Bridging Business Value Models and Process Models in Aviation Value Webs via Possession Rights

Vincent Pijpers
VUA University Amsterdam
De Boelelaan 1105
1081 HV Amsterdam
The Netherlands
pijpersv@few.vu.nl

Jaap Gordijn
VUA University Amsterdam
De Boelelaan 1105
1081 HV Amsterdam
The Netherlands
gordijn@few.vu.nl

Abstract

While exploring value webs -cooperating enterprises- it is common to view such webs from multiple perspectives: (1) the business value perspective, (2) the business process perspective, and (3) the information system perspective. The value perspective explains why a web can exist from a commercial perspective, whereas the process perspective shows the interacting processes of enterprises and the IT perspectives shows the supporting IT architecture. These perspectives each take a different view on the same phenomenon: the value web. Because the phenomenon is for each viewpoint the same the perspectives need to be consistent. This paper introduces an approach to arrive at a business process model of a value web that is consistent with a business value model of the same value web. We propose a step-wise approach that starts with considering the independent transfer of ownership right of a value object and the actual object itself, and finally considers time ordering of these transfers. We illustrate our approach using an industrial strength case study in the aviation sector.

1 Introduction

Enterprises increasingly operate as *value webs* rather than just on their own. Value webs [15] are collections of enterprises that jointly satisfy a complex consumer need, each enterprise contributing its own specific expertise, products and services. Well known webs include constellations such as Cisco Systems and Dell, but many other exemplars can be found. For instance in the field of energy, entertainment, and as we will present in this paper, in the aviation industry. Obviously, value webs are enabled by, and heavily rely on, the use of information technology to coordinate process execution for production and service provisioning.

We have argued in previous work [8] that exploration of a value web, initially for composition of the web and business development purposes, and ultimately with the aim to develop supporting information systems, requires at least three *different perspectives* [11]. The *information system* perspective represents the information systems architecture supporting the value web. Emphasis is here on the interoperability between various information systems of enterprises. Standards, *and* their proper use, such as ebXML, BPEL, WS-Coordination, and UDDI come into mind here [2]. Stakeholders are IT-architects. The *business process perspective* shows the *operational* organization of *intra*-enterprise activities (control&data flow, resource allocations, etc) *and* the cross-organizational coordination of these activities. Important players are here business process (re)-engineers. Various alternatives exist to describe business processes including UML activity diagrams [3] and Petri nets [16]. Finally, the *business value perspective* illustrates which companies participate in the value web, as well as *what* of *economic value* is transferred between each other, *and* what is requested *in return*. Stakeholders are business developers and CxO's. The perspective is e.g. used to understand and evaluate economic sustainability for each participating enterprise. Effectively, the value perspective provides the business rationale for the other two perspectives. Various modeling approaches have been proposed for the business value perspective, amongst others e^3*value* [8], the BMO [12], and REA [6].

The separation of stakeholder concerns into various perspectives is a well known approach to structure executive decision making about value webs, and has proven to be of value in various industrial strengths case studies we have done [1, 7–10]. Specifically, *commercial* decisions (business value perspective) are quite different from *operations* decisions (business process perspective), and therefore should not be mixed-up and be taken by the appropriate stakeholders. However, if using a multiple-perspective approach, executive decisions should also be *consistent* over viewpoints, and consequently, the viewpoints themselves should be consistent with each other, as each viewpoint takes a *different* perspective on the *same* phenomenon. This paper addresses how to reach such consistency between two specific perspectives, namely the business value perspective and the business process perspective.

In the current line of research two approaches for maintaining consistency between the mentioned viewpoints are used: (1) *informally*, by giving a set of guidelines how to use e.g. the business value perspective for finding a related business process perspective and vice versa [4], and (2) *formally*, by stating consistency rules between perspectives, which e.g. can be checked by model checkers [18,19]. The proposal to maintain consistency as discussed in this paper is an exemplar of the first approach; its contribution is unique because it shows how to move gradually in a *structured* way from a business value model to a business process model.

This paper is structured as follows. In Sec. 2 we give a concise introduction into e^3*value* modeling. Subsequently, Sec. 3 we present process modeling as another perspective to explore while designing value webs. To understand (1) that value models are not equal to process models, and (2) to understand *what* has to be bridged then precisely, Sec. 4 presents the differences between value models and process models, and introduces the steps needed to be taken to fill the gap between value models and process models. Next a case study in the aviation industry will be described and our bridging will be applied. Finally, we discuss lessons learned, and present conclusions and suggestions for further research.

2 Value Modeling

For representing the value perspective, we use the e^3*value* ontology [8]. E^3*value* models which enterprises (actors) are involved in a value web, *what* they transfer of *economic value* with each other, as well as what they request in return (the so-called economic *reciprocal* transfers). An e^3*value* model does *not* describe *how* value transfers are actually done. It only describes the economic value transfers that occur. As we have experienced in case studies, deciding *which* economic value transfers exist, is already sufficient complex for stakeholders in its own right; that is why we postpone decisions regarding business processes. An e^3*value* model can also be used to calculate net profits per actor over different periods of time. In this paper the e^3*value* ontology will however only be used to understand the various economic transfers of value objects between actors. In the next paragraphs the e^3*value* ontology constructs will be introduced that can be used to represent a value model.

Actor. An actor is perceived by its environment as an independent economic (and often also legal) entity. Economically independent refers to the ability of an actor to be profitable after a reasonable period of time (in the case of an enterprise), or to increase economic utility for him/herself (in the case of an end-consumer). In a sound, viable, value model each actor should be capable of making a profit or to do utility increase. *Value object.* Actors transfer value objects. A value object is any object which has value for an actor. An object has economic value for an actor when the actor can use the object to satisfy a need or when the actor can use the object for transfer with another object. *Value port.* An actor uses a value port to show to its environment that it wants to provide or request value objects. The concept of port enables us to abstract away from the internal business processes, and to focus only on external actors and other components of the value model that can be 'plugged in' to request a value object or to deliver a value object. *Value interface.* Actors have one or more value interfaces, grouping value ports offering and requesting value objects. It shows the mechanism of economic reciprocity. If an actor transfers value objects via its ports, the value interface shows atomicity; either all ports in a value interface precisely transfer one value object, or none at all. This ensures that if an actor offers something of value to someone else, it always gets in return what it wants. *Value transfer.* A value transfer is used to connect two value ports with each other. It represents that two actors owning the connected ports are willing to transfer value objects. This concept will also be elaborated on later. *Value activity.* Operational activities which can be assigned as a whole to actors are called a value activity. Actors perform value activities, and to do so a value activity must yield a profit or should increase economic value for the performing actor. Consequently, we only distinguish a value activity if at least one actor, but hopefully more, believes that it can execute the activity profitably. Value activities can be decomposed into smaller activities, but the same requirement stays: the activity should yield profit. This also gives a decomposition stop rule.

So far only the relations *between actors*, taking the form of value transfers, have been described. In most cases however, an actor has multiple value interfaces and these value interfaces can also be related. To be able to connect multiple value interfaces of a same actor, *dependency paths* are introduced. A dependency path connects value interfaces *within a same actor*, meaning that if one of the value interfaces is triggered the connected value interfaces also must be triggered [8]. A dependency paths consists of dependency nodes and connections. A *dependency node* is a consumer need, an AND-fork or AND-join, an OR-fork or OR-join, or a boundary element. A consumer need is the trigger for the transfer of value objects. A boundary element models that no more value transfers can be triggered. A *dependency connection* connects dependency nodes and value interfaces. It is represented by a dashed line. A *dependency path* is a set of connected dependency nodes and connections, that leads from one value interface to other value interfaces or a boundary element of the same actor. The meaning of the path is that if value objects are transfered at value interface *I*, then value interfaces pointed to by the path that starts at interface *I* are triggered according to the and/or logic of the dependency path. If a branch of the path points to a boundary element, then no more interfaces (and thus transfers) are triggered.

The main purpose of a dependency path is to reason about the *number* of value transfers as a result of the occurrence of a consumer need. If we assign economic value to the value objects transfered, and we count the number of value transfers, it is possible to calculate a net value flow sheet [8] for each actor involved. Such a sheet shows the economic feasibility, since a positive net value flow is an indication for economic sustainability. The aim of a dependency path is *not* to show any time ordering with respect to value transfers. In an e^3*value* model there is no notion of time at all [8]. The dependency paths only show which value interfaces are also triggered if a connected value interface is executed, not the order of the value transfers. To determine the order of value transfers a process model is needed [18].

3 Process Modeling

For the business process perspective, UML 2.0 activity diagrams are used (see www.uml.org). UML activity diagrams are also used by [18] and in the coordination standards BPSS [5] and RosettaNet [13]. It would however be possible to use another technique such as Petri Nets [18].

The activity diagram notion is relative simple and uses few symbols. *Ovals* represent activities, *rectangles* represent objects (data, goods or money), *unbroken arrows* represent control flows and *dashed flows* represent object flows. Control flow can be structured using *solid bars* to represent parallel splits and parallel joins, *diamonds* to represent choices, a *bullet* to point at the start of the process, and a "*lamp*" (crossed circle) represents the end of a flow. A parallel split indicates that parallel processes start. The ordering of actions in different parallel processes is not specified: if A is parallel to B, this means that A can occur before, during or after B. The activity diagram is structured in such a way that the actions of a single actor are listed in a single column. The name of actor is placed on top of the column. A column is also referred to a "swim lane".

At this point both value models and process models have been discussed. In the next section the difference between

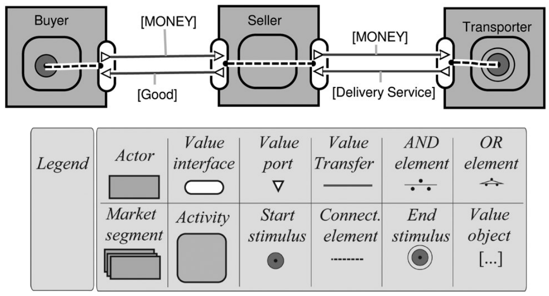

Figure 1. Example: Economic e^{3}value model

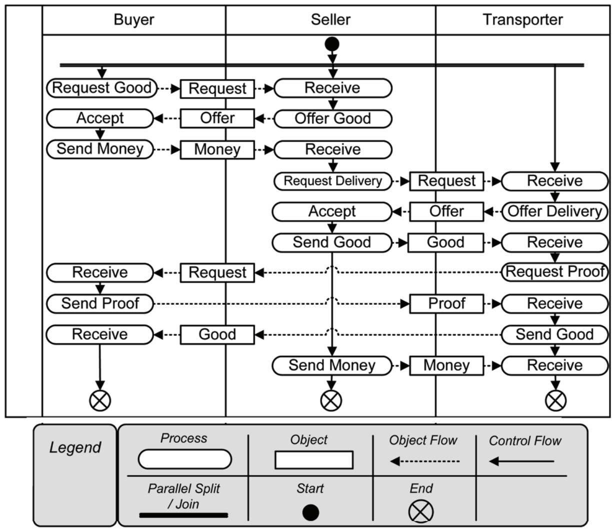

Figure 2. Example: Process Model

the models will be discussed with the help of a small example.

4 Bridging Value Models and Process Models

To demonstrate the difference between value models and process models a simple example will be given. Consider a buyer who purchases a good from a seller. The seller is however located on the other side of the world, therefore the seller hires a transporter to deliver the good to the buyer. The delivery charges are included in the price of the good. In Fig. 1 the value model is given. The value model shows the transfers of value objects between actors. A *value object* is considered something that has *economic* value for someone [18]. Furthermore, a value model shows where the value objects emerge from: value activities. The behavior of actors is not modeled in value models. Neither is the order of transfers of value objects considered. From the

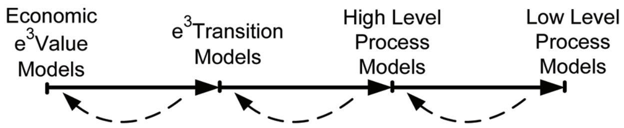

Figure 3. The e^{3}transition approach

example value model it cannot be derived if the buyer first pays or if the good is first delivered. In addition, the actual flow of the good cannot be derived from the value model. Would the flow of the good be examined with the aid of the value model, the result would be that the good directly flows from the seller to the buyer, while in reality this is not true.

It is however not the goal of the value model to incorporate such information. The value model is to understand economic reciprocity of value transfers ('one good in turn deserves another'), and to analyze the economic profitability of the value activities of the actors. Behavior and order are beyond the business value perspective and are part of the process perspective. To model these aspects a process model is needed. A process model shows the *order* of (value) activities and of transfers of value objects. In contrast to the value model, the process model also models the actual flow of value objects. In the example process model (Fig. 2) it can be seen that first the buyer requests an object, for which the seller makes an offer. Next the buyer transfers money to the seller. The seller, in turn, hires a transporter to deliver the good. Subsequently, the transporter receives the good from the seller and delivers the good to the buyer. Finally the transporter receives money for the delivery of the good. What the process model does not specify is which value objects are part of a single value transfer, nor does the process model specify which processes create value objects. Such aspects are modeled in the value model.

Both models incorporate information which is not present in the other model, hence both models conceptualize a different aspect of the same cooperation between the actors. Making a process model, which is consistent with a value model, is however a difficult task. Although in this simple example it seems to be trivial, in larger cases it is not. Our approach to arrive at such consistency is to stepwise derive a process model from a value model, leading to a greater certainty that the process model is a correct implementation of the value model.

4.1 The e^{3}transition approach

Directly deriving a process model from a value model seems difficult due to the different perspectives of both models. We propose an approach to aid in deriving a process model from a value model by incorporating a step-wise transition model. The approach is labeled the *e^{3}transition* approach. In Fig. 3 the approach is visualized. On the far left side *economic e^{3}value models* are shown, which most commonly is the starting point during the development of a multi-perspective model of a value web. Next to *economic e^{3}value* models are *e^{3}transition* models. An *e^{3}transition* model is still an *economic* model in the sense that they only show *economic* value objects. There is however one important difference: An economic *e^{3}value* model does not consider the possibility of an independent flow of the value object and the ownership right over the value object. A value transfer in an economic *e^{3}value* model implicitly implies the transfer of both the *value* object and its *ownership right*. In contrast, an *e^{3}transition* model *does* consider the possibility of an independent flow of the actual object from the ownership right. If an object is transferred independent from its ownership right, we consider this a possession transfer. Possession transfers are not seen as economic value transfers, because possession of an object by itself is not sufficient to consume, or trade an object. A transport company can *possess* an object for a while, but does not *own* the object. Only for the actor who finally receives the object, which is the actor to whom the ownership rights are transfered, does the object have value. The difference between economic value models and transition models and which steps are needed to move from the one to the other, will be elaborated in Section 4.1.1.

Right from the *e^{3}transition* model a *high level process model* is located. A high level process model specifies the *time-ordering* of value transfers and who initiates a transfer. Therefore, it is not an *e^{3}value* model anymore, but still relatively close to it, since it still shows the exchange of value objects. We elaborated on process models in section 3. The relationship between *e^{3}transition* models and high level process models will be discussed in Section 4.1.2. On the far right side *low level process models* are given. It is beyond the scope of this paper to explain how a low level process model should be made from a high level process model. It is however important to describe the difference between a high level process model and a low level process model. A high level process model contains the order of transfers of value objects as well as initiating processes. Furthermore, actors are still independent economic entities, whereas actors in a low level process model can be persons or informations systems. Furthermore, if any of the modeled processes, objects or actors in a high level process model is decomposed then it becomes a low level process model.

4.1.1 From e^{3}value to e^{3}transition Models

We now discuss the difference between economic *e^{3}value* and *e^{3}transition* models into more detail. To explain this difference, we have to elaborate on the notion of value objects further. In the *e^3value* ontology a value object is either a good or service [8]. The focus in *e^3value* is on understanding *what*, from an economic motivational point of view, of value is transfered and exchanged between actors. However, from a *legal* motivational point the goods and services themselves are not of primary interest, but the *rights* related to these goods and services (e.g. ownership right). As is noted by [17], when a product is bought an actor acquires rights over the object. In each legal system specific rights are bound to objects [14]; for example the right to sell the object, or the right to consume the object. In the setting of transferring objects from one actor to another, two rights seem to be most relevant: *ownership right* and *possession right*.

- Ownership right is best described as the right to *use* and *claim* possession of a value object [14]. If an actor has ownership right over an object, but the object is in the possession of another actor, then the actor can claim the object. Ownership rights over an object can be independent transferred from the actual object. For example, if an item is purchased by an actor via Amazone.com, then the actual good is still in the possession of Amazone.com for a few days. The ownership right over the good, however, is already transferred to the buyer. Often a proof of ownership right is needed for claiming possession of the object, most commonly this is a document. The documents in which such rights are specified are labeled *control documents* [10]. Since ownership rights on a value object entitle the right owner to somehow *use* the object, ownership rights are value objects [17] and therefore are modeled in *both* *e^3value* and *e^3transition* models.

- Possession right is the right to *have* actual (and if possible *physical*) possession of an object [14], but *not* to use the object. For instance, a transport company possesses a value object for a while during transportation of an object from a seller to a customer. Possession rights are relevant because if the ownership rights are transferred independent from the actual object, then the actual object still has to be transferred between actors. Such a transfer can only be legal if the possession rights are transferred also. So, a (value) object cannot be legally transferred independent from its possession right. The combination of possession right and an object is however not sufficient to create value out of the object. If an actor *only* has a possession right on an object, s/he is not entitled to consume or to trade the object. Consequently, we do not consider the combination of an object and its possession right as a value object. For this reason we do not include independent flows of objects and their possessions rights in economic *e^3value* models. In *e^3value* models ownership right, possession right and the actual good are always transferred simultaneously. In contrast, in *e^3transition* models we *do* consider independent transfers of an object and its possession right, since understanding the *physical* flow of objects brings us closer to process models.

So, in Fig. 1 the buyer acquires ownership right over the good. If the buyer has ownership right over the good, then the buyer can claim the good. In trading environments it is however possible that a buyer never physically claims the good, but instead sells the ownership rights over the good to a third party. The trader never has physical possession of the good. So, an economic value model does not incorporate independent transfers of ownership right on one side and the actual good in combination with its possession right on the other side. An economic value model assumes that if an actor has the ownership rights over a good, then the actor will somehow acquire possession of the good or will trade the ownership rights to a third actor. For this reason, it is not possible to *directly* derive from an economic value model the actual flow of value objects.

An *e^3transition* model (see Fig. 4) *does* include the independent flow of an object and its possession right and the independent flow of the ownership right of the object. The *e^3transition* model is still based on the *e^3value* ontology, but now independent transfers of ownership rights on an object and the actual object itself are considered also. Since the *e^3transition* model does not focus on *economic* transfers but incorporates independent ownership right and object transfers, the technique is designated the *e^3transition model*. In the *e^3transition* model still no concept of time is incorporated, therefore the model remains in the domain of business value modeling and does not enter the domain of process modeling. So, the modeler only has to think here about the difference flows of the ownership right of an object and the actual object itself, and nothing more. This is a clear and bounded step, that can be reasonably well considered by a modeler. To migrate from an economic *e^3value* model to an *e^3transition* model two steps have to be taken:

1. Adopt the same actors, value activities and customer needs from the *e^3value* model.

2. Answer the question “Is the ownership right of a value object transferred independent from the actual object?” for each value object which is transfered between two actors in the *e^3value* model. If this is not the case, copy the transfer of the value object from the *e^3value* model. Otherwise, replace the transfer of the original value object with a transfer of “Ownership right of [object]”. Furthermore, add to the model all the transfers of the actual object, in such that the original providing actor remain the same and that the original receiving actor

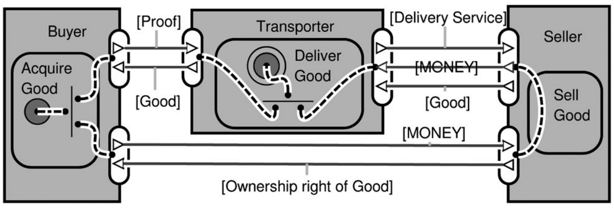

Figure 4. Example: e^3transition model

equals the last actor to whom the object is transferred. Give the transfer of the actual object the same label as original value transfer in the e^3*value* model. Because the possession right is transferred simultaneously with the object, this does not to be modeled explicitly. Note that additional information is needed to determine all the transfers of the actual object, this cannot be derived from the e^3*value* model.

For "good", in Fig. 1, it holds that the ownership right over the good is transferred independent from the actual object. Therefor "good" is replaced by "Ownership right of good". Furthermore, the good and its possession right are first transferred between the seller (the original provider of the object) and the transporter. This is done using the same value transfer as the delivery for a fee. The final addition to the original economic value model is the transfer between the transporter and the buyer (the original receiver of the object). In this transfer the good and its possession right are transferred from the transporter to the buyer in exchange for proof of ownership right (labeled "Proof"). The final e^3*transition* model is given in Fig. 4.

4.1.2 From e^3transition to processes

The e^3*transition* model still does not model behavior of actors or the time ordering of transfers. For this a process model is needed. The process model needs to show who initiates a value transfer, what the sequence of value transfers is and which processes are performed by the actors. These three aspects cannot be derived from the e^3*transition* model, what can be derived is who the actors are and what the actual flow of value objects is. To make a process model based on an e^3*transition* model the following steps have to be performed:

1. Each actor in the e^3*transition* model becomes a "swim lane" in the process model.

2. Every value transfer in the e^3*transition* model is an exchange of an object between the same actors in the process model, where the providing actor has a process "Send [object]" which is connected via the object to the process "Receive" in the receiving actor. The object in the process model equals the value object in the e^3*transition* model. There are two value transfers which are optional: the transfers of "ownership right" and of "proof". The first only has to be modeled in the process model if there is a transfer of an object in which the ownership rights are stated. For example, when the ownership right of a house is transferred an object, in which the ownership right is stated, is transferred between actors: the "deed of sales". When a person buys a magazine no proving document is transfered. "Proof" only needs to be transfered between actors if one of the actors requests the proof. In many real life situations an actor does not require proof. In this step answers should be found for the questions "Is there a document transferred between actors in which the ownership right is specified?" and "Does the actor require "proof" from another actor?". Note that additional information is needed to determine the answers for both questions.

3. A value transfer and its related reciprocal value transfer(s) have to be initiated by a participating actor via sending a "request" to another actor who is also part of the value transfers. In the process model this is seen by a "request [object]" process in the swim lane of the requesting actor connected via an object "request" to a process "receive" in the swim lane of the requested actor. This step answers the question "Who initiates a value transfer and its corresponding reciprocal value transfer(s)?". A request is optional followed by an "offer". If the negotiation process prior to the actual transfers needs to be modeled in the process model, then the transfer of an "offer" between actors is modeled with a similar notation to a "request". This was seen in the example.

4. After it has been identified which exchanges and processes should occur in the process model the exchanges and processes have to be placed in the right order. The main question here is "What is the order of the processes?". Processes can either occur sequential or parallel.

When these steps have been performed a process model has been created based on the e^3*transition* model. The execution of the process model should in the end lead to the creation of value as modeled in the economic e^3*value* model. If these steps are applied on the example then the process model based on the e^3*transition* model equals the process model in Fig. 2.

We have introduced our approach using a small and easy to understand example. In the next section, we test the approach in an industrial strength setting, namely the Dutch Aviation Sector.

5 The Dutch Aviation Sector

The Dutch Aviation Sector (which will be labeled DAS) is one of the important pillars of the Dutch economy. The DAS is responsible for an annual turnover of €20 billion and offers employment to over 80,000 people. The DAS covers a wide range of cooperating organizations and therefore forms a complex value web. Modeling the entire DAS value web would be an enormous if not impossible task. Therefore only a small part of the value creation and processes has been chosen for this paper: the landing and docking of planes. Although landing and docking might seem as activities which are performed by just one carrier, the carrier cannot actually land on its own. Assistance is needed from the Air-Traffic Control and from the airport (Amsterdam Airport Schiphol). Before the value model will be elaborated the three main actors will be discussed. The first important actor is *KLM Royal Dutch Airlines*, hereafter referred to as KLM. KLM is a carrier and is responsible for the actual transportation of persons from place A to place B. Although a plane first has to take-off before landing, in this case study only planes are considered which are already in the air and which desire to land at Amsterdam Airport Schiphol. The second main actor is *Amsterdam Airport Schiphol* (common name for NV Schiphol group), which will be labeled AAS. AAS owns and is responsible for the infrastructure at the airport. AAS is responsible for the runways as well as the gates and is also the owner of the control tower. The last main actor is Air Traffic Control the Netherlands, which will be labeled ATC. ATC is responsible for air traffic management which includes the *safe* landing of the planes as well as the taxiing of the planes from the runways to the gates.

In Fig. 5 an economic e^3*value* model for landing and docking is presented. Starting at the top value transfer: ATC acquires money from KLM by providing a landing service. For KLM to land a runway is also required, which is provided by AAS is return for money. In addition, AAS provides operational information to ATC, which is a necessary requirement for ATC to provide landing services. The costs made by AAS for providing the operational information are however charged to KLM. The described value transfer is regarding landing the plane. ATC also delivers a service of taxiing planes to a gate (the bottom value transfer), for this value transfer the same description is applicable. In the value activity "Dock Plane", performed by KLM, it is modeled that KLM can either only use the service of taxiing a plane or that KLM uses both the landing service and taxiing service. It is not possible to only use the landing service. In addition, AAS delivers two more value objects to ATC, namely a control tower and a financial service. For both value objects ATC pays AAS a fee. The line running through the value activity "Provide Landing" indicates that

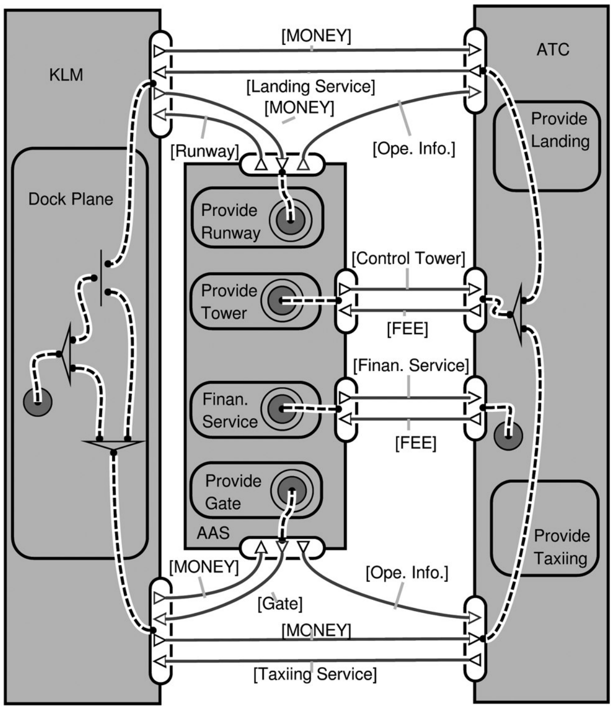

Figure 5. Dutch Aviation Sector: Economic e^{3}value model

all connected value transfers have to be executed to perform that activity.

Following the e^3*transition* approach an e^3*transition* model is made (Fig. 6). In this case, the model emphasizes the *actual* money flow. In contrast to Fig. 5, where it is shown who *conceptually* and *finally* pays to whom for provisioning a service, Fig. 6 shows the difference in the flow of *ownership rights* of money and the *actual money* for a single payment by KLM to ATC for either the landing or taxiing service. This provides a starting point for business process design. So, as can be seen from Fig. 6, KLM does not pay ATC directly, instead KLM transfers the money to AAS who transfers the money on to ATC. This is the financial service mentioned in the previous paragraph. In the e^3*transition* model it is visible that ATC receives the ownership right (labeled "ownership right of money") when one of the services is provided. When KLM uses either service an additional value transfer is executed, namely the transfer of money to AAS. This value transfer is connected to the value transfer between AAS and ATC where the money is transferred from AAS to ATC.

At this point a high level process model can be derived from the e^3*transition* model using the steps described in section 4.1.2. The process model is given in Fig. 7. The

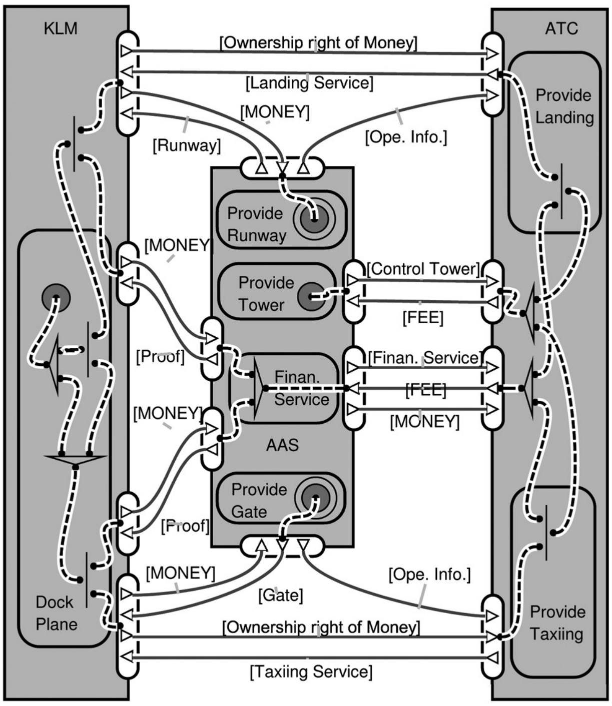

Figure 6. Dutch Aviation Sector: e³transition model

process model is simplified by modeling a single transfer of objects, loops are not considered. Nor is indicated in the model how often processes can occur within a year. The process model starts with two parallel processes. ATC first request AAS for a financial service and for a control tower. Both objects are provided by AAS and AAS receives fees from ATC for both objects. The payment is a monthly occurrence. After these processes have been completed all three actors wait for a plane to land or to dock at a gate. When this is the case KLM request ATC for the landing or taxiing service. ATC processes this request as well as the operational information, which is provided by AAS. AAS also provides a runway and gate to KLM. In a year over 400,000 planes can be landed and docked. When KLM has landed or docked a plane, KLM sends AAS a monthly amount of money. Not only the money for AAS is transfered, the money for ATC is also transfered. In the last processes AAS transfers money to ATC. Transfers of money occur on a monthly base.

If the process model is analyzed then it is visible that each actor of the e^3*transition* model is a swim lane in the process model. In addition, every value transfer is a "Send [object]" process connected via an object to a "Receive" process in the process model. The two optional value transfers described in Sec. 4.1.2 are present in the e^3*transition* model, but not in the process model. The transfer of ownership right of money between KLM and ATC is not facilitated by a transfer of a document and thus not modeled in the process model. The transfer of "proof" is also not modeled. All three actors have worked together for a long period of time and do not require proof from each other. Furthermore, there are four requests in the process model, which equals the number of value exchanges in the economic value model, but not in the e^3*transition* model. There is no request process for the value exchange of money and proof between KLM and AAS. The reason for this is that the transfer of money is automated. No request has to be made.

6 Reflective Learning

In our research the following lessons were learned when applying the proposed methodology at the DAS:

From value to process The question still remains if the execution of the process model will lead to the value transfers and creations modeled in the process model. With the aid of the e^3*transition* approach is has been possible to make a high level process model based on an economic e^3*value* model. Should the high level process model be executed then all actors would reach an end-state. In this end-state the actors would have acquired the value objects which enter the in-ports of the actor and would have provided the value objects which are send from the out-ports. If the process model and value model are compared according the correctness criteria made by [18] then both correctness criteria are met. The execution of the process model would therefor lead to the value transfers and value creation in the value model.

Loops and cardinality. How many times a value transfer occurred was difficult to incorporate into the process model. For instance, the control tower is continuously provided by AAS to ATC, while a money transfer is after fixed periods of time. This could not be properly derived from the value model, although it is somewhat possible to incorporate the cardinality of value transfers. Furthermore, to avoid complexity of the process model by adding a number of loops and constrains, the process model only shows the execution of a single landing / docking. Loops and constrains were also not included because the goals of the high level process model was only to show the order of the transfers and the initiating actor. The cardinality of value transfers and processes is however valuable information and should be consistent between the two models. This problem was dealt with by considering only a single occurrence of each value transfer and process.

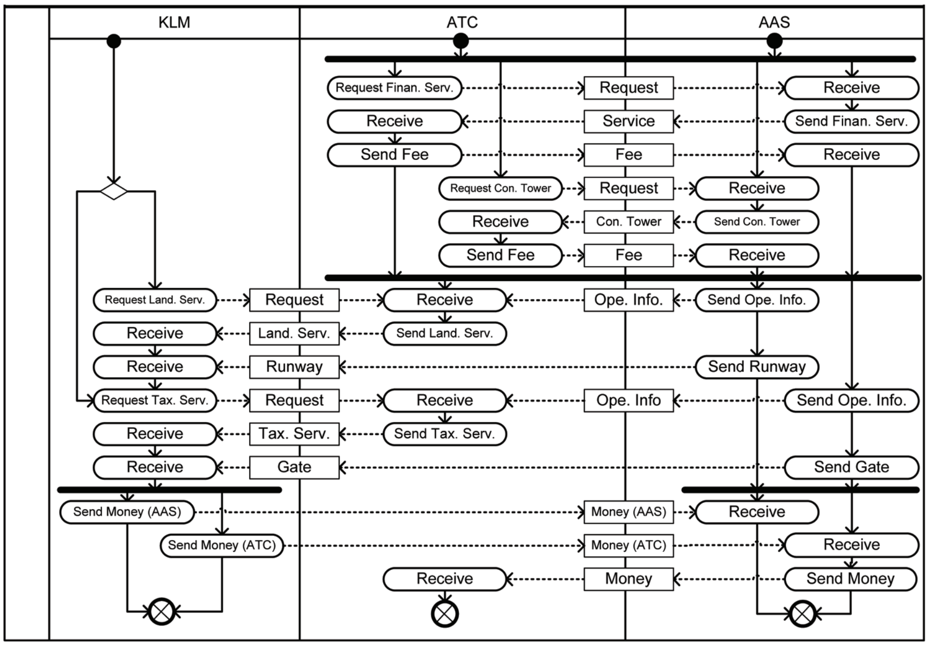

Figure 7. Processes in the Dutch Aviation Sector

Economic *e*^{3}*value* model required. Early versions of the *e*^{3}*value* model represented more the *e*^{3}*transition* model than the economic *e*^{3}*value* model. If from the *e*^{3}*transition* model a process model is derived and the *e*^{3}*transition* model can be made without the economical value model, then why is the economical value model needed? First of all, from an economic perspective the *e*^{3}*transition* model is not correct and cannot be correctly used to reason about profitabilities. Second, it was found that the earlier models focused more on the flow of goods than on *value transfers*. The model therefore represented more a process model then value model. Making a correct economic value model reduces the possibility in making such errors, due to its clear modeling guidelines and restrictions.

7 Related Work

The *e*^{3}*transition* approach is one method to make a process model next to a value model, other approaches are however also under investigation. A chaining methodology is proposed by Andersson, Bergholtz, Gregoire, Johannesson, Schmitt and Zdravkovic [4]. The chaining methodology also considers rights. The conceptualization of "rights" is however different. "Custody-flows" and "Evidence-flows" are modeled, which roughly correspond with possession right and ownership right. The chaining methodology also proposes that for each value transaction there is a negotiation process, an actualization process and a post-actualization process. Finally, for each process a pattern has to be chosen. A pattern is defined as "fixed" business processes and can prescribe that additional processes and actors have to be incorporated in the process model [4]. The combination of patterns for the processes per value transfer will lead to a final process model. The addition of these extra processes and the identification and configuration of the patterns adds complexity when deriving a process model for a value model. The *e*^{3}*transition* approach does not have that complexity, yet still reaches a similar result.

Zlatev and Wombacher do not propose a methodology for deriving a process model from a value model. Their methods assumes that a value model and process model are made independent from each other. To verify if both models are consistent the equivalence of a common semantic model is checked [19]. From both the value model and process model a reduces model is made. The reduces models are compared to identify if the original models are consistent. If there is no consistency between both models then either the process model or value model has to be modified. Which model should be modified and how this should be done is however not examined nor was the method tested in an actual setting.

8 Conclusion

In this paper, we have proposed an approach to align two different perspectives on value webs. Starting with an economic e^3*value* model, stating who offers *what* of value to *whom* and requests *what* in return. Next, this economic e^3*value* model is the basis for an e^3*transition* model. This e^3*transition* model extends the economic value model with independent transfers of *ownership rights* of an object and the actual object itself, which are ignored in economic value models because possession by itself does not carry any economic value. We found in our case study that the consideration of independent transfers of a good and its corresponding ownership right can be used as a good starting point for high level business process design. As a next step, we consider time ordering in e^3*transition* models. Decisions on time ordering can be used to create a high level process model.

What has not yet been examined is if a high level process model, derived from a value model, is a good starting point for a low level process model. Because of the assumption that a low level process model is the basis for an IT architecture (the third perspective on value webs), it is important that the low level process model is also a correct representation of the value web. Furthermore, process models were modeled as activity diagram, other techniques are however present to model process models. It has not yet been investigated if any of these other techniques might be a better match to the e^3*value* ontology then an UML activity diagram.

Acknowledgments The authors wish to thank Paul Riemens, Hans Wrekenhorst and Robert Ossenvoort from Air-Traffic Control The Netherlands for providing case study material and for having many fruitful discussions.

References

[1] J. Akkermans, Z. Baida, J. Gordijn, N. Peña, A. Altuna, and I. Laresgoiti. Value webs: Using ontologies to bundle real-world services. *IEEE Intelligent Systems*, 19(4):57–66, 2004.

[2] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services: Concepts, Architectures and Applications*. Springer, Berlin, D, 2004.

[3] J. Amsden, T. Gardner, C. Griffin, and S. Iyengar. Draft UML 1.4 profile for automated business processes with a mapping to BPEL 1.0. Technical report, IBM, Armonk, NY, June 2003.

[4] B. Andersson, M. Bergholtz, B. Gregoire, P. Johannesson, M. Schmitt, and J. Zdravkovic. From business to process models - a chaining methodology. In T. Latour and M. Petit, editors, *Proceedings of CAISE'06 Workshops and Doctoral Consortium*, pages 211–218, Namur, B, 2006. Namur Univeristy Press.

[5] ebXML. ebXML business process specification schema version 1.01. *www.ebxml.org/specs/ebBPSS.pdf*, 11 may 2001.

[6] G. Geerts and W. McCarthy. An accounting object infrastructure for knowledge-based enterprise models. *IEEE Intelligent Systems and Their Applications*, pages 89–94, July-August 1999.

[7] J. Gordijn and H. Akkermans. E3-value: Design and evaluation of e-business models. *IEEE Intelligent Systems*, 16(4):11–17, 2001.

[8] J. Gordijn and H. Akkermans. Value based requirements engineering: Exploring innovative e-commerce idea. *Requirements Engineering Journal*, 8(2):114–134, 2003.

[9] J. Gordijn, E. Yu, and B. Raadt, van der. E-service design using i* and e3value modeling. *IEEE Software*, 23(3):26–33, May 2006.

[10] V. Kartseva, J. Gordijn, and Y.-H. Tan. Inter-organisational controls as value objects in network organisations. In E. Dubois and K. Pohl, editors, *Proceedings of The 18th International Conference on Advanced Information Systems Engineering (CAiSE 2006)*, volume 4001 of *LNCS*, pages 336–350, Berlin, D, 2006. Springer.

[11] B. Nuseibeh, J. Kramer, and A. Finkelstein. A framework for expressing relationships between multiple views in requirements specification. *IEEE Transactions on Software Engineering*, 20(10):760–773, 1994.

[12] A. Osterwalder. *The Business Model Ontology - a proposition in a design science approach*. PhD thesis, University of Lausanne, Lausanne, Switzerland, 2004.

[13] RosettaNet. Rosettanet overview. *www.rosettanet.org, website*.

[14] H. Snijders and E. Rank-Berenschot. *Goederenrecht*. Kluwer, Deventer, NL, 2001.

[15] D. Tapscott, D. Ticoll, and A. Lowy. *Digital Capital - Harnessing the Power of Business Webs*. Nicholas Brealy Publishing, London, UK, 2000.

[16] K. M. van Hee. *Informations Systems Engineering - A formal approach*. Cambridge University Press, Cambridge, UK, 1994.

[17] H. Weigand, P. Johannesson, B. Andersson, M. Bergholtz, A. Edirisuriya, and T. Ilayperuma. On the notion of value object. In E. Dubois and K. Pohl, editors, *Proceedings of The 18th International Conference on Advanced Information Systems Engineering (CAiSE 2006)*, volume 4001 of *LNCS*, pages 321–335, Berlin, D, 2006. Springer.

[18] R. J. Wieringa and J. Gordijn. Value-oriented design of service coordination processes: correctness and trust. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 1320–1327, New York, NY, 2005. ACM Press.

[19] Z. Zlatev and A. Wombacher. Consistency between e3-value models and activity diagrams in a multi-perspective development method. In R. Meersman, Z. Tari, M.-S. Hacid, J. Mylopoulos, B. Pernici, Ö. Babaoglu, H.-A. Jacobsen, J. P. Loyall, M. Kifer, and S. Spaccapietra, editors, *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences*, volume 3760 of *LNCS*, Agia Napa, CY, November 2005.