Consistency Checking Between Value Models and Process Models: A Best-of-Breed Approach

Vincent Pijpers, Jaap Gordijn,

Free University, FEW/Business Informatics, De Boelelaan 1083a, 1081 HV Amsterdam, The Netherlands. {v.pijpers, Gordijn}@few.vu.nl

Abstract. The abstract should summarize the contents of the paper and should Business value models and process models describe the same subject from a different perspective. Therefore, it is important that both models are consistent with each other. To do consistency checking, we construct an intermediate model that captures the physical transfers in a value model, thereby reducing the conceptual gap between value and process models. This physical transfer model can then be checked for consistency with a process model via the already existing "reduced model" approach. A reduced model is a simplified representation of a value model or process model, where common concepts represent aspects from both the value and process model. We illustrate our approach using a small case study in the electricity sector.

Keywords: Business Processes, Value Modeling, Consistency.

1 Introduction

Value webs are groups of organizations which cooperate to jointly create value by meeting complex customers needs [13]. In earlier work it has been argued that to arrive at cross-organizational information systems for value webs, value webs should at least be analyzed from *three* different perspectives [9]: (1) the *information technology* perspective, stating information technology supporting activities of the value web, (2) the *business process* perspective, focusing on *inter*-organizational activities, e.g. described as UML activity diagrams [14], and (3) the *business value* perspective, representing what companies transfer of *economic value* between each other. Various modeling approaches have been proposed for the business value perspective, amongst others e^3*value* [4], BMO [10] and, REA [3].

Using a multi-perspective approach implies that perspectives need to be consistent with each other, as they describe the same artifact. In this paper we focus on consistency between the business value perspective (analyzed with e^3*value*) and business process perspective (analyzed with UML activity diagrams).

Currently, a few design-time approaches exist to consider consistency between e^3*value* models and activity diagrams:

- One approach is to *stepwise derive* a process model from a business model (eg. [1, 11]). In this approach the e^3*value* model is the starting point from which a new process model is designed. During the design process, the e^3*value* model triggers questions to stakeholders about the desired business processes. However, although the *e^3value* model is used as an input to find a corresponding process model, it is certainly not the case that the *e^3value* model can (even automatically) be 'translated' into a process model. The disadvantage of this approach is that it supposes that a process model is non-existent, whereas in many organizations there is already a process model. In addition these approaches neglect to verify if the derived process model is a correct representation of the original value model.

- A second approach assumes that the process model and the *e^3value* model *already co-exist next to each other*, so the process model has not been derived from the *v* [2, 15]. The value model is based on the value aspect of the value web, while the process model is based on the coordination and internal processes of organizations in the value web. To determine if the business model and process model are consistent, *reduced* models are made, which use concepts and relations that *e^3value* model process models have in common. Hereafter, the reduced models can be compared and checked on consistency. A downside of this approach is that it is possible that the original value and process model in fact are consistent, while the reduced models show otherwise (see eg. [15]).

The problems, which both approaches have to deal with, stem from the large conceptual distance between value models and process models [7]. However, after analyzing a number of *e^3value* models and process models (eg. [1, 8, 11, 15]), the difficulty in achieving consistency appears to be caused by two specific differences:

1. Value objects can be *directly* transferred between two actors in an *e^3value* model, while in a process model the physical transfer of the value object is facilitated by an *third* actor, so *indirect*. For example, if a person buys a good at store X and the delivery (by logistic party Y) is included in the purchase, no *value* transfer between the customer and logistic party Y will be present (as from a value transfer point of view, the good is transferred between the customer and the store, and not between the customer and the logistic party). In a process model however, there will be a *physical* exchange of the good between the customer and the logistic party.
2. There exist value objects, which are transferred between actors in an *e^3value* model, but who are *not* exchanged in a process model at all. For example, a ride on a roller coaster is modeled as a value transfer in an *e^3value* model, while in a process model nothing is exchanged between the actor taking the ride and the actor owning the roller coaster. Instead, the transfer requires a series of activities to be carried out. Therefore, there is no clear relationship between the value transfer and control or object flows.

To deal with these two specific issues, we combine the two solution approaches for consistency checking of *e^3value* models and process models. More specifically, we use [11] to narrow the semantic gap between value models and process models, and we use [15] to develop reduced models for an *e^3value* model and a process model, to allow for consistency checking.

In short, our proposal is to take an *e^3value* model as a starting point and subsequently:

1. Derive a model that considers the physical object flows only. These physical object flows are derived from a given *e^3value* model, following clear guidelines. We refer in the following sections to the derived *e^3value* model as an *e^3value(physical)*

model. The *e*^{3}*value(physical)* does not show *value* transfers, instead it shows *physical* transfers.

2. Determine consistency between the *e*^{3}*value(physical)* model and an activity diagram via the reduced models method [15]. We extend the reduced models with the number of occurrences of the transfers, so that we can determine if the number of occurrences of a value transfer matches the occurrences of exchanges in the process model.

The benefit of doing consistency checking this way is that by incorporating the *physical* flow of objects in an *e*^{3}*value* model the conceptual difference between value models and process models is considerably reduced. This method resolves possible problems regarding *e*^{3}*value* models and activity diagrams *before* checking consistency. In addition, we also check if the number of occurrences of a value transfer is indeed executed by the process model.

We validate our approach by analyzing if a reduced model based on the *normal* *e*^{3}*value* model (so not on the *e*^{3}*value(physical)* model) will indeed lead to *different* consistency conclusions (cf. the original proposal of [15]), compared to using the reduced *e*^{3}*value(physical)* model.

This paper is structured as follows. First, we present the running case study. Then, we apply our proposed method for achieving consistency. Hereafter we validate whether this approach does indeed result in its acclaimed benefits. We will end with presenting conclusions and making suggestions for further research.

2 Case Study: Electricity in The Netherlands

In this case study, we focus on the Dutch electricity grid. We have done extensive fieldwork in the electricity industry with respect to business value modeling (see eg. [6]). In this specific case study, a Supplier (such as Essent or Nuon) provides electricity to a Consumer, and the Consumer pays for this. Furthermore, the Supplier acquires electricity from a Producer. The Consumer must also obtain power distribution capabilities from a Distributor. In practice, cables and transformers are needed to transport the electricity, and the Consumer has to pay for this to the Distributor. Finally, the Distributor does not desire to collect the money from the various Consumers. Instead, the Distributor sells these “debts” to the Supplier; the supplier collects the debts for the Distributor and respectively gets a fee for this. These enterprises and their value flows can be found in the *e*^{3}*value* diagram in Fig. **Error! Reference source not found.**. We assume that this *e*^{3}*value* diagram is for one year, and that the Consumer has a contract period of one year also. This implies that the Consumer has one need per year. If we then count the number of value transfers, each transfer happens precisely one time per year for each Consumer.

Fig. **Error! Reference source not found.** provides a high level activity diagram for the same case study. Deliberately, the activity diagram closely follows the *e*^{3}*value* model, so that we can precisely study the conceptual differences of both notations, which have to be bridged. The Consumer requests an electricity distribution service from the Distributor, and requests also electricity from the Supplier. The Distributor receives many requests (as there are many Consumers), and delivers distribution services to each individual Consumer. For efficiency purposes, the Distributor aggregates on a quarterly basis all debts related to the distribution services offered to the Consumers, and sends the aggregated debts to the Supplier. The Supplier collects then the money for the Distributor, as can be seen later on, and pays the Distributor minus the fee for collecting the debts.

Note that the *e³value* model explicitly says that the Distributor sells the distribution service to the Consumer, and so the Distributor gets paid by the Consumer. This is indeed conceptually the case from a *value* perspective, but not from a *process* perspective. From a process point of view, the Supplier collects money on behalf of the Distributor. As said, the Consumer also requests for electricity from the Supplier. The Supplier aggregates all requests from Consumers, and obtains electricity in large amounts from a Producer. Also, the Supplier sends an invoice on a monthly basis to the Consumer. This monthly invoice includes the invoice for the distribution service. The Consumer pays the invoice, and the obtained money is used by the Supplier to pay the Distributor, and the Producer. The Producer finally, generates electricity, and delivers this electricity to the Distributor. The Distributor delivers the electricity to the Consumer. Also here, there is a difference with the *e³value* model, as from a value perspective, the electricity is provided from the Producer to the Supplier and from the Supplier to the Consumer.

So, the question right now is: Are the *e³value* model (Fig. **Error! Reference source not found.**) and the UML activity model (Fig. **Error! Reference source not found.**) consistent with each other?

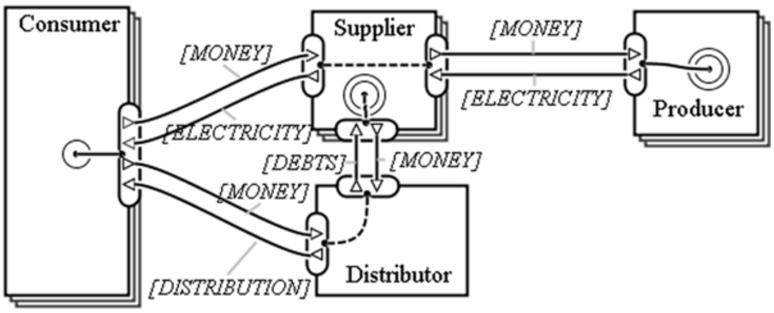

Fig. 1. *e³value*: Dutch Electricity - solid lines are *value* transfers

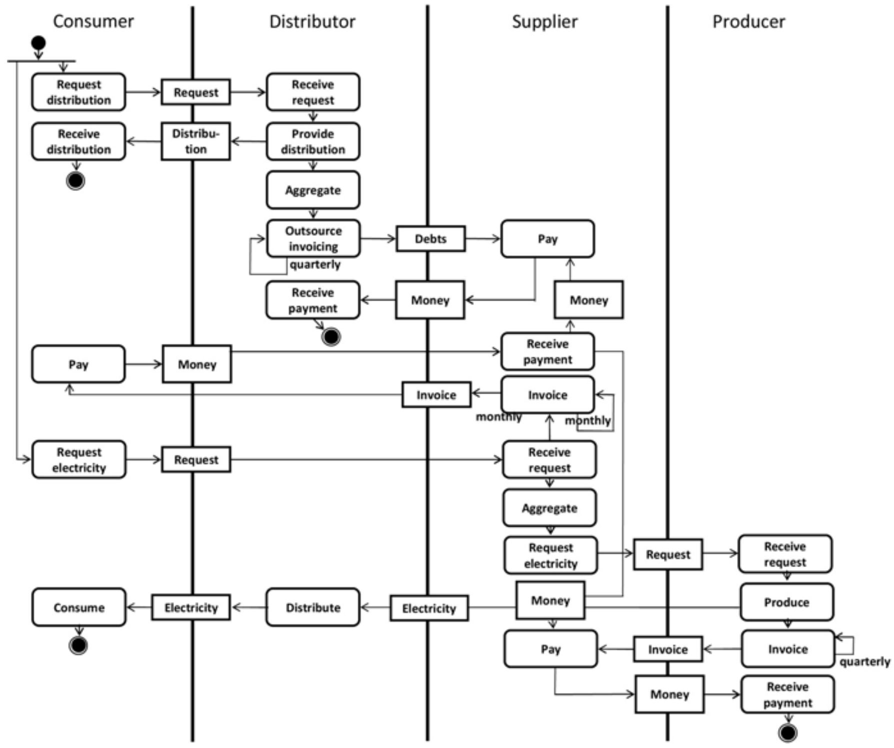

Fig. 2. Activity Diagram: Dutch Electricity

3 The e3value (physical) model

Our task is now to check whether the e^3*value* model in Fig. **Error! Reference source not found.** is consistent with the UML activity model in Fig. **Error! Reference source not found.**. To this end, we first derive an e^3*value (physical)* model from the e^3*value* model, such that the e^3*value(physical)* model will represent the *physical* transfer of the value objects rather than the *value* transfer. In addition, we count the number of occurrences of value transfers in the e^3*value* model, and propagate the found number of occurrences to the e^3*value(physical)*model. We use for the e^3*value(physical)* model the same counting mechanism for transfers as we do for the e^3*value* model, namely dependency paths. As a result, we know then how many physical transfers of value objects occur, and between whom. By considering physical transfers, rather than value transfers, we decrease the semantic gap between e^3*value* models and process models.

3.1 Counting the number of value transfers in an *e³value* model

The first step is to count the number of value transfers in an *e³value* model. In short, this is done by taking the number of consumer needs in an *e³value* model, and traversing the dependency paths connected to the consumer needs. As value transfers are part of the dependency paths, each time a value transfer is encountered while traversing, the number of occurrences for the value transfer is increased with the number of occurrences of the need that triggers the transfer. The actual algorithm to count the number of value transfers is more complex than sketched above, a tutorial can be found at [5]. For a well-formed *e³value* model, the *e³value* tool set (see www.e3value.com) can fully automatically derive the number of occurrences for all transfers in an *e³value* model. Therefore, we do not elaborate further on this step here.

3.2 Ownership versus possession

The *e³value(physical)* model considers the physical flow of objects, and not the value flow anymore. Therefore, an *e³value(physical)* model is not an *e³value* model. To make an *e³value(physical)* model, the approach as described in [11] is followed. In brief, [11] incorporates the physical flow of a value object by distinguishing between the transfer of the *ownership right* of an object and the *physical possession* of the same object.

Ownership right is best described as the right to *claim* physical possession of a value object [12]. 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 independently transferred from the actual physical object. Often a proof of ownership right is needed for claiming possession of the object; most commonly this is some document. The documents in which such rights are specified are labeled *control documents* [8]. Since an ownership right on a value object entitles the right owner to do something useful with the object (e.g. consuming or selling it), ownership rights are of value, therefore a value transfer in an *e³value* model actually imply an ownership right transfer.

In contrast to the transfer of ownership rights, there is the *physical* transfer of an object. A physical transfer of a value object implies the exchange of the physical *possession* of the object between actors. This can also mean that the “receiving” actor himself goes to the object (eg. exchange of land). Physical possession of an object is however not sufficient to create value out of a value object; If an actor *only* has physical possession of an object, s/he is not entitled to consume or to trade the object. For instance, a transport company possesses a value object for a while during transportation of the object from a seller to a customer, but the transport company has no *ownership right* of the object. Consequently, an *e³value* model does not consider possession by itself as an *economic valuable* object, and therefore does not include such aspects.

In short, an *e³value* does not differentiate between the transfer of the *ownership right* and the *physical possession* of an object. An *e³value* value model assumes that if an actor has the ownership rights over a object, 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 *e^{3}value* model the physical flow of value objects.

To reduce the conceptual gap between value models and process models, we do differentiate between the transfer of ownership rights and physical possession and subsequently model this in an *e^{3}value(physical)* model. By including the transfer of ownership rights and the physical transfer of objects we answer the question:

Q1: Is the ownership right for a value object transferred independently from the physical possession?

We ask question Q1 for each value transfer in the original *e^{3}value* model. If the answer is 'no', we just copy the value transfer from the economic value model, as this value transfer also implies a physical transfer of the same object. In *e^{3}value(physical)*, we show only these physical transfers. If the answer on Q1 is 'yes', we remove the transfer of the original value object, and we add to the *e^{3}value(physical)* model all the physical transfers related to the physical possession of the object, in such a way that the origination and final destination of the value object are still equal to the same actors as in the original *e^{3}value* model.

3.3 Counting the number of physical transfers in an *e^{3}value(physical)*model

As explained above, there is a large conceptual difference between a *value* transfer and a *physical* transfer of the same value object. A value transfer refers to a transfer of ownership, whereas a physical transfer refers to a transfer of possession. Therefore, although we use the same counting mechanism (dependency paths), the number of physical transfers may be different from the number of the corresponding value transfers. The question to ask is:

Q2: How many times is a physical transfer needed for a considered value transfer?

Answering this question requires knowledge about the business process. Consider for instance the value transfer of a value object denoting money. Usually, such a value transfer is used to model that a customer has to pay for obtaining a product. While an *e^{3}value* model represent *that* we have to pay for a product (and also *how much*, by using pricing formula's), a process model shows *how* (many times) a (partial) payment has to be done. For instance, if we construct an *e^{3}value* model with a time-scope of a year, we can show that in order to obtain electricity during that year (one value transfer), we have to pay a certain amount of money that year (also one value transfer). Both value transfers (electricity and money) occur only one time that year, and formulas indicate the amount of electricity and the money transferred. In a process model, however, we would like to say that this yearly payment can be broken down into 12 monthly periods. So, the process model shows *how* the activity of payment is done, whereas the *e^{3}value* model shows *that* the payment is done.

3.4 Case study: ***e³value***(physical)

We first use the *e³value* tool to count the number of value transfers in the *e³value* model (see Fig. **Error! Reference source not found.**). On a per Consumer basis each transfer happens once.

Second, for each of the value transfer in the *e³value* model, we analyze if the ownership right for a value object is transferred separately from the physical possession of that same value object. This is the case for three value transfers: 1) Money from Consumer to Distributor, 2) Electricity from Supplier to Consumer, and 3) Electricity from Producer to Supplier. These value transfers are not copied from the *e³value* model to the *e³value(physical)* model, but the required physical transfers of money and electricity to realize the value transfers are added to the *e³value(physical)* model. For the other value transfers, the value transfers indicate a physical transfer also, so they are copied from the *e³value* model to the *e³value(physical)* model.

Third, we have to find the number of occurrences of the physical transfers. As said, this can only be done by having knowledge about the business process, or by making assumptions about the business process, if we do not know the process on beforehand. In this case study, it can be seen from the process model that:

- The consumer pays 12 times per year for electricity, therefore for one money value transfer between the Consumer and Supplier, there are 12 physical payment transfers in the *e³value (physical)* model.
- The same holds for the ‘money value transfer between the Consumer and Supplier reflecting payment for distribution services.
- Similarly, the Supplier pays the Distributor 4 times per year. Notice that the Supplier in the physical world withholds a small fee for providing this service, so in the physical model there is no money transfer from Distributor to Supplier.
- The Supplier pays the Producer 4 times per year.
- The distribution and electricity objects refer to *continuous* production processes, therefore, we consider the number of occurrences for the related transfers as ∞.

The *e³value* tool provides support for stating the *cardinality* of a transfer. So, a transfer with a cardinality of 4, is 4 times executed per dependency path execution. Therefore, we can use the same dependency path counting mechanism as in *e³value*.

If we do not have the right knowledge about the business process (which is often the case) for finding the occurrences, the consistency checking algorithm should signal an error with respect to mismatching the number of physical transfers (in an *e³value(physical)* model and a process model respectively).

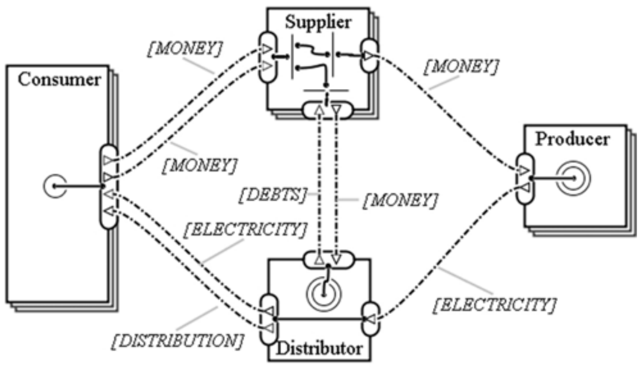

Fig. 3. *e*[3]*value(physical)* model: Dutch Electricity - dashed lines are *physical* transfers

4 Reduced Models

To determine the consistency between the e^3*value(physical)* model and the activity diagram we now make reduce models for both the e^3*value(physical)* model and the activity diagram. A reduced model is a simplified representation of a single alternative dependency path in an e^3*value* model or of a single execution sequence in an activity diagram [15] (see Fig. **Error! Reference source not found.**(a) for an example). In a reduced model concepts from an e^3*value(physical)* model *or* an activity diagram are represented by common concepts, leading to a reduced e^3*value(physical)* model *and* a reduced activity diagram.

4.1 Common concepts

The reduced models incorporate the following modeling notations:

- A *business unit* (called unit for short) corresponds to 1) an *actor* from the e^3*value(physical)* model and 2) a *swim lane* from the activity diagram. A unit is an active actor which is able to send and receive objects. A unit is not limited to organizations; it can also be a business unit [15].
- A *common object* corresponds to 1) a *value object* from the e^3*value(physical)* model and 2) an *object* from the activity diagram.
- A *common exchange* (called exchange for short) corresponds to 1) a *value transfer* in the e^3*value(physical)* model and 2) an *object exchange* in the activity diagram. An exchange represents the exchange of an object between two units disregarding order, reciprocity and bundling [15].
- An *occurrence* is given to each common exchange. The occurrence represents the number of types a common exchange occurs in a set period of time. The occurrence corresponds to 1) the exact number of times a value exchange occurs in an e^3*value(physical)* model and 2) the number of times an object exchange *can* occur in a activity diagram.

Fig. **Error! Reference source not found.** shows the visual notation of the reduced model concepts, where (a) represents a business unit, (b) represents a common object, and (c) represents a common exchange of a common object.

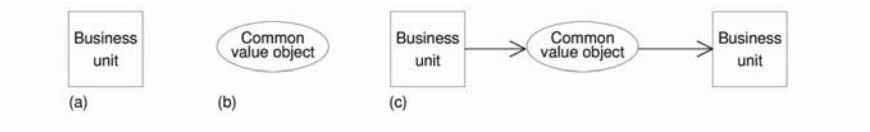

Fig. 4. Modeling notation of the reduced models

4.2 Mapping the *e³value(physical)* model and the activity diagram onto reduced models

According to [15] *value objects* from the value model can be mapped to *common objects* in three ways:

1. *One-to-none.* An object, which is present in the *e³value(physical)* model and does not have a counterpart in the process model, is disregarded in the reduced model. If we take however the *e³value(physical)* model as a starting point (and not the *e³value* model as [15] does), it is not likely that we encounter these one-to-none cases, since the *e³value(physical)* model only contains physical transfers, which should be matched by business processes.
2. *One-to-one.* If a value object has a direct relation with an object in the process model, there is also one equivalent common object in the reduced model. Therefore, this object is mapped.
3. *One-to-many*. If a physical transfer of an object in the *e³value(physical)* model corresponds to a sequence of exchanges between two swim lanes in an activity diagram, the object is also mapped, but once. Again, if we take the *e³value(physical)* model as a starting point, it is not likely that we encounter these one-to-many cases, since the *e³value(physical)* model only contains physical transfers, which should be matched by business processes. Therefore, mapping the *e³value(physical)* model on a reduced model is straightforward.

Also according to [15], objects from the process model can be mapped to *common objects* in three ways:

1. *One-to-none.* An object, which is present in the process model, and does not have a counterpart in the *e³value(physical)* model is, disregarded in the reduced model. For example, a control object (such as a request) is not considered in an *e³value(physical)* model. In other words: all objects in a process model that can not directly be related to a value object in an *e³value* model are not considered.
2. *One-to-one.* If an object in a process model has a direct relationship with an object in the *e³value(physical)* model, it is mapped to the reduced model.
3. *Many-to-one.* If a sequence of exchanges in the process model matches the exchange of a single value object, it is according to [15] represented by a single common object in the reduced model. Again, if we take the *e³value(physical)* model as a starting point, it is not likely that we encounter these sequences.

Although theoretically other mapping relationships can exist they are considered irrelevant. For example *none-to-one*, which would lead to common objects which would not be found in either the value model or process model.

4.3 Migration to reduced models

To migrate from an *e^{3}value(physical)* model *or* an activity diagram to a reduced model three steps have to be performed [15]:

1. The first step is to find all possible "execution traces" in an *e^{3}value(physical)* model (in *e^{3}value* a dependency trace) or activity diagram (in UML an execution sequence). These traces are caused by OR-forks in *e^{3}value(physical)* models and choices in activity diagrams. Since forks and choices are not always comparable in both models [15], they should not be incorporated in the reduced models. So, for each possible dependency trace and execution sequence, an independent reduced model has to be made. The approach of comparing the resulting "execution traces" independently is well known [15].
2. The next step is to make *transformation tables*. The value objects, which are not disregarded in the *e^{3}value(physical)* model, are mapped to objects for the reduced model. The same is done for objects from the process model. Actors from the value model and swim lanes from the process model are mapped to units.
3. The final step is to make the actual reduced models.

4.4 Case study: Reduced models

Reduced e^{3}value (physical) model.

From the *e^{3}value*(physical) model a reduced model, (Fig. **Error! Reference source not found.**(a)) has been made by following the steps described in the previous section. Due to space limitations the transition tables are not given. The reduced model based on the *e^{3}value* (physical) model shows that there are only one-to-one relations between the actors & business units, value objects & common objects and value transfers & common exchanges, as expected. Here, we can indeed see that from semantics point of view, the *e^{3}value* (physical) model is closer to a business process model than the original *e^{3}value* model. Additionally, both money objects, as transferred between Consumer and Supplier, can be mapped to one money object, but then this should be reflected in the amount of money transferred by that one object.

Reduced Activity Diagram.

From the activity diagram, a reduced model is made also (Fig. **Error! Reference source not found.**(b)). Again, due to space limitations the transition tables are not given. To start with, all swim lanes have been converted to units. Each object is converted to a common object except for "requests" and "invoices", since "requests" and "invoices" are control objects and do not have a counterpart in an *e^{3}value* model. Each exchange of an object between two swim lanes has been converted to a common exchange. To determine the occurrences of the exchanges we looked at the loops in the activity diagram and how often a loop occurred.

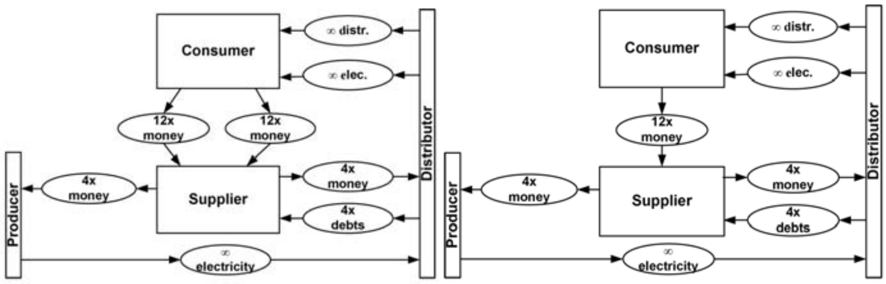

Fig. 5. Reduced models.

5 Consistency

For the business model and process model to be consistent, according to [15], there should be:

1. A correct mapping from the e^3*value(physical)* model to a reduced model. There is a correct mapping between an e^3*value(physical)* model and a reduced model when every physical transfer in the e^3*value(physical)* model is mapped to a common exchange in the reduced model. This includes that (i) every value object is represented in the reduced model, (ii) the sending and receiving actors of a value object are not mapped to a single business unit in the reduced model, and (iii) the hidden occurrence of a value transfer are made visible on the corresponding common exchange.
2. A correct mapping from an activity diagram to a reduced model, There is a correct mapping between an activity diagram and a reduced model when: every exchange in the activity diagram is mapped to a common exchange in the reduced model. This includes that (i) every object is represented in the reduced model, (ii) the sending and receiving swim lane of the object representing the exchange are not mapped to a single business unit in the reduced model, and (iii) each of common exchange is given an occurrence which corresponds to the number of times the corresponding exchange in the activity diagram can be executed.
3. Both reduced models should be equivalent. Two reduced models are equivalent if both models contain the same business units, the same common objects, the same the receiving and sending business units of a common object and, the occurrence for each common exchange of the reduced business model is equal or larger then the occurrence of the same common exchange of the reduced process model. Should this not be then the process model is not able to execute all the value transfers modeled in the business model. If the process model is capable of executing more exchanges than modeled in the business model it is not a restraint.

Case study: Consistency.

If the reduced business model (Fig. **Error! Reference source not found.**(a)) is compared to the reduced process model (Fig. **Error! Reference source not found.**(b)), it can be seen that the same business units are present, the same common objects are present and the same common exchanges are present. Only one difference can be identified: in the reduced value model money is transferred twice between Consumer and Supplier, while in the reduced process model this is only once. This is not a consistency problem, since the total amount of money exchanged is equal. Furthermore, it can be seen that each of the occurrences of the common exchanges in the reduced *e³value(physical)* is equal to the occurrences of the corresponding common exchanges in the reduced process model. Therefore it can be concluded that there is a correct mapping of both models and thus that the original *e³value* and process models are consistent.

6 Validation

It is our claim in this paper that an *e³value (physical)* model, which shows the *physical* transfer of value objects, is a necessary step to properly analyze the consistency between the original *e³value* model and a process model. To validate our claim we have made a reduced model of the original *e³value* model, cf. the guidelines of [15], and compared it to the reduced process model. Fig. **Error! Reference source not found.** provides the reduced model for the original *e³value* model (Fig. **Error! Reference source not found.**).

If this reduced model is compared to the reduced process model (Fig. **Error! Reference source not found.**(b)), then they appear to be different. For instance, in the reduced *e³value* model, there is no transfer between Supplier and Distribution Network, while in the reduced process model there is. As a result, the conclusion would be that the *e³value* model and activity diagram are not consistent; while in reality they are.

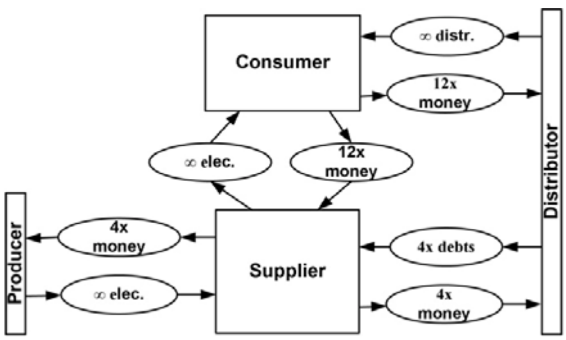

Fig. 6. Reduced model for the *e³value* model

The differences could (partially) be resolved by using the concept of *transitivity*. Transitivity removes intermediary units from a chain of common exchanges by directly representing the common exchange between the starting unit of the chain and the last unit of the chain [15]. Using transitivity however implies modifying the reduced process model such that it will match the modified *e^3value* model. This step is difficult to see for stakeholders, because in the reduced process model various money transfers occur, and identifying which should be replaced is not directly visible. Therefore, this should not be just a matter of technical model reduction; rather it reflects important conceptual knowledge about the domain at hand. This is precisely what we do with the *e^3value(physical)* model, we conceptualize the *physical* transfers as a result of *value* transfers, without considering yet the time ordering of these transfers, or the other required interactions. These become visible during business process design.

7 Related Works

Andersson, Bergholtz, Gregoire, Johannesson, Schmitt and Zdravkovic propose a chaining methodology [1]. Consistency is achieved by deriving a process model from a business model. Such a method is also proposed by Pijpers and Gordijn [11]. There are however difference between both methods. Although both first migrate from an original business model to an intermediary model by incorporating rights the conceptualization of rights is however different. Furthermore, the method of migrating from an intermediary model to a process model differs. The chaining methodology of Andersson et al. proposed that for each value transaction there is a negotiation process, an actualization process and a post-actualization process and for each process a pattern has to be chosen. A pattern is defined as fixed business processes. A pattern can prescribe that additional process and actors have to be incorporated in the process model [1]. The combination of patterns for the process per value transfer will lead to a final process model. Pijpers and Gordijn differ in this step because the map elements of the intermediary model to a high-level process model. The high-level model should be basis for any lower level process model [11].

8 Conclusions

The goal of this paper was to find a proper method for arriving at consistency between business value models and process models. We proposed, tested and validated that 1) modifying an *e^3value* model to incorporate the actual *physical* transfers of value objects and 2) comparing reduced models from the process model and the modified *e^3value* model is a clear cut and correct method for determining consistency between business and process models.

By separately stating the physical transfers of value objects in an *e^3value (physical)* model, we reduced the conceptual gap between business and process models. Also, stating value transfers that are also physical transfers reflects important domain knowledge which is necessary to arrive at a process model. This should not be hidden in a model-reduction step. As a result, the reduced models show similarity and ultimately consistency. Had this step not be taken, the reduced models would not show similarity and would incorrectly prove inconsistency.

Acknowledgments

This work has been partly sponsored by NWO project COOP 600.065.120.24N16 and the EU-funded project FENIX (518272).

References

[1] 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.

[2] L. Bodenstaff, A. Wombacher, M. U. Reichert, and R. J. Wieringa. Monitoring collaboration from a value perspective. In E. Chang and F. K. Hussain, editors, *2007 Inaugural IEEE International Conference on Digital Ecosystems and Technologies, Cairns, Australia*, volume 1, pages 134–140. IEEE Computer Society Press, February 2007.

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

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

[5] Jaap Gordijn. Assessing economic feasibility in e3value. http://e3value.few.vu.nl/docs/misc/AssessingEconomicSustainability.pdf, October 2007.

[6] Jaap Gordijn and Hans Akkermans. Business models for distributed energy resources in a liberalized market environment. *The Electric Power Systems Research Journal*, 77(9):1178–1188, 2007. Preprint available. doi:10.1016/j.epsr.2006.08.008.

[7] Jaap Gordijn, Hans Akkermans, and Hans Van Vliet. Business modelling is not process modelling. In *Conceptual Modeling for E-Business and the Web, ECOMO 2000*, volume 1921 of *LNCS*. Springer, 2000.

[8] Vera Kartseva, Jaap Gordijn, and Yao-Hua Tan. Inter-organisational controls as value objects in network organisations. In Eric Dubois and Klaus 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.

[9] 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.

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

[11] V. Pijpers and J. Gordijn. Bridging business value models and process models in aviation value webs via possession right. In *Proceedings of the 39th Hawaii International Conference On System Sciences (HICCS)*. IEEE, 2007.

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

[13] D. Tapscott, D. Ticoll, and A. Lowy. *Digital Capital - Harnessing the Power of Business Webs*. Harvard Business School Press, Boston, MA, 2000.

[14] UML. Uml 2.0. *www.uml.org, website*.

[15] Zlatko Zlatev and Andreas Wombacher. Consistency between e3-value models and activity diagrams in a multi-perspective development method. In *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.