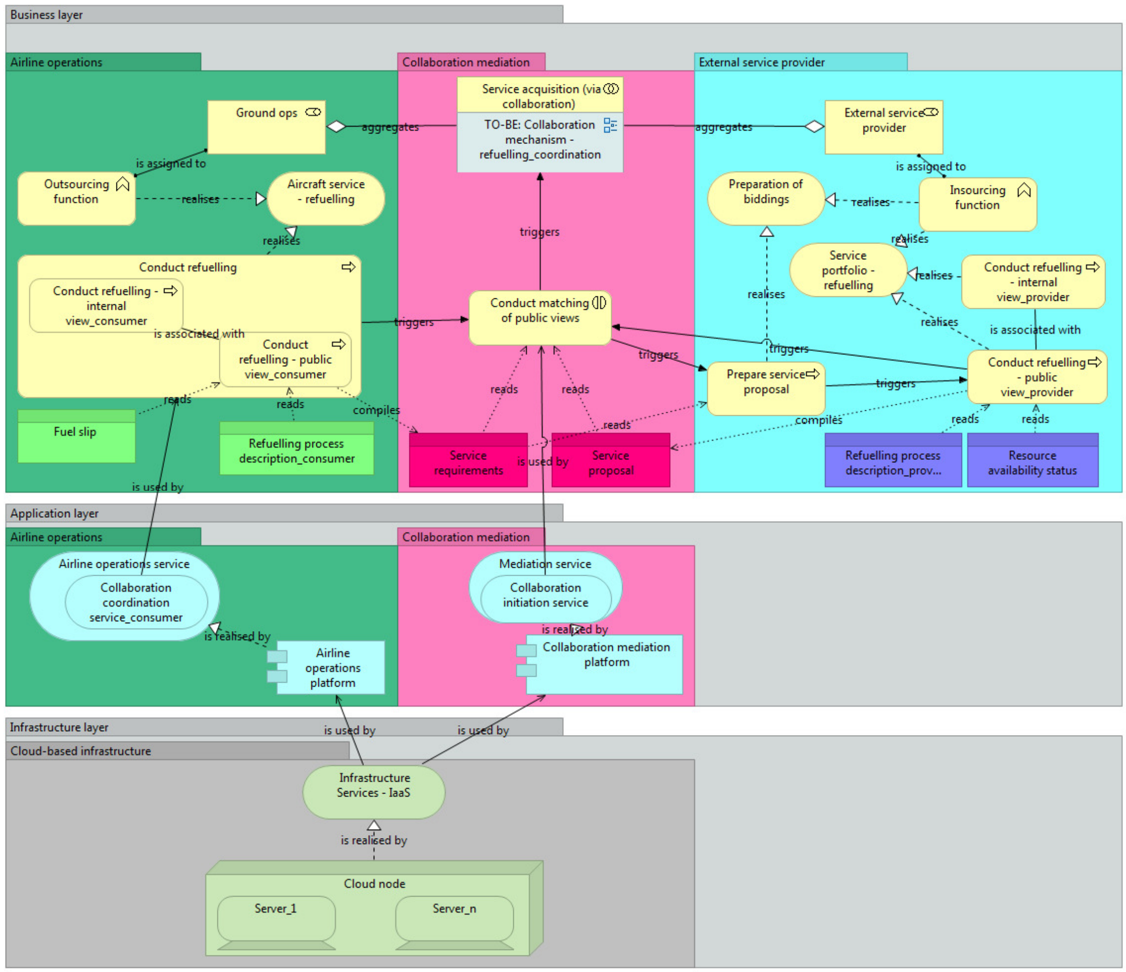


Figure 53. The bidding part of the collaboration mechanism. (Rein Nõukas)

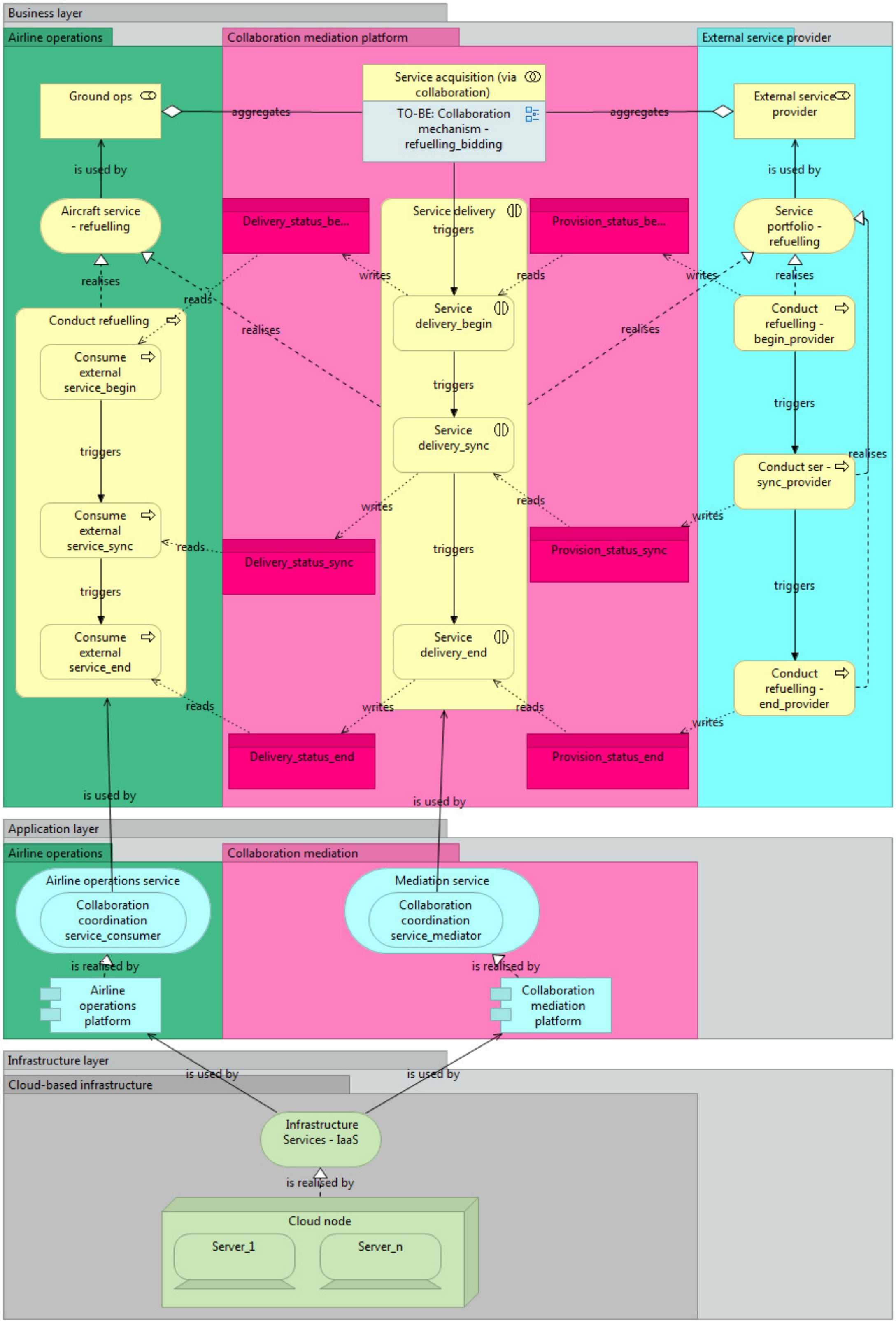


Figure 54. The coordination part of the collaboration mechanism. (Rein Nõukas)

As was discussed already in Chapter 4.1.2.2 *The TO-BE state of the business layer* a business process to be taken as a sample is '*Conduct refuelling*'. This business process realises the business service named as *'Aircraft service – refuelling'*. As the airline has separated the business process into public and internal views there are two mutually associated sub-processes – *'Conduct refuelling - internal view_consumer'* and *'Conduct refuelling - public view_consumer'*.

Once the sub-process *'Conduct refuelling - public view_consumer'* becomes aware of the new business object *'Fuel slip'* it compiles a data object '*Service requirements'*. This data set includes both qualitative and quantitative parameters regarding the service in acquisition, including concrete work orders from the business object '*Fuel slip'* and expected process description from the data object *'Refuelling process description_consumer'*. Internal process projection options are discussed thoroughly in Chapter 4.1.2.2 *The TO-BE state of the business layer*. In order to enable the business interaction instance '*Conduct matching of public views'* to evaluate the posted bids (issued in the form of data object *'Service proposal')* according to the consumer's, resp. *Ground ops*, prerequisites, a set of preferences as evaluation criteria are also included in the *'Service requirements'*. Once ready, this XML-based data object is posted to the collaboration mediation engine's repository that launches, in turn, the business interaction instance '*Conduct matching of public views'*.

The *External service provider*'s insourcing business function is realised by the following business services – '*Preparation of biddings'* and *'Service portfolio – refuelling'*[41]. A bid, in the form of a data object *'Service proposal'*, can be prepared by the business process '*Prepare service proposal'* once the airline has published *'Service requirements'* in the collaboration mediation engine's repository as the published service acquisition intent triggers the bidding preparation process.

In order to successfully complete the bidding, the business process '*Prepare service proposal'* needs to access data objects *'Refuelling process description'* and *'Resource availability status'*. The former provides a concrete description of the components of the workflow the service provider needs to deploy in order to conduct refuelling. The latter data object provides the up-to-date list of the resources and their predicted availability during the execution period of the service in question.

[41] As the title indicates, the *'Service portfolio – refuelling'* is part of a larger service portfolio covering all the service provision domains of the *External service provider*.

Biddings from the service providers and requirements posted by the consumer are assessed during the business interaction '*Conduct matching of public views'*. Upon choosing the most optimal bid that meets the evaluation criteria, a business collaboration instance '*Service acquisition (via collaboration)*' will be triggered.

At this point the bidding phase hands over the control to the next phase in the collaboration instance – coordination. The main purpose of the coordination phase is to keep both parties synchronized during the enactment of the collaboration instance. The specific events, taking place at spatial and temporal sphere, during which the coordination has to be executed are defined by the consumer. The information about these events is manifested, see Figure 75 above, by the *'Service requirements'* to be consumed by the *Collaboration mediation platform* and *External service provider*.

The coordination phase, see Figure 54 above, is enacted by the business interaction instance '*Service delivery'*. It is maintained by a special application called C*ollaboration mediation platform* that is enacted by a *Collaboration mediation service*. The platform is located in a cloud computing environment that is accessible to both collaboration counterparts according to Figure 51 – *Ground ops* (resp. *Airline*) and *External service provider.*

Both the *Airline*'s business service '*Aircraft service – refuelling'* and the *External service provider*'s business service '*Service portfolio – refuelling'* use the business interaction instance '*Service delivery'* in order to materialize their respective goals – consume a service and provide a service. Service consuming is realised via three subsequent business processes: '*Consume external service_begin'*, '*Consume external service_sync'* and '*Consume external service_end'*. Service provision is realised via three subsequent business processes: '*Conduct refuelling - begin_provider'*, '*Conduct refuelling - sync_provider'* and '*Conduct refuelling - end_provider'* that are part of the general business process *'Conduct refuelling*. The latter itself is a realization of a business service *'Aircraft service – refuelling'*.

Before describing the abstract workflow during the coordination phase, it is important to re-emphasise the scope of the current modelling effort – only the collaboration-related elements belonging to the operational sphere of the parties are included to the models located in Figure 53 and Figure 54 above.

Upon the initiation of the collaboration instance the *External service provider* posts a data object '*Provision_status_begin'* to the collaboration mediation engine's repository. As per the consumer's preferences the content of this data object is converted to another data object

'Delivery_status_begin' and made available for the consumer via business process '*Consume external service_begin'*.

During the collaboration, the service provider synchronizes the status of the service provision by compiling a data object '*Provision_status_sync'* and posts it to the collaboration mediation engine's repository. This activity can be carried out at innumerate times according to the consumer's pre-set pattern given in the data object '*Service requirements'*. The consumer can access the data object *'Delivery_status_sync'* and update according to the received information its internal business process statuses and trigger another workflows if needed.

Finally, upon completion of the service provision instance, the *External service provider* posts via a business process '*Conduct refuelling - end_provider'* a data object '*Provision_status_end'* to the collaboration mediation engine's repository. This data object is converted to another data object *'Delivery_status_end'* and made available for the consumer via a business process '*Consume external service_end'*. The consumer can access the data object *'Delivery_status_end'* and update according to the received information its internal business process statuses and trigger another workflows if needed. The *External service provider* can use, in turn, the data object '*Provision_status_end'* to update the content of another data object *'Resource availability status'* (located at Figure 53) in order to be prepared for any upcoming bidding occasions. Basically, this means the resources that were used to provide a service receive an idle status.

4.2 Reconfiguration of airline's digital architecture

The reconfiguration of an airline's digital architecture is the continuation of Chapter 4.1. The deliverable of the latter - the proposed collaboration mechanism, located in Figure 53 and Figure 54 – is used in order to derive an abstracted view of the airline's business layer. By having defined the abstract layout of the business layer of an airline from the collaboration perspective, the application and infrastructure layers can be composed in order to provide a robust overview of the components of the airline's digital architecture.

4.2.1 The abstract business layer of an airline

Due to its data-heavy nature the collaborations inside the virtual enterprise's system environment requires very fast-paced and well-optimized data flow. Therefore the efforts for defining the abstract business layer of an airline are focused on depicting the data flow. The other aspects of the abstract business layer are out of focus for this thesis.

In this chapter the author describes in detail the data flow during a service outsourcing instance between a consumer, a mediation platform acting as a virtual marketplace and a service provider. The content of the models and the following discussion is based on the content from Chapter 4.1.2.3 *The proposed collaboration mechanism*.

Figures 53 and 54 already describe the data flows between and inside *Airline, External service provider* and *Collaboration mediation platform* using the refuelling service as a sample. In the turnaround phase, other services have the logic of execution similar to the refuelling service. Therefore, the author assumes the following discussion to be applicable also in the case of other services populating the turnaround phase of the day of operation of an airline.

Next, the author focuses on providing a data flow description from a more abstract perspective in order to present a universally applicable design pattern in the context of enabling the collaborations. As it is seen in Figure 55, every outsourcing instance begins with the creation of a consumer's public view, *resp.* a data object 'Service requirements', by a business process *'Compile public view_consumer'*. For this purpose, a consumer has to have prepared a description of a requested service in the form of two data sets – *'Consumer process_input'* and *'Consumer process description'*. The former provides a list of qualitative and quantitative parameters covering questions like what, when, how much and many. Whereas the latter defines if and to what extent the service provider has to follow the predefined workflow pattern.

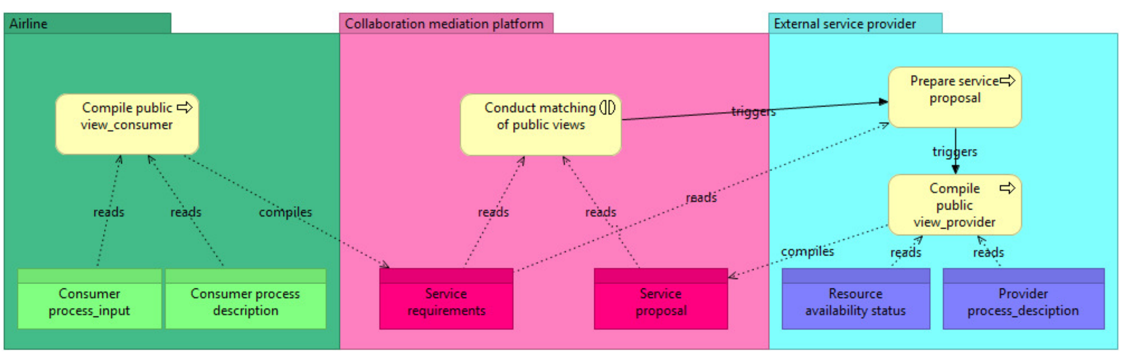


Figure 55. Abstract data flow model for bidding. (Rein Nõukas)

Data object '*Service requirements*' is posted in the service mediation platform's repository. Once the mediation platform becomes aware of the new service acquisition intent it informs the service provider and, thus, triggers the latter's internal process '*Prepare service proposal*'. This process accesses the *'Service requirements'* and initiates another process - '*Compile public view_provider*' - in order to produce a bid in the form of a data object '*Service proposal*'. This data object is, then, posted in the service mediation platform. For successful completion of a bid, the business process '*Compile public view_provider*' needs to access data objects *'Provider process description'* and *'Resource availability status'*. The former provides a concrete description of the components of the workflow the service provider needs to deploy in order to conduct an outsourced service. The latter data object provides the up-to-date list of the resources and their predicted availability during the execution period of the service in question.

In order to conduct a business interaction instance, the service collaboration platform has to access both the '*Service requirements*' and *'Service proposal'*. The platform matches the consumer's outsourcing intent and bids from service providers. The bid, that is most optimally meeting the consumer's pre-set evaluation criteria, will be declared as winner. More detailed discussion regarding the smart contracting in the context of service-oriented cloud computing in a loosely-coupled collaboration lifecycle is provided by Norta (2015).

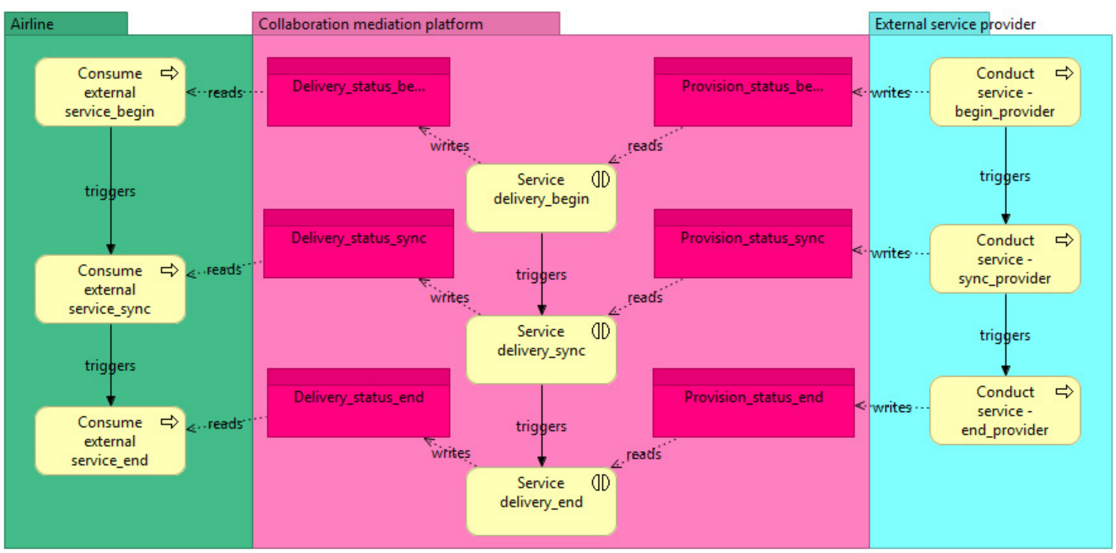


Figure 56. Abstract data flow model for coordination. (Rein Nõukas)

Once the outsourcing partner has been selected and declared, the collaboration instance moves forward to the coordination phase (see Figure 56 above). Upon starting the service provision, the service provider informs the consumer of the status of the service provision. For this purpose, provider's process '*Conduct service – begin_provider*' sends a message '*Provision_status_begin*', in the form of a data object, to the mediation platform. The latter, then, converts the message to another data object '*Delivery status_begin'* and forwards it to the consumer.

During the service provision, the service provider informs the consumer of reaching and completing certain pre-set milestones in order to enable the consumer to stay synchronized. For this purpose, the provider's process '*Conduct service – sync_provider'* sends a message '*Provision_status_sync',* in the form of a data object, to the mediation platform. The latter, then, converts the message to another data object 'D*elivery status_sync'* and forwards it to the consumer.

After completing the service provision, the service provider informs the consumer of the completeness of the service provision. This way the state of the service provision instance is updated. For this purpose, the provider's process *'Conduct service – end_provider'* sends a message '*Provision_status_end',* in the form of a data object, to the mediation platform. The latter, then, converts the message to another data object '*Delivery status_end'* and forwards it to the consumer.

4.2.2 The architecture model

In this chapter the author abstracts away from the detailed view given in Chapter 4.1.2.3 *The proposed collaboration mechanism* in order to shift the focus from the business layer to the application and infrastructure layer while describing the layout of the proposed collaboration mechanism. The goal is to present the expected composition of the collaboration initiation and implementation coordination procedures in the cloud-based BPaaS environment. In order to describe the abstract architecture of the collaboration mechanism a model is constructed. The model is located in Figure 57 below.

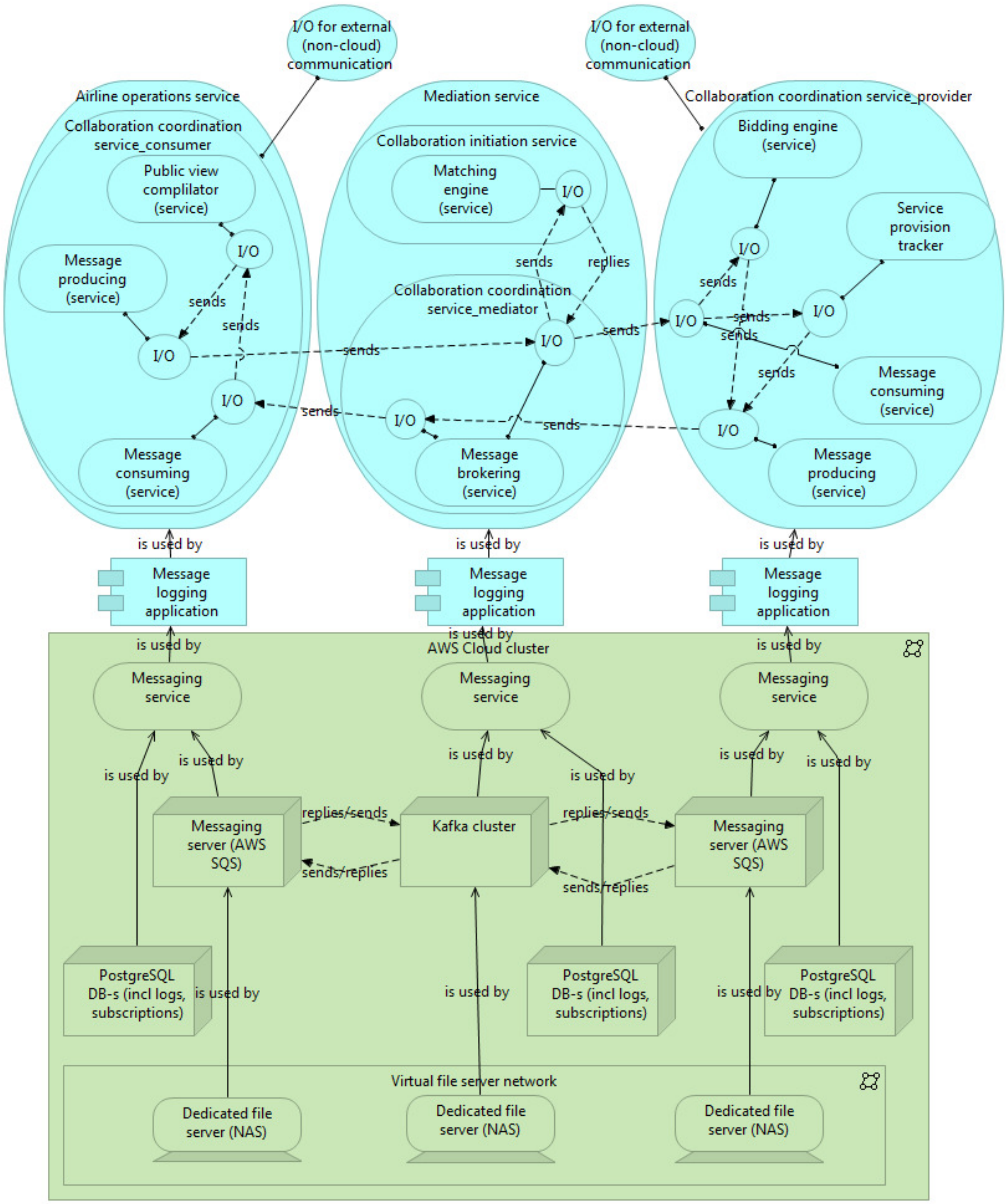


Figure 57. The abstract architecture of the collaboration mechanism. (Rein Nõukas)

The Figure 57 depicts the configuration of the collaboration-enabling cloud-based solution using abstract system components. The architecture consists of the application layer and the infrastructure layer, the modelling tool used is ArchiMate.

There are three main classes of services and their respective owners:

- *Airline operation service,* owned by an airline

- Me*diation service,* owned by a collaboration mediator
- *Collaboration coordination service,* owned by an external service provider

All service classes consist of function-specific services, like *Public view compilator, Matching engine, Service provision tracker* and *Bidding engine*, and the supporting utility services, like *Message consuming, Message brokering* and *Message producing*, that are used for messaging. The operational features (*resp.* the business layer) of the function-specific services are described thoroughly in Chapter 4.1.2.3 *The proposed collaboration mechanism*. The services interact with each other via the API[42]-based interfaces marked as *I/O, Input* or *Output*. The direct coupling, point-to-point connections, of message producers and consumers is avoided by using *Message brokering service* that enables the utilization of publish–subscribe messaging design pattern.

According to Codeproject (Paul, 2009), in a topic-based publish-subscribe pattern (see Figure 58 below), sender applications tag each message with the name of a topic, instead of referencing specific receivers. The messaging system then sends the message to all applications that have asked to receive messages on that topic. Message senders need only concern themselves with creating the original message, and can leave the task of servicing receivers to the messaging infrastructure. In this pattern, the publisher and subscriber can communicate only via messages. The publish-subscribe pattern (*ibid.*) solves the tight coupling problem. It is a very loosely coupled architecture, in which senders do not even know who their subscribers are.

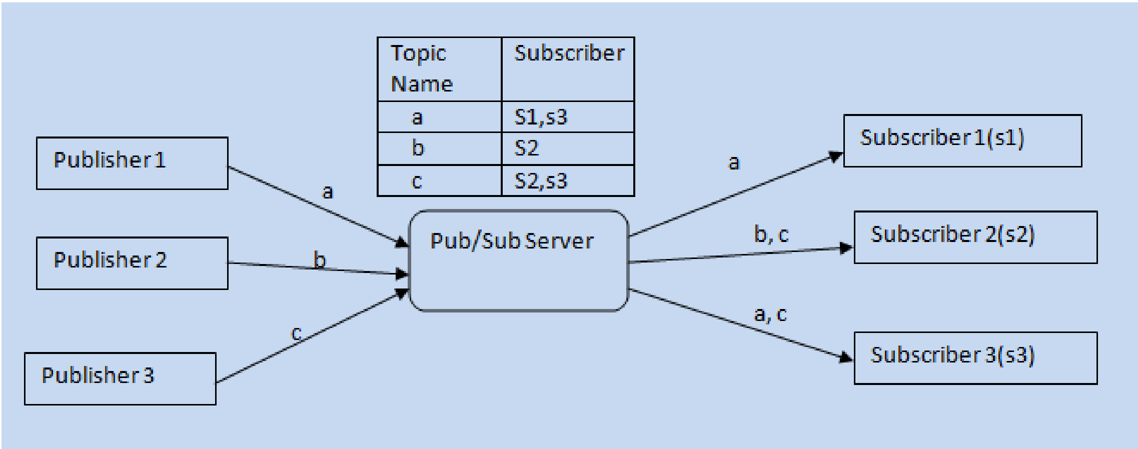


Figure 58. Topic-based publish-subscribe messaging design pattern. (Paul, 2009)

[42] API, an abbreviation of application program interface, is a set of routines, protocols, and tools for building software applications. The API specifies how software components should interact. (Beal, 2014)

Additionally, this pattern enables highly scalable architecture for the proposed BPaaS solution as the amount of collaboration instances per certain time frame is not restricted by any manual configuration-related constraint.

Collaboration coordination service_consumer, a subclass of *Airline operations service,* and *Collaboration coordination service_provider* both have similarly configured an interface for communicating with the external entities that are located outside of the cloud instance. For example, airlines may have several legacy tools and manual processes that provide input data for the collaboration instances. The status updates from the service provider's crews, like dealing with passengers, maintenance, cargo etc., are uploaded via the I/O interface as well.

All service classes have a similarly configured interaction solution in the infrastructure layer. This is reasoned by the intensive message exchange between the counterparts in the collaboration instance, see detailed data flow descriptions in Chapter 4.2.1 *The abstract business layer of an airline*. Application component *Message logging application* is used by the services to issue tasks to *Messaging service.* In turn, that infrastructure service issues tasks to *Messaging server* and *object-relational databases*.

The enactment of the topic-based publish–subscribe messaging design pattern is carried out by utilizing the Apache Kafka cluster. *Messaging servers* assigned to an airline and service providers interact with the *Kafka cluster* in order to:

- *Post* messages to the subscribed topics, if they are acting as producers of the messages;
- *Fetch* messages from the subscribed topics, if they are acting as consumers of the messages.

Apache Software Foundation (2014) describes Kafka as a distributed, partitioned, replicated commit log service. It provides the functionality of a unified, high-throughput, low-latency messaging platform for handling real-time data feeds. Communication between the clients and the servers is carried out with a simple, high-performance, language agnostic TCP protocol.

According to the Apache Software Foundation, a topic is a category or feed name to which messages are published. Topics act like catalogues for allocating the messages per predefined criteria, covering all aspects of the interaction inside the virtual enterprise. The most important instances and their respective initiators, that receive a separate topic, are as follows:

- Airline:

    - Generic cooperation requirements, generated by an airline operations service;
- Service:
    - Specific cooperation requirements, generated by *Airline operations service;*
    - Service acquisition instance:
        - Acquisition intent, generated by *Airline operations service;*
        - Bids per intent, generated by *Mediation service;*
        - Matching results ,generated by *Mediation service*;
        - Collaboration coordination, generated by *Mediation service*.

Both *messaging servers* and the *Kafka cluster* share the same awareness of the structure of the topics. The list of valid and available topics are maintained and updated by the *Kafka cluster*. The topics are updated every time there is an airline-initiated change, *e.g.* the SLA is updated, in the parameters that have been earmarked as a separate topic, or there is a new service acquisition instance posted by the airline.

Messaging servers use *dedicated file servers* to store and access the messages themselves. In parallel, the *object-relational databases* store the data of the logs and subscriptions to the topics. Accurate and up-to-date indexing of messages enables the *Kafka cluster* to allocate them to respective topics based on the index attributed to the messages.

According to the author's perception, the most essential requirement for the *cloud cluster* is its ability to process in real-time the multi-directional message flow between participants of the virtual enterprise. The size of a single message file is modest both in relative and nominal terms as only machine-readable, *f. ex.* XML-based, information is shared among the parties populating the virtual enterprise. Despite the small size, the messages are very numerous as messaging is the only way for the members of the virtual enterprise to keep each other up to date and maintain situational awareness. Therefore, the *file servers* have been placed into the same *virtual file server network*, a sub-cluster inside the cloud cluster, to enhance the data transfer speed.

4.3 Conclusions

In Chapter 4 the TO-BE state of the airline's business processes was specified. Also, the digital architecture of an airline was designed in order to enable the emergence of the B2B collaboration environment that integrates the transactions of the service consumer and service provider at a software level.

At first the applicability of cloud computing and the enactment of collaborations were analysed in the context of the AS-IS model. As this model depicts the airline with low digital maturity the silo-based business process tools and on-site computing resources deny any kind of meaningful automated collaboration with external service partners. In order to overcome the insufficient situation originating from the business layer that was based on the AS-IS model, a new cloud-enabling design for the business layer was proposed. The reorganized business processes enable 'to open up' the airline to the external partners by utilizing the public and private view of a process.

Next, the reorganized business layer was used in order to derive relevant data flow patterns that are universally applicable for the participants of the collaboration environment. Finally, an effort was made to compile the application and infrastructure layer of the airline, also partially the architectural constructs were provided for the collaboration mediator and the external service partner. All these layers were needed in order to identify the remote parts of the virtual enterprise model.

In order to enable the emergence of the TO-BE state of the airline's business processes and the airline's digital architecture the following research question and two sub-questions were presented:

- RQ 2: How to increase an airline's digital maturity level?
    - rq 2.1: What are the re-organized airline's business processes?
    - rq 2.2: What is the re-configured airline's digital architecture?

By using the business process co-operation viewpoint it was possible to create an abstract business process model that enabled the introduction of the private and public views of the business processes. This, in turn, allowed to compose a collaboration-enabling process view encompassing the airline, mediator and service provider. By having both the bidding and the coordination part of the collaboration mechanism defined and delivered, the TO-BE state of the airline's business processes was achieved – the rq 2.1 is answered.

The airline's digital architecture, *resp.* the business, application and infrastructure layers, was based on the TO-BE state of the airline's business processes. An important role was played by the inclusion of the cloud computing related concepts in the reconfiguration of the architectural constructs. By the provision of the collaboration-enabling digital architecture of an airline, encompassing also the other key entities of the collaboration environment – mediator and service provider, the rq 2.2 is answered.

Chapter 4 has enabled the finalizing of the transformation of an airline with low digital maturity, a result of Chapter 3, to become a vertically integrated organization that is aware of its interdependent and sequential data flows and workflows. The result of Chapter 4 – the digital maturation of an airline - brought along the dissolution of the silo-based architecture and the built-in ability to use public views for facilitating the collaborations. This, in turn, enables an airline to transform oneself further into an open, by using public views of processes, and networked, by enabling inter-organizational collaborations, virtual organization. This kind of configuration already resembles the concept of decentralized autonomous corporation (Buterin, 2014) – a novel way to govern the transactions of its members and organize the decision-making process. By having formed the abstract model of a digitally mature airline the RQ 2 is considered to be answered.

The author acknowledges that the constructs provided in this chapter are abstract in their nature and require a thorough further analysis in the context of a proper case study. The case study must be based on real quantifiable and qualitative parameters in order to verify the applicability of the proposed constructs.

The end state during the evolution of the digitally mature airline is achieved when the airline becomes part of a virtual enterprise, a networked ecosystem of customers, mediators and suppliers that share the resources for the benefit of all its members. The author uses the BPaaS concept to characterize the nature of that virtual enterprise in the commercial setting. The emergent nature of the BPaaS-based solution in the context of this thesis is further discussed in Chapter 5.2 *Emergence of a new business model & B2B ecosystem.*

5. Discussion

Chapters 3 and 4 have given a detailed understanding of how to transform an organization with limited or missing awareness of its business processes into an entity that is integrated both from intra- and inter-organizational perspective. One of goals of this chapter is to give a top-down view towards the configuration of the cloud-based collaboration-enabling model for the aviation industry from an airline perspective by presenting a walkthrough that depicts the digital maturation transformation process of an enterprise. Additionally, the commercial viability of the model is presented in the form of the emergence of a new BPaaS-based business model and B2B ecosystem.

5.1 Transformation heuristics[43]

By utilizing the practical experience derived from the design and modelling efforts in Chapter 3 and 4 the author abstracts away from the details of the aviation industry and presents a walkthrough that can be used to enhance the organizational digital maturity as defined in Chapter 1.2.2 *The Digital Airline Maturity Model*. An expected co-result of this transformation is the organization's built-in ability to engage into business process level collaborations with external partners by using respective virtual mediators or service brokers enabled by the cloud computing environment.

This chapter provides an aggregated and systemic knowledge of how to solve the challenge of organizational transformation using the business process perspective. Given the nature of the design science method the transformation process is expressed as an abstract method. The emerging cloud computing technologies provide novel and commercially sensible ways for configuring the business processes at intra- and inter-organizational level thus heralding the era of virtual enterprises - very flexible and competitive ecosystems of peer organizations.

[43] Definition of heuristics: using experience to learn and improve; involving or serving as an aid to learning, discovery, or problem-solving by experimental and especially trial-and-error methods. (Merriam-Webster, 2015)

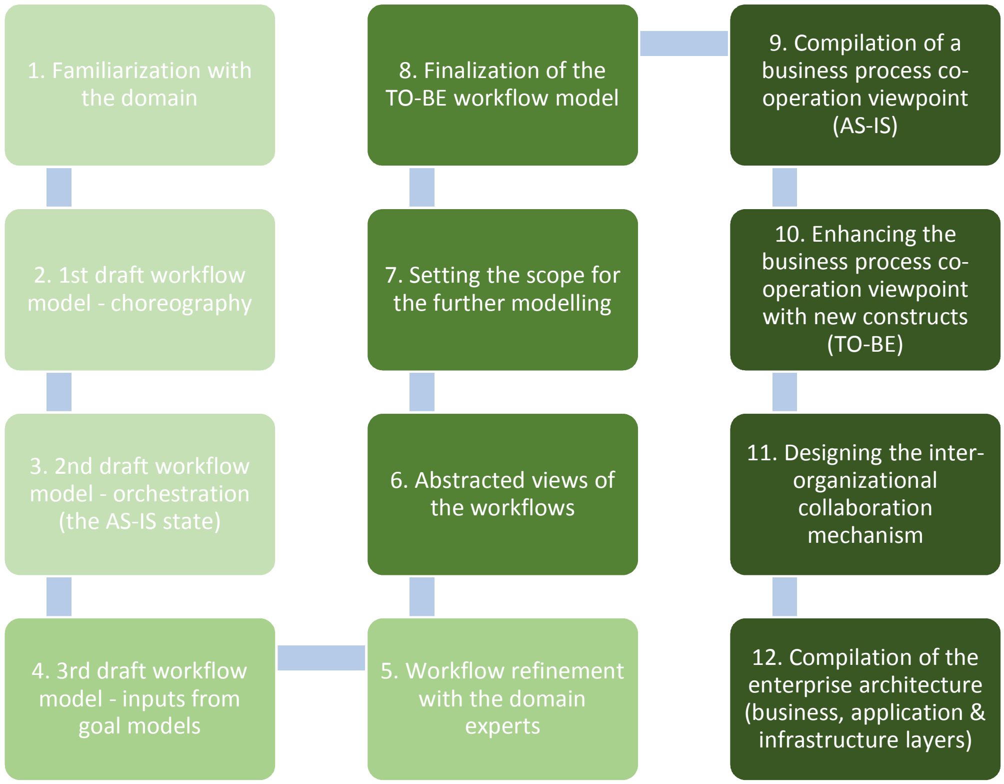


Figure 59. Generalized steps of the airline transformation. (Rein Nõukas)

Figure 59 depicts the twelve steps that are taken in order to transform the airline's operational setup. Although the steps are presented as a sequence, the author admits that certain step(s) can be run in parallel in case of sufficient analytical resources and knowledge-sharing capabilities. The transformation enables the airline to become digitally mature and be compatible with the prerequisites of joining the virtual enterprises. These steps are grouped into four phases labelled arbitrarily to reflect each group's main characteristics. The phases and the steps belonging to each group are described as follows:

A. Preparation phase:

- Main focus: learning the problem domain via the normalization of the AS-IS work flows:
    - 1. Getting familiarized with models, documents, process descriptions covering the domain and the operations structure of an enterprise.
    - 2. Mapping the detected workflows using the proper abstraction level provided by choreography, emergence of the 1st workflow model. Creating a more aggregated and comprehensible AS-IS workflow model by using subprocesses, sets of tasks sharing the same goal.

- 3. Transferring sets of interdependent clusters of subprocesses to a less abstract format using orchestrations, emergence of the 2nd workflow model. Freezing the AS-IS state of the enterprise's business processes.

B. Refinement phase:

- Main focus: amending the existing workflows with industry's best practises:
  - 4. Amending the AS-IS workflow model with the input from model that are using non-BPMN languages in order to fill gaps and missed details in the BPMN-based workflows.
  - 5. Refining the workflow model using domain experts and industry sources in order to get the most up-to-date depiction of the actual business processes and best practises. Laying foundation for the TO-BE workflow model.

C. Focusing phase:

- Main focus: setting the scope for the amended workflows:
  - 6. Defining the critical path using a value chain model. Special attention has to be put on transforming detected data flows, in the form of data-related dependencies, into proper sequence flows.
  - 7. Using domain experts to remove irrelevant workflow parts in order to devise a concrete scope for the further modelling efforts.
  - 8. Embedding iteratively all workflow optimizations to the TO-BE workflow model. Adding a necessary level of detail to transactions inside the model. Freezing the proposed final TO-BE state of the enterprise's business processes.

D. Reconstruction phase:

- Main focus: implementing collaboration-enabling constructs at the enterprise architecture level:
  - 8. Compilation of a business process co-operation viewpoint by removing the temporal parameters from the business processes. Setting the focus on describing the abstract business services and their relationships, in the form of dependencies, with other business process components.

- 9. Reconstructing the business process co-operation viewpoint by adding collaboration-enabling constructs, specifically the separation of public and private views, to the business processes.
- 10. Expanding the impact of the new constructs throughout the business layer of an enterprise by reconstructing the collaboration-enabling data- and workflows, specifically the bidding and coordination mechanisms. The enterprise is opened up for collaboration with external partners at the business process level using service mediators. Emergence of the robust virtual enterprise model from the service consumer's perspective.
- 12. Completing the transformation process by deriving data- and workflow design patterns from the reconfigured business layers and implementing the required changes in the underlying application and infrastructure layers of the enterprise. The finalization of the virtual enterprise model from the service consumer's perspective.

Due to the continuous learning throughout the transformation process the development of the workflow models takes place in an iterative cycles. This means that if something important and valid in terms of workflow configurations is discovered or detected it must be embedded in the currently unfrozen version of the workflow model. The updates are added only to the latest version in order to avoid lengthy reverse engineering of workflow constructs.

During the iterative cycles it is worthwhile to use diagrams depicting business processes and architectural constructs at varying levels of abstraction in order to detect, for example, interesting patterns and inefficient configurations. While atomic level BPMN modelling is useful for configuring concrete transactions it lacks the ability to show the general design patterns and system behaviour. Therefore, business process modelling is amended with enterprise architecture level modelling that enables to get a full vertical and multi-perspective overview of organization's operational sphere and virtual infrastructure. Also, it is encouraged to utilize *ad hoc* syntax and notation that is not compatible to the existing BPMN frameworks in order to enable presenting and understanding the semantics of certain construction flaws that cross the disciplinary boundaries.

5.2 Emergence of a new business model & B2B ecosystem

In this chapter the author discusses the commercial application of the proposed TO-BE model. The model enables airlines to set up partnerships with external service providers in a novel way, thus affecting the business model of the aviation software vendors. An interesting discovery, originating from the TO-BE model that is located in Figure 51 above, lies in the emergence of new business opportunities at varying levels of virtual and physical infrastructure.

The airlines can set up their own private and hybrid clouds that cover both infrastructure and applications in order to publish the public views of their outsource-able business processes. These public views are in general the service acquisition intents of the service consumers. This way the airlines can also host the bidding and coordination enabling environment. According to Figure 51, this cloud environment is also accessible for the service providers. However, it might not be the most logical execution of the service mediation environments. The economies of scale, enjoyed by large cloud hosting companies indicate that airlines do not have any financially-backed reason to set up their own Virtual Private Servers (VPS), *resp.* cloud-based networks. Dedicated hosting companies would take up the role of hosting the service mediation environment by offering IaaS and PaaS.

On top of it, the software vendors in the aviation domain are forced to review their business model and product strategy. Instead of producing silo-based tools to be run by digitally immature airlines, a declining segment, the future profits lie in creating a B2B ecosystem for both airlines and their service providers via providing BPaaS.

To sum it up, there is a new market for cloud hosting companies to cover the provision of cloud- and platform-related services. Also aviation software vendors, as domain experts, have the opportunity to evolve from providing the static bundles of tools into operating business processes as a service (BPaaS). This way the BPaaS operators cater the needs of airlines that can focus solely on operating flights, instead of dealing with internal system integration and process maintenance challenges.

BPaaS (Lynn, et al., 2014) is designed to be service-oriented and supported through defined interfaces. It has a high degree of standardisation to offer the same optimised service delivery system to many different (competing) organisations. The characteristics of BPaaS are summarized in the SEIFS framework (*ibid.*):

- Situation: The service is positioned on the top layer of the cloud stack (i.e. above SaaS, PaaS, and IaaS).

- Elemental: The service supports the configuration or redesigning of existing business processes.
- Integration: The service is seamlessly integrated into existing services (e.g. through APIs).
- Flexibility: The service supports multiple development environments and programming languages.
- Scalability: The service is scalable supporting different customer demands and service optimisation over time.

Lynn et al. (*ibid.*) continue by providing a user and provider perspective towards BPaaS:

- User: a model in which provides standardised business processes on a pay-as-you-go basis. This allows access to shared resources (people, application, and infrastructure) from a single service provider.
- Provider: the delivery of business process outsourcing (BPO) services that are sourced from the cloud and constructed for multi-tenancy. The pricing mechanisms are typically consumption-based or subscription-based.

The establishment of BPaaS covering the vertically integrated airline is beneficial for all parties involved. Airlines, as end-users of the service mediation platform, benefit from faster product and process deployment times, also the capital costs are reduced via the subscription-based licencing. Additionally, airlines can increase their operational efficiency by deploying the most optimal bids, in the form of service offerings (via the provider's public views), from their service providers.

As external service providers get automated and direct access to airlines' public views, their operational costs decrease and more efforts can be placed in raising the competiveness of the service offering, *f. ex.* tracking airline's needs for services in real-time, focusing on specific segments and becoming operational efficiency champions. The emergence of BPaaS-based collaboration environment results in the lowered technical and operational threshold for the external service providers as there are set in place standardized transaction mechanisms to exchange data and coordinate business processes with their consumers. The reduction of costs in service 'purchasing' and conducting transaction are also be reflected in the fees charged from the passengers and cargo customers in case of a functioning competition-based market environment.

The reasoning given above applies to a situation in which the airlines are cost-driven and – conscious. For example, according to IATA (2010) labour-related expenses (see Table 5 below) comprise up to one-fourth of the airline's cost-base in Europe, whereas in Asia Pacific region the figure is as low as 14,7%. Therefore, it is relevant to expect a financially self-dependent (*resp.* non-(governmentally) subsidised) airline that has its main bases in high-living cost countries to keep optimal the number of employees on its payroll.

	North America		Europe		Asia Pacific		All Major Airlines	
	2001	2008	2001	2008	2001	2008	2001	2008
Labour	36.2%	21.5%	27.2%	24.8%	17.2%	14.7%	28.3%	20.1%
Fuel	13.4%	34.2%	12.2%	25.3%	15.7%	36.7%	13.6%	32.3%
Aircraft Rentals	5.5%	3.0%	2.9%	2.5%	6.3%	4.5%	5.0%	3.5%
Depreciation and Amortisation	6.0%	4.5%	7.1%	5.7%	7.4%	7.8%	6.7%	5.9%
Other	38.9%	36.9%	50.7%	41.8%	53.4%	36.3%	46.4%	38.2%

Table 5. Percentage share of airline operating costs, by region of airline registration. (IATA, 2010)

The non-core functions of an airline that are not covered by airline-employed staff are outsourced to external partners inhabiting each airport. These functions populate mostly the turnaround phase and maintenance related services, and, to a smaller extent, the information processing services, like weather forecasting.

The outsourcing is also justified by the fact that the turnaround is very labour-intensive compared to the flight preparation and take-off phases. This means that an airline's operating costs can fluctuate a lot depending on the peak hours and traffic volume. Assuming that usually airlines try to maximise the number of resources during peak hours (see example of peak hours at airport in Figure 60) and keep these minimal during early mornings and late evenings outsourcing is the only financially reasonable and sound method of running operations. The airline is charged only for the actually consumed services (*resp.* pay-as-you-go principle) provided by external service providers.

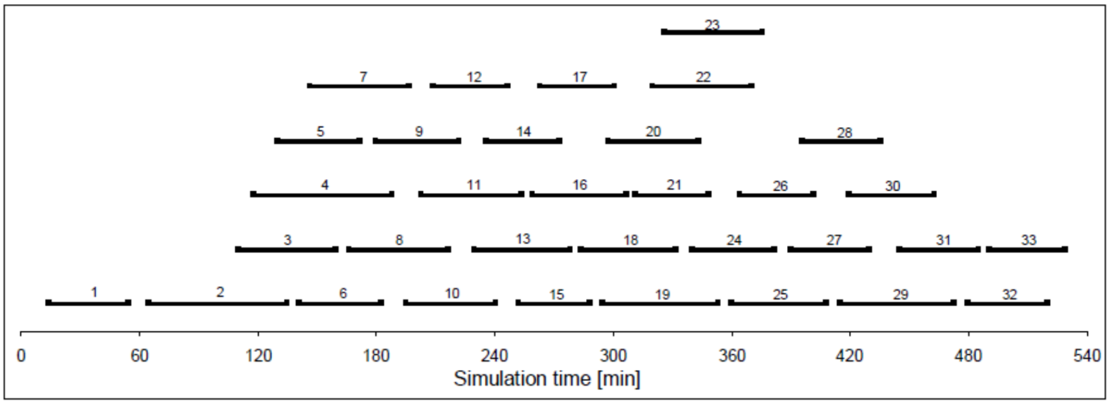


Figure 60. Generated traffic example at an airport. (Vidosavljević & Tošić, 2010)

Figure 60 depicts a day of operations in an airport from the traffic perspective. In the morning there is only one aircraft needing terminal-servicing, *resp.* turnaround. As the day passes, the volume of the aircraft arriving for terminal-servicing increases, requiring the airport also to increase the capacity of ground operations and passenger management to handle the incoming work load. During evening hours the number of aircraft arriving in parallel to the airport decreases again, and the airport reduces its terminal-servicing capacity consequently.

In order to provide further justification for service outsourcing and, in turn, to prove the relevance of the need to set up a cloud-based service mediation environment, the author reviews fiscal figures related to airline's cost-base during the turnaround phase.

Wu and Caves (2000) discuss in their publication, see Table 6 below, the aircraft operational costs incurred with the aircraft during their non-flying time.

	British Airways	British Midland	KLM	Lufthansa	American Airlines	United Airlines
Revenues[b]	12,226	890	5699	9986	15,856	17335
Variable costs[b]						
Fuel and oil	(1149)	(50)	(580)	(1014)	(1726)	(1898)
Maintenance	(663)	(64)	(350)	(441)	(937)	(1049)
Station expenses	(1602)	(93)	(875)	(1434)	(2102)	(2195)
Passenger service expenses	(1637)	(139)	(535)	(1168)	(1775)	(1895)
Subtotal[c]						
(Revenues − Costs)	7172	576	3359	5929	9316	10,298
Flight hours (h)	*840,223*	*118,392*	*433,339*	*988,393*	*2,039,569*	*1,865,195*
Schedule time costs ($/h)	*8535*	*4865*	*7751*	*5998*	*4567*	*5521*

[a]Sources: Digest of Statistics, Financial Data Commercial Air Carriers, ICAO 1997. Digest of Statistics, Fleet-Personnel, ICAO 1997.
[b]Units in US $ (millions), () Values of cost items.
[c]Subtotal = (Revenues) − (Costs).

Table 6. Hourly schedule time costs of major airlines. (Wu & Caves, 2000)

Table 6 gives solid indications of the airline's cost base when an aircraft is at the airport for servicing and turnaround – a set of processes conducted usually by the external service providers. *Station expenses* and *Passenger service expenses* comprise more than two-thirds of the airline's variable costs. The financial considerations force airlines to seek advanced service outsourcing solutions in order to bring down the costs via lowered competition barriers for the service providers.

Chapter 5 hosted a discussion regarding the outcomes of Chapters 3 and 4 in the context of abstract transformation methods and a novel B2B business model. Chapter 6 summarizes the findings of this thesis and provides an outlook for the further research.

6. Conclusions

The final chapter of this thesis summarizes the research efforts. At first, in Chapter 6.1, the general conclusions are drawn. Next, in Chapter 6.2, the answers to the thesis' research questions are provided. Finally, in chapter 6.3, prospective ideas for the future research are discussed in order to expand the findings and results of this thesis in the wider academic and commercial context.

6.1 General conclusions

In this thesis a collaboration-enabling environment is constructed using the aviation industry as an example. This environment accommodates and facilitates the cooperation of the service consumers, *resp.* airline companies, and external service partners at the business process level. The focus, therefore, is set on the transformation of the airline's organization to become digitalized and developing a supporting platform for service mediation between digitalized organizations of the collaboration environment.

The outcome of the thesis is the emergence of virtual enterprise model, situated in the aviation domain that comprises native components, controlled directly by an airline, and remote components that rely on the mutual dependency instead of control. This multi-party model displays the configurations that enable the intra- and inter-organizational coordination of business processes at software level.

The research in this thesis is backed up with the theoretical approaches covering the concepts of outsourcing, collaborations and Virtual Enterprises. These topics combined with business process modelling and cloud computing provide a basis for innovative business models for redefining the understanding of the B2B transaction configuration models.

In detail, the following tasks are covered in this thesis:

Identification of the airline's business processes:

> The initial workflow model describing the airline's workflows is transformed into a normalized state of the business processes by using the BPMN 2.0 framework. At first choreography is used to map the process clusters. Consequently, the model is amended with the full BPMN notation by using orchestration.

Refinement of the airline’s business processes:

The AS-IS state of the airline’s business processes is amended with industry’s best practises and applicable goal models in order to devise the TO-BE vision. The latter is used to create an abstract value chain model that depicts clearly the sequential relationships between the activities and data objects, in general the atomic workflow components. The value chain model, in turn, is used to devise the concrete scope for further modelling. The outcome of the refinement process is a finalized TO-BE state of the airline’s business processes.

Reorganization of the airline’s business processes:

Prior to the reorganization of the airline’s business processes, the missing functionality is detected that is originating from the constraints of the enterprise architecture. This gap denies an airline’s service partners to interact with the airline at the business process level. In order to overcome this situation, the public and private process views are introduced. By having a novel process configuration, the airline is able to disclose information to its external partners. This enables to reconstruct the airline’s business layer in order to support the emerged B2B collaboration mechanism. The B2B collaboration mechanism enables the intra- and inter-organizational coordination of business processes at software level, enabling the airline to automate the service outsourcing acquisition and coordination instances. Additionally, in conjunction with the airline’s business layer, the other parties of the B2B collaboration mechanism - the external service partner and service mediator - are presented in the form of business layers. These business layers cover only the functionality that is required to run the collaboration instances.

Reconfiguration of the airline’s digital architecture:

The abstract data flow models for bidding and coordination phases of the collaboration are configured. This is done by using the reconstructed business layers of the airline, external service partner and service mediator. The data flow models are based on the loose coupling principle, enabling to adopt the service-oriented architecture for the entire B2B collaboration mechanism. This is followed by the reconfiguration of the airline’s digital architecture. It consists of the application and infrastructure layers that support the functioning of the B2B collaboration mechanism.

As a consequence, it is possible to provide transformation heuristics for reasoning about the complexities of the intra- and inter-organizational transformation in the context of the digital maturation for B2B collaboration. The transformation heuristics provide an aggregated and abstract knowledge about the evolution of the business processes from human-operated workflows inside one organization to automated business transactions between parties of a business ecosystem.

This thesis is about the application of the concept of BPaaS in the context of the aviation industry as a means of proof-of-feasibility. As a consequence, the author proposes the implementation of the B2B collaboration mechanism to be located in the cloud by using the concept of BPaaS. This way, the collaborating partners inside the virtual enterprise – a cluster of autonomous entities with a common goal - can set up loose coupling between their respective business processes enabling automated and autonomous collaboration instances. The aim of loose coupling (Josuttis, 2007) is to minimize dependencies between system components. This concept is deployed to deal with the requirements of scalability, flexibility and fault tolerance that surface in multi-party configurations.

Commercially, the transition to a cloud-based operations in the context of BPaaS, opens up the possibility to engage into one-to-X relationships. In this kind of relationships the outsourcing process is mediated by a technological solution, *e.g.* a service mediator that establishes terms and conditions per each outsourcing, *resp.* exchange of resources, instance. Furthermore, by dissolving the silo-based architecture of airlines, in the context of joining the separated clusters of the business processes into integrated workflows, the airline can become aware of all of its process states and instances in real time.

6.2 Answers to the research questions

The research questions that are defined for this thesis have been answered as follows:

RQ 1: How to normalize and map airline's business processes?

The identification and refinement of the business processes are executed by analysing and re-modelling the processes populating the day of operations of an airline. By modelling airline's business processes using both BPMN 2.0 compliant choreography and orchestration it is possible to depict the direct and indirect relationships and dependencies between participating cross-organizational entities. Also control flows are easily visualized while maintaining readability (*resp.* applicability) of a diagram. Using the aforementioned methods is especially relevant in domains that are populated by a large number of peer entities resulting in a high complexity.

- **rq 1.1: What are the re-built airline's business processes?**
  The airline's business processes during the day of operation span across 4 operational phases: *Short term planning*, *DOO shift_1, DOO flight preparation, DOO flight calculation* and *DOO actual flight*. In total there are detected 59 various activities that are mostly clustered into subprocesses. These activities are enacted by 11 human-operated agents and 10 software operated agents. The activities and agents are used in order to construct a normalized BPMN-based workflow model.

- **rq 1.2: What kind of methods can be used to refine the airline's business process model?**
  The normalized BPMN-based workflow model is refined by utilizing of the input from the domain experts, merging the goal models with the workflow model and construction of an abstract value chain model. The refinement process provided a level of detail to the airline workflows, that is, in turn, vital for organizing the airline's business processes in real-life situations. The refinement process resulted in the re-arrangement of the activities into three subdomains of the day of operations: *Flight preparation, Turnaround* and *Take-off*. All activities that are not directly contributing to the completion of the goals of the aforementioned subdomains, are discarded from the workflow model. Also, the analysis shows that the implementation of the disruption management processes inside the main workflow model is impractical as the disruptions are unpredictable by nature – therefore a separate use cases must be set up prior to modelling the respective workflows.

RQ 2: How to increase an airline's digital maturity level?

The transformation of an airline with low digital maturity is finalized by using the collaboration and cloud computing perspectives. The airline's silo-based architecture is dissolved by relocating the data processing and sharing processes to the cloud computing environment. This enables the publishing of non-confidential information in the form of public views of business process to external service providers thus, making an airline ready for facilitating collaborations in an automated manner. The cloud-based collaboration instances between airline and external service partner are mediated by a separate entity – service broker. In parallel with horizontal inter-organizational integration, the airline becomes a vertically integrated organization that is aware of its interdependent and sequential data- and workflows as all its business processes are located in the same environment.

- **rq 2.1: What are the re-organized airline's business processes?**
  By using the business process co-operation viewpoint it is possible to create an abstract business process model that enables the introduction of the private and public views of the business processes. This, in turn, allows to compose the collaboration-enabling process view encompassing the airline, mediator and service provider. By having both the bidding and the coordination part of the collaboration mechanism defined and delivered, the TO-BE state of the airline's business processes is achieved – the configuration of the business processes is aligned with the airline's digital maturation.

- **rq 2.2: What is the re-configured airline's digital infrastructure?**
  The airline's digital architecture, in the form of the business, application and infrastructure layers, is based on the TO-BE state of the airline's business processes. An important role is played by the inclusion of the cloud computing related concepts in the reconfiguration of the architectural constructs. In parallel with provisioning of the collaboration-enabling digital architecture of an airline, it is necessary to define also the corresponding functionality of the other key entities of the collaboration environment – mediator and service provider. This enables to detect the technological challenges for facilitating multi-party transactions and, based of these, present a full set of architectural solutions.

The aforementioned research questions and their respective answers contribute towards achieving the main research objective:

> *How to configure a cloud-based collaboration-enabling ecosystem for the aviation industry from an airline perspective?*

At first, the business processes of airline have to be understood thoroughly and placed into the context of validated workflow models. The airline has to achieve a vertical intra-organizational integration that allows it to coordinate and monitor all of its processes in real time and at runtime. This state is followed by readying the airline for the inclusion to the collaboration-enabling ecosystem by placing its data processing and sharing components of the information systems in the cloud computing environment. The ecosystem is based on the service-oriented architecture enabling one-to-many relationships between its residing autonomous entities that are presented in the form of services. The ecosystem is the representation of the horizontal integration of its entities. The flexibility, scalability and fault tolerance of the ecosystem is preserved by adopting the concepts of loose coupling and publish-subscribe patterns for facilitating the multi-party transactions. In this ecosystem the collaboration instances are based on the universal patterns that enable both the initiation and coordination phases of the collaboration instance.

6.3 Future work

Throughout the course of the research tasks several prospective topics are identified that can be proposed for the future research. These ideas are discussed briefly in order to expand the findings and results of this thesis in the wider academic and commercial context.

Validation of the proposed transformation heuristics, either as a whole process or per phase/step.

- Amending and extending the proposed transformation heuristics with constructs that enable to include quantifiable criteria, like process completion time and costs, and qualitative criteria, like process flexibility and level of predictable behaviour.
- Application of the proposed transformation heuristics in some non-aviation related data-heavy B2B business domain in which the virtual enterprises are becoming a potentially valid evolutionary goal.

Expanding the proposed BPaaS-based business model and B2B ecosystem by adding more in-depth analysis, both from technical, social and business perspectives.

- Governance of the outsourcing relationships

  There is a need to elaborate on the complex issues raising from the governance of the outsourcing relationships in the rapidly changing conditions as are prescribed by the author in the collaboration-enabling model. As a strategic resource, outsourcing must be governed accordingly (Felton, Hudnut, & Heeckeren, 1996). The outsourcing is not just conducting certain transactions but it is about effective management of collaborations. This way it can be ensured that outsourcing really becomes an economically viable solution for all parties involved - an airline, BPaaS operator and external service provider. According to IT Governance Institute (Simmonds & Gilmour, 2005) the scope of the governance and its execution comprise a wide-ranging set of responsibilities, roles, objectives, interfaces and controls required to anticipate change and manage the introduction, maintenance, performance, costs and control of third-party provided services.

Part of the governance of outsourcing is the set-up of SLAs to regulate the mutual expectations of service provider and customer. The European Network of Excellence in Software Services and Systems (Liegener, 2012) describes Service Level Agreement as a formal written agreement made between two parties: the service provider and the service recipient. The SLA itself defines the basis of understanding between the two parties for delivery of the service itself.

The airline has to be able to measure neutrally the metrics set in the SLA. This ability, consequently, also contributes towards predicting the trust-worthiness of the BPaaS provider as the airline has to know the concurrent risk levels related to using a BPaaS solution. As discussed by Müller et al. (2012), it is valid to include a third-party monitoring organization or an SLA management tool to supplement the vendor's data, especially in case of mission-critical services where the business itself is at risk if service levels are not met. The extra expense of these additional methods can be worthwhile for critical services, e.g. running airline operations.

Focusing on the implementation of the proposed collaboration-enabling model in the cloud computing context. This is done by expanding the BPaaS based solution described in Chapter 4.2.2 *The architecture model* and presented in Chapter 5.2 *Emergence of a new business model & B2B ecosystem*.

- Solution deployment

The solution deployment focuses on the configuration of the infrastructure layer. The functionality of that layer has to be specified in accordance of the requirements of the collaboration-enabling platform. Data storing in cloud and data transmissions between counterparts, including consumers, virtual marketplace and service providers, are the cornerstones of the proposed BPaaS based environment. All parties expect their data to be stored securely and messages to be delivered accurately and with no delays. Without being able to maintain these expectations, in the form of building trust, this kind of business process outsourcing solution has no market in a real-life situation.

In order to deliver a highly autonomous and secure cloud environment the Agent-Oriented Modelling[44] concept can be explored. It enables to utilize a multi-agent system (Giorgini & Henderson-Sellers, 2005), a system composed of cooperative or competitive agents that interact with one another in order to achieve individual or common goals. The key characteristics of agents are widely understood to be highly autonomous, proactive, situated, and directed software entities (*ibid.*). Agents have to be able to enact various roles during runtime. One agent can have several roles in parallel that are taken up and enacted depending on the circumstances, *resp.* environment, and rules of engagement set by a cloud BPaaS operator and/or end-users. The beauty lies in the system's open architecture – an agent can leave or join the system, *resp.* cloud cluster, dynamically making the system highly scalable and flexible to respond to external impacts.

The collaboration-enabling cloud environment can be defined as a multi-agent system if it is acting on a peer-to-peer basis. Some cloud nodes consist of only one agent that is controlling all of its servers and virtual servers, or some nodes have many agents, having each agent to represent one (virtual) server in a block of servers. This depends on the architecture-related preferences and risk management policies. Having no central locus of control means the cloud BPaaS operator has reduced significantly the risk of losing control over the informational resources in its possession in case of a physical or cyber-attack. Other benefits include removing the human factor almost completely from the process as the multi-agent system is self-reliant by being proactive while no manual interference is needed. Moreover, the agents exchange constantly information about data they possess, making it easy and un-delayed to query necessary bits of it at runtime.

- Cloud elasticity

The operational and financial performance of the proposed BPaaS solution relies heavily on the configuration of the cloud elasticity patterns. Elasticity (Herbst, Kounev, & Reussner, 2012) is the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as

[44] The concept is actively researched by Professor Kuldar Taveter in Tallinn University of Technology – R. N.

possible. As the airlines' models of operation differ regarding the staff composition, outsourced processes and the fleet there is not reasonable to propose a clear set of configuration for setting up a cloud environment that hosts all airline's operations. Also, the level of risk tolerance, manifested in setting up several reserve and back-up system, determines the airline's preferences in the aforementioned topic.

A set of scenarios can be used to analyse and simulate the behaviour of the collaboration-enabling cloud solution in the context of cloud elasticity. These scenarios are based on the case studies taken from the aviation domain. The analysis is needed in order to understand the impact of various sizes and formats of data sets towards the data processing capacity of the proposed BPaaS solution.

Elasticity of the architectural constructs given in this thesis can be assessed in the context of Vienna Elastic Computing Model. This model (Dustdar & Truong, 2012) aims at introducing techniques and frameworks to support multi-dimensional elastic processes atop multiple cloud systems of software-based and human-based services. This enables, in turn, to investigate further the possibilities to virtualize the human-operated and software-operated computing elements in the dynamic business processes of airlines and its service ecosystem.

References

Apache Software Foundation. (2014, November 5). *Kafka 0.8.1 Documentation*. Retrieved from Apache Kafka: http://kafka.apache.org/documentation.html#introduction

Aalst, W. v. (1998). Formalization and Verification of Event-driven Process Chains. *Computing Science Reports*, 29. Retrieved from http://alexandria.tue.nl/extra1/wskrap/publichtml/9714860.pdf

Aerohive Networks Inc. (2013). *Public or Private Cloud: The Choice is Yours.* Aerohive Networks Inc. Retrieved February 8, 2015, from http://www.aerohive.com/pdfs/Aerohive-Whitepaper-Public-or-Private-Cloud.pdf

Balashova, J. (2014). Tallinn.

Beauvoir, P. (2014, September 29). *ArchiMate*. Retrieved from Archi: http://www.archimatetool.com/about

Boeing. (2014, August 14). *Digital Airline*. Retrieved from Boeing: http://www.boeing.com/boeing/commercial/aviationservices/integrated-services/digital-airline.page

Brambilla, M. (2011). Choreography vs. Orchestration. Retrieved May 20, 2014, from http://www.bpminstitute.org/community/bpm-group/choreography-vs-orchestration

Breiter, G., & Naik, V. (2013). A Framework for Controlling and Managing Hybrid Cloud Service Integration. (pp. 217-244). IEEE Computer Society. Retrieved February 8, 2015, from http://www.academia.edu/5321173/A_Framework_for_Controlling_and_Managing_Hybrid_Cloud_Service_Integration

Buterin, V. (2014). *A next-generation smart contract and decentralized application platform.* Retrieved March 23, 2015, from http://buyxpr.com/build/pdfs/EthereumWhitePaper.pdf

Camarinha-Matos, L., & Afsarmanesh, H. (2001). Virtual Enterprise Modeling and Support Infrastructures: Applying Multi-Agent System approaches. *Lecture Notes in Artificial Intelligence*, 335-364.

Campagnolo, D., & Costa, G. (2006). Modularità e Business Processing Outsourcing. *Sviluppo e Organizzazione*, 97-114.

Chandler, A. J. (1977). *The Visible Hand: The Managerial Revolution in American Business.* Cambridge: MIT.

Ciborra, C. (1993). *Teams, Markets and Systems.* Cambridge: Cambridge University Press.

Coase, R. H. (1937). The Nature of the Firm,. *Economica*, 386-405.

Duipmans, E. (2012). *Business Process Management in the Cloud with Data and Activity Distribution (Master Thesis).* Twente: University of Twente. Retrieved from http://essay.utwente.nl/62490/1/MSc_E_Duipmans.pdf

Dustdar, S., & Truong, H.-L. (2012). Virtualizing Software and Humans for Elastic Processes in Multiple Clouds– a Service Management Perspective. *International Journal of Next-Generation Computing, 3*(2), 109-126. Retrieved March 29, 2015, from http://www.infosys.tuwien.ac.at/Staff/sd/papers/Zeitschriftenartikel%20H.%20L.%20Truong%20Virtualizing%20Software%202012.pdf

Felton, R. F., Hudnut, A., & Heeckeren, J. v. (1996). Putting a Value on Board Governance. *The McKinsey Quarterly, 4*, 170-175.

Giorgini, P., & Henderson-Sellers, B. (2005). *Agent-Oriented Methodologies.* Idea Group Pub. Retrieved from http://www.troposproject.org/files/Henderson01.pdf

Grandori, A. (1984). *Teorie organizzative.* Giuffrè: Milan.

He, H. (2014, December 14). What Is Service-Oriented Architecture. *Publicação eletrônica em*. Retrieved from http://uic.edu.hk/~spjeong/ete/xml_what_is_service_oriented_architecture_sep2003.pdf

Herbst, N. R., Kounev, S., & Reussner, R. (2012). Elasticity in Cloud Computing: What It Is, and What It Is Not. *Proceedings of the 10th International Conference on Autonomic Computing*, (pp. 24–28). San Jose. Retrieved March 29, 2015, from https://sdqweb.ipd.kit.edu/publications/pdfs/HeKoRe2013-ICAC-Elasticity.pdf

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly, 28*(1), 75-105. Retrieved from http://www.brian-fitzgerald.com/wp-content/uploads/2014/05/Hevner-et-al-2004-misq-des-sci.pdf

IATA. (2010). *IATA Economic Briefing.* IATA. Retrieved from http://www.iata.org/whatwedo/Documents/economics/Airline_Labour_Cost_Share_Feb2010.pdf

IBM. (2011). *IBM Cloud Computing Reference Architecture 2.0.* Document. Retrieved March 29, 2015, from http://www.ibm.com/developerworks/websphere/library/techarticles/1203_lau/1203_lau.html

Josuttis, N. M. (2007). *SOA in Practice: The Art of Distributed System Design.* Sebastopol: O'Reilly Media Inc.

Liegener, B. (2012, April 26). *Service Level Agreement (SLA)*. Retrieved from S-Cube network: http://www.s-cube-network.eu/km/terms/s/service-level-agreement

Lynn, T., Mooney, J., Helfert, M., Corcoran, D., Hunt, G., Van Der Werff, L., . . . Healy, P. (2014). Towards a Framework for Defining and Categorising Business Process-As-A-Service. *21st International Product Development Management Conference*, (pp. 1-22). Retrieved March 29, 2015, from http://www.researchgate.net/profile/Theodore_Lynn/publication/263505922_Towards_a_Framework_for_Defining_and_Categorising_Business_Process-As-A-Service_(BPaaS)/links/0a85e53b1d62408b3e000000.pdf

Mell, P., & Grance, T. (2011). *NIST Definition of Cloud Computing.* Gaithersburg: The National Institute of Standards and Technology. Retrieved from http://www.nist.gov/itl/cloud/

Merriam-Webster. (2015, January 08). *Heuristic*. Retrieved from Merriam-Webster: http://www.merriam-webster.com/dictionary/heuristic

Miers, D. (2006). A Process “A Ha” Moment – Two sides of the same coin – Orchestration and Choreography (and BPDM). Retrieved MAY 20, 2014, from http://bpmfocus.wordpress.com/category/orchestration/

Moore, J. (2006). Business ecosystems and the view from the firm. *Antitrust Bulletin, 51*(1), 31-75. Retrieved September 24, 2014, from http://aai.beekeeperdev.com/files/Business%20ecosystems%20and%20the%20view%20from%20the%20firm,%20antitrust%20bu_081320081450.pdf

Müller, C. O.–C. (2012). SALMonADA: A Platform for Monitoring and Explaining Violations of WS–Agreement–Compliant Documents. *Proceedings of the 4th International Workshop on Principles of Engineering Service-Oriented Systems* (pp. 43-49). IEEE Press. Retrieved March 28, 2015, from http://upcommons.upc.edu/e-prints/bitstream/2117/17809/1/SALMONADA.pdf

Norta, A. (2012). *Safeguarding Trusted eBusiness Transactions of Lifecycles for Cross-Enterprise Collaboration.* Helsinki: University of Helsinki.

Norta, A. (2015). Transactional Process-Lifecycles for Decentralized Autonomous Organizations in Open Collaboration Ecosystems. Tallinn.

Norta, A., & Eshuis, R. (2010). A Framework for Service Outsourcing using Process Views. *Enterprise Distributed Object Computing Conference (EDOC), 2010 14th IEEE International* (pp. 99 - 108). Vitoria: IEEE. Retrieved from https://helda.helsinki.fi/bitstream/handle/10138/24230/norta10framework.pdf?sequence=2

Norta, A., Grefen, P., & Narendra, N. C. (2014). A reference architecture for managing dynamic inter-organizational business processes. *Data & Knowledge Engineering*(91), 52-89.

OMG. (2011). *Business Process Model and Notation 2.0.* Object Management Group, Inc. Retrieved August 24, 2014, from http://www.omg.org/spec/BPMN/2.0/PDF/

OMG. (2011, March 01). *Documents Associated With Business Process Model And Notation v2.0.* Retrieved ARPIL 22, 20014, from Object Management Group: http://www.omg.org/spec/BPMN/2.0/PDF/

Paul, R. (2009, March 23). *Topic-based publish/subscribe design pattern implementation in c#-Part II (Using WCF).* Retrieved from Codeproject: http://www.codeproject.com/Articles/34333/Topic-based-publish-subscribe-design-pattern-imp

Peffers, K., Tuunanen, T., Rothenberg, M. A., & Chatterjee, S. (2008). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems, 28*(3), 45-77.

Rossignoli, C. M. (2007). Virtual network organizations and e-procurement: a new outsourcing perspective. *First Information Systems Workshop on Global Sourcing: Services, Knowledge and Innovation* (pp. 1-28). Val d'Isere: University of Verona.

Rouse, M. (2011, January 01). *business process management (BPM)*. Retrieved from Techtarget.com: http://searchcio.techtarget.com/definition/business-process-management

Signavio GmbH. (2014, August 14). *Signavio*. Retrieved from http://academic.signavio.com/

Simmonds, A., & Gilmour, D. (2005). Governance of Outsourcing. *IT Governance Domain Practices and Competencies*, 23. Retrieved from http://www.isaca.org/Knowledge-Center/Research/Documents/Governance-of-Outsourcing_res_Eng_0105.pdf

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications, 1*, 7-18. Retrieved from http://netcomm.bjtu.edu.cn/wp-content/uploads/2014/03/Cloud-computing-state-of-the-art-and-research-challenges.pdf

TechTarget. (2014, December 14). *Web 2.0*. Retrieved from WhatIs.com: http://whatis.techtarget.com/definition/Web-20-or-Web-2

Ulrich, D., & Barney, J. B. (1984). Perspectives in organizations: resource dependence, efficiency, and population. *Academy of Management Review,*, 471-481.

Ward-Dutton, N. (2011). Choreography vs. Orchestration. Retrieved MAY 20, 2014, from http://www.bpminstitute.org/community/bpm-group/choreography-vs-orchestration

Weske, M. (2007). *Business Process Management - Concepts, Languages, Architectures.* Berlin: Springer. Retrieved from http://www.randytopp.nl/RT/0Portal/book/manag/book/38_BPM.pdf

White, S. A. (2009). Introduction to BPMN. Retrieved May 20, 2014, from http://www.omg.org/bpmn/Documents/Introduction_to_BPMN.pdf

White, S., & Miers, D. (2008). *BPMN Modeling and Reference Guide: Understanding and Using BPMN.* Future Strategies Inc.

Williamson, O. E. (1975). *Markets and Hierarchies. Analysis and Antitrust implication.* New York: The Free Press.

Williamson, O. E. (1979). Transaction Cost Economics: «The Governance of Contractual Relation». *Journal of Law and Economics*, 233-261.

Wu, C.-L., & Caves, R. (2000). Aircraft operational costs and turnaround efficiency at airports. *Journal of Air Transport Management, 6*(4), 201-208.