

Dynamic Consistency between Value and Coordination Models - Research Issues

Lianne Bodenstaff *, Andreas Wombacher, and Manfred Reichert

Information Systems Group, Department of Computer Science,
University of Twente, The Netherlands
{l.bodenstaff,a.wombacher,m.reichert}@utwente.nl

Abstract. Inter-organizational business cooperations can be described from different viewpoints each fulfilling a specific purpose. Since all viewpoints describe the same system they must not contradict each other, thus, must be consistent. Consistency can be checked based on common semantic concepts of the different viewpoints. This is sufficient for equal concepts, while weakly related concepts, e.g. related to runtime behavior of viewpoints, have to be considered explicitly. In this paper we identify dynamic consistency issues correlated to the runtime behavior between value and coordination viewpoints on behalf of an example. In particular, an issue class on occurrence estimations of execution options and an issue class on granularity differences in modelling are identified and illustrated.

1 Introduction

Modelling inter-organizational business cooperations (e.g. between a seller and a buyer) constitutes a crucial task that can be done from different viewpoints. Each viewpoint emphasizes an important aspect of the cooperation. Some well-known viewpoints are the Information Systems viewpoint and the process viewpoint. In this paper two of these viewpoints are dealt with.

The *value viewpoint* gives an indication on the profitability of the cooperation for making a decision. The value model describing this viewpoint models which objects of economic value (e.g. money or goods) are exchanged between parties. Furthermore, estimations, e.g. on the number of occurrences of an object of value, are modelled. The value model enables talking about the commercial interests of the different business actors. This is important because by definition, inter-organizational design crosses commercial boundaries. The value model abstracts from

* Supported by the Netherlands Organisation for Scientific Research (NWO) under contract number 638.003.407 (Value-Based IT Alignment)

processes and object flow, i.e., it models *what* objects of value are exchanged between the different business actors but not *how* this exchange is realized. The *coordination viewpoint*, in turn, represents the interactions and interdependencies between the cooperating parties in terms of exchanged messages (e.g. an order request message). The model describing the coordination viewpoint represents *how* the actors in the model cooperate, i.e., it represents the coordination of the exchanges.

Together, the two viewpoints describe *what* is exchanged of value between the parties and *how* these exchanges can be realized. Multi-viewpoint descriptions of complex systems must maintain consistent across the viewpoints. To ensure that both models indeed describe the same cooperation, they have to be checked for consistency, i.e., we have to validate whether the overlapping system specification contained in both viewpoints is not contradicting. For example, two models are contradicting if the value model describes the exchange of money between two actors while this exchange cannot be realized by the accompanying coordination model.

Our work will build on the approach presented in [1], which provides a proper foundation for checking consistency between a value and a coordination model. So far, this approach has solely considered consistency checking of static aspects, i.e., during design time. In this paper we refer to consistency of the static aspects as *static consistency*. Static consistency checking does not consider the runtime behavior of a model. Therefore, certain aspects of a model, e.g. estimations made in the value model, are not considered. However, these estimations should still be consistent with the dynamic aspects of the coordination model. Consistency of the dynamic aspects between models will in this paper be referred to as *dynamic consistency*. In this paper we investigate the need for dynamic consistency checking using concrete examples. We identify two classes of issues concerning mismatches between value and coordination models.

To illustrate relevant issues, we use a real life, running example in which we abstract from details for the sake of simplicity. This example consists of a health insurance company which provides a one-year insurance to its customers based on a monthly paid premium. Insured customers can claim refunds for treatments they paid themselves. Furthermore, the insurance company gets money from CVZ for every paid refund to the customer. CVZ is a Dutch organization distributing tax money

from the government to the insurance companies. CVZ gets the funding on an annual basis in exchange for a proof of proper distribution of the tax money. In the following sections this example is modelled as a value and as a coordination model.

The paper is structured as follows: Section 2 and Section 3 explain in detail issues related to the value and coordination modelling. We look at general concepts of both models to illustrate the research issues. After that, static and dynamic consistency aspects with respect to the value and coordination model are discussed in Section 4. Section 5 holds the core of this paper and identifies research issues in dynamic consistency checking between the value and coordination models introduced before. In Section 6 we discuss related work. We end this paper with a summary and outlook in Section 7.

2 Value Model

For inter-organizational design the value viewpoint is especially important because all actors involved are profit-and-loss responsible, i.e., they will only participate in the cooperation if they can make a profit. Therefore, they need to be able to discuss the commercial interest for every actor involved. A value model enables a financial validation of a possible cooperation between businesses. Its purpose is to estimate the revenue for the actors in the cooperation and to assist managers in making decisions on investments concerning the cooperation. The revenue is calculated through a method of cost-benefit analysis (like e.g. Net Present Value (NPV) [2], Return on Investment (ROI) [3] and Real Options Analysis (ROA) [4]). These methods are based on the future behavior of the business concerning the exchange of objects with economic value between the actors in the model. The general concepts used in these value models are described next.

In a value model the *actors* considered as being important for the cooperation are modelled. Any cooperation has as the goal to fulfill some *consumer need*. This consumer need is realized by exchanging objects with economic value between the actors in the model. The transfer of an object with economic value will be referred to as a *value transfer* for the rest of the paper. Thus, the consumer need is fulfilled by realizing a set of value transfers between the different actors in the model. If there is more than one possibility of fulfilling the consumer need then there

is more than one set of value transfers in the value model. Generally, all possible options in fulfilling the consumer need are represented in the value model. Furthermore, the cost-benefit analysis is based on the economic value of each transfer and the *estimations* on the *number of occurrences* of the different options as well as value transfers.

In this paper we use e³-value [5] to illustrate relevant issues in dynamic consistency checking. We chose this modelling technique for illustration because of its graphical representation. However, the issues raised in this paper apply to value models in general and not solely to e³-value. e³-value was originally designed for supporting decision making on new e-business models by providing a tool that allows to calculate the profitability of all actors involved. As a method for the cost-benefit analysis, e³-value uses Net Present Value (NPV). The NPV of an actor is the amount of money an investment is worth, taking into account its cost, earnings, and the time value of money [2].

We start with explaining the basics of e³-value through a small example case. The case is the already mentioned business model of a health insurance company and one of its customers. After introducing the basics of e³-value, we explain the business case as sketched in the introduction to illustrate the more complex constructs.

2.1 e³-value - Basics

We informally describe the semantics of basic e³-value concepts [5], based on the example case from Figure 1. The example depicts two actors: an insurance company and its customer. Furthermore, two value transfers are present. One value object, premium, is transferred from the customer to the insurance company. The other value object, the insurance itself, is transferred from the insurance company to the customer. Such a combination of value transfers where both actors exchange two or more value objects in one transaction is referred to as a *value exchange* in the e³-value model. In e³-value a distinction is made between different kinds of value objects. A value object is either a *product*, *service*, *money* or *consumer experience*. In this example the premium is a value object of the money type and the insurance provided by the insurance company can be considered as a service. In Figure 1 the consumer need is “having a health insurance for one year”. This is represented by placing the *start stimulus* at the customer. Since the insurance is paid every month and the

consumer need is having insurance for *one year*, twelve payments have to be realized for fulfilling one consumer need. Therefore, an *explosion element*, annotated with 'A' in Figure 1, is added. This element multiplies every occurrence of a consumer need by twelve. Now, the set of value objects that needs to be transferred to fulfill the consumer need, consists of all value transfers connected through the *dependency path* in the model. Here, both transfers are element of the same set of transfers that fulfills the consumer need. The dependency path starts at the start stimulus and ends at the insurance company with an *end stimulus*.

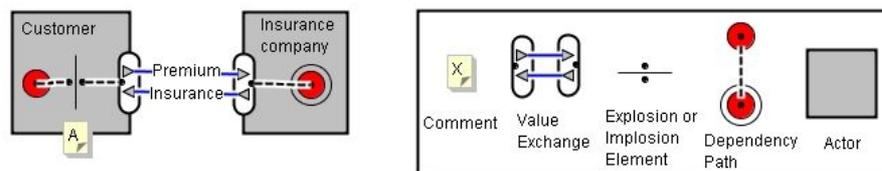


Fig. 1. e³-value model example

The estimations are made in the *profitability sheet* of the e³-value model. In this sheet the formulas necessary for calculating the revenue of the cooperation are represented. In our example, the value transfer of paying the premium is quantified in the profitability sheet by a monetary value. The premium is paid on a monthly basis for the period of one year. Therefore, the ratio on the explosion element will, in this case, be 1 : 12. After quantifying the value transfers, the revenue for every actor in the cooperation can be calculated. The profitability sheets consist of a set of Excel-sheets with the expected profit or loss amounts for every actor involved.

2.2 e³-value, the Business Case

We introduce the value model of our sample business case as explained in the introduction as an e³-value model. The graphical part of this case is depicted in Figure 2. The consumer need is “having a health insurance for one year”. Every month there are two possible sets of value transfers that can fulfill the consumer need. Either the customer claims restitution for treatments he paid for himself and he pays the monthly premium,

or he has not received treatments suitable for restitution. Therefore, he just pays the monthly premium. When the customer claims a restitution, the insurance company claims compensation from CVZ (the Dutch organization that distributes tax money from the government to the insurance companies). CVZ, in turn, gets its funding from the government. In Figure 2 these value transfers are represented. The health insurance company has multiple customers, represented as a *market segment* in the figure. Thus, the dependency path does not represent the behaviour of one customer but is an estimated average over the set of customers the insurance company has.

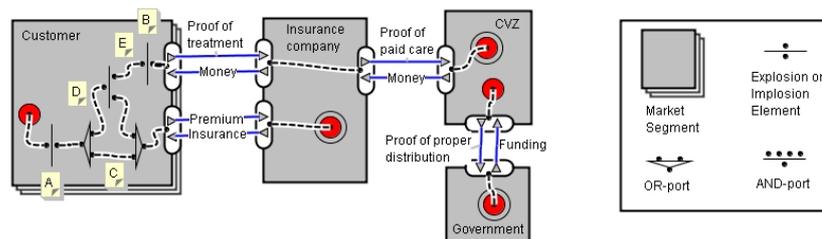


Fig. 2. e³-value model, business case

Again an explosion element, annotated with ‘A’ in Figure 2, is added to represent the twelve payments for having one year of health insurance. The choice between the two options for fulfilling the consumer need is represented as an OR-split in the figure. An OR-port represents a choice between dependency paths. After an OR-split only one of the dependency paths is chosen and in an OR-join all dependency paths entering the join can continue to the outgoing dependency path.

When the customer has not received treatments that month, the path annotated with ‘C’ is chosen. The set of value transfers in this dependency path consists of the payment of the premium by the customer and of receiving the insurance itself from the insurance company. These two value transfers are the first set of value transfers that can fulfill the consumer need.

If the customer did receive treatment that month, the path annotated with ‘D’ is chosen. This dependency path splits through an AND-split. An AND-port represents a parallel occurrence of two or more depen-

dependency paths. In an AND-split the dependency path is split into two or more paths that continue in parallel. In an AND-join all entering dependency paths share the continuation of the dependency path. The bottom path from the AND-split, represents again paying premium and receiving the insurance like the path annotated with 'C'. The two dependency paths are connected through an OR-join, indicating that the value transfers in the remaining part of the path occur in both sets of possible fulfillments of the consumer need. The upper path (annotated with 'E') represents claiming a restitution. Note that the value transfers of the upper and lower path of the AND-split together fulfill the consumer need. To enable more than one restitution claim per month, another explosion element, annotated with 'B' is added. This element is associated in the profitability sheets with a ratio representing the average claimed restitutions a month. When the customer gives proof of received treatment, the insurance company pays the restitution. The insurance company, in turn, claims restitution from CVZ. These value transfers together, are the second set of value transfers.

Within CVZ the dependency path ends and starts again. This represents the third set of value transfers. In Section 5.4 the reason for intermitting the dependency path is explained. In exchange for proof of a proper distribution of government money, CVZ gets its funding from the government.

In the profitability sheets, associated with the graphical representation of the value model, the estimations are denoted. First, the market segment customer is quantified by estimating the number of customers of the insurance company. Second, the ratio of the first explosion element, annotated with 'A' in Figure 2, is set to 1 : 12. Furthermore, the OR-split, representing the two ways of fulfilling the consumer need, is quantified with a ratio. The ratio between only payment of the premium and payment of the premium together with a restitution claim is estimated. Also the ratio at the second explosion element, annotated with 'B', is quantified in the profitability sheets. For every monetary value transfer (i.e., the premium, the restituted money to the customer, the restituted money to the insurance company and the funding from the government) the quantification is given in the profitability sheets. Now, the expected revenue for every actor in the model can be calculated.

3 Coordination Model

In a cooperation, parties interact with each other by exchanging messages. These messages are exchanged in a particular order which is not represented in the value model. The coordination of the exchange of these messages is represented in a coordination model. The purpose of the model is to check e.g. soundness, deadlock-freeness and fairness of the coordination of tasks in the cooperation. Thus, the coordination model is important to determine conceptual problems of the cooperation of the different actors at an early stage of the system development. Coordination model examples are e.g. Finite State Automata (FSA) [6], Petri Nets [7], Workflow Nets [8], message sequence charts [9], Statecharts [10–12], activity diagrams [13, 14] and flowcharts [15, 16]. Next, the general concepts used in a coordination model are explained.

In the coordination model all *actors* involved are represented. The goal is to model the *inter-organizational workflow*, i.e., the workflow between the different actors in the cooperation, to accomplish the consumer need. An inter-organizational workflow is coordinated by representing the interdependency between the *tasks* of the different actors. Furthermore, the coordination model might represent different ways in fulfilling the consumer need by allowing *choices* between different tasks in the model. Choices result in multiple sets of tasks and message exchanges that can be executed fulfilling the same need. These sets of tasks and message exchanges are to be executed in a particular order, structured in the coordination model. The ordering of the tasks and the *message exchanges* are referred to as the *execution sequences* in the coordination model.

In this paper we use Petri Nets [17] to represent coordination models. We chose Petri Nets as a representation because its graphical representation allows clear illustration of our problem statement. Furthermore, Petri Nets have a formal semantics and a variety of tools is available for defining, analyzing and simulating models. Although the issues stated in this paper are illustrated by Petri Nets, they represent general issues in dynamic consistency checking concerning a coordination model, independently from the modelling technique used.

Next, we proceed with an example to introduce the basic concepts of Petri Nets after which we describe the business case in terms of a Petri Net.

3.1 Petri Nets - Basics

The basic concepts of a Petri Net are introduced by means of an example as depicted in Figure 3. This Petri Net shows the tasks becoming necessary to fulfill the consumer need of having a health insurance for one year, introduced in Subsection 2.1.

The static part of the Petri Net consists of *places* (indicated as circles) and *transitions* (indicated as rectangles) which are connected with each other through *arcs*. Places represent message exchanges and transitions represent tasks in the coordination model represented as a Petri Net. Furthermore, the dynamic part of the Petri Net enables simulation of executions in the model. Next, the dynamic parts are described. A place can hold zero or more tokens. A distribution of tokens over places represents the state of a Petri Net. A transition is called enabled, i.e., it may fire, if each place connected to the transition with an outgoing arc, holds at least one token. When a transition fires a token is removed from each of these places and a token is put in every place connected with the transition through an incoming arc. A Petri Net and a distribution of tokens over places is also referred to as an *instance of a coordination model*.

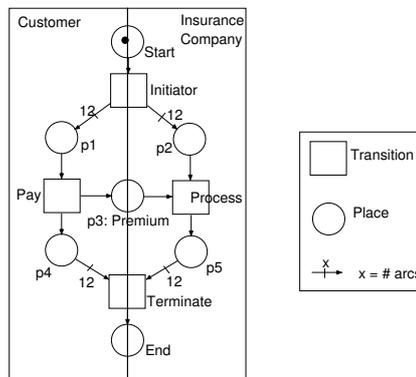


Fig. 3. Petri Net, example

When a token is placed in the *start* place in Figure 3, the *initiator* transition places twelve tokens in place *p1* of the customer as well as in place *p2* of the insurance company. These twelve tokens represent the twelve payments for having insurance for one year. Now, the transition

Pay is enabled. When the transition fires, one token is removed from place $p1$ and a token is put in place $p3$ and $p4$, respectively. As a result of the token in place $p3$, which represents the message transfer of the premium, transition *Process* is enabled. When this transition fires, a token is removed from $p3$ as well as from $p2$ and a token is put in place $p5$. When the *Pay* and the *Process* transitions have fired twelve times, the *Terminate* transition is enabled. Firing this transition removes the twelve tokens in place $p4$ and place $p5$ respectively, and puts a token in the *End* place of the model.

3.2 Petri Nets, the Business Case

In Figure 4 the sample business case as described in the introduction is represented as a coordination model in terms of a Petri Net. The model represents coordination of the interdependencies between the different actors by message exchanges. These message exchanges are modelled as places on the border between two actors. After a token is put at the start place the first *initiator* transition is enabled. If the transition fires, it enables *initiator* transitions for every actor. The customer has two parallel execution sequences. The first ensures the message exchange of the payment of the *premium* to the insurance company. The second sequence depicts the message exchanges of asking for a *restitution* to the insurance company and of receiving the restitution *payment* from the insurance company. After the customer has paid all monthly premiums and received all restitution payments from the insurance company, he sends a *complete* message that his tasks are completed to the insurance company. The insurance company receives twelve payments from the customer and, in parallel, receives zero or more requests for *restitutions*. For every request of *restitution* the insurance company sends a *payment* to the customer as well as a request for *restitution* to CVZ. After the insurance company received all payments from CVZ and has performed all payments to the customer, it sends a *complete* message to CVZ. CVZ receives, after sending a message with proof of proper distribution, *funding* for a year by the government. In parallel, CVZ receives messages from the insurance company for *restitutions* and pays the restitutions to the insurance company. After the insurance company exchanges the *complete* message and CVZ has received *funding* from the government, a token is

available in every place needed for enabling the *end* transition to terminate the process.

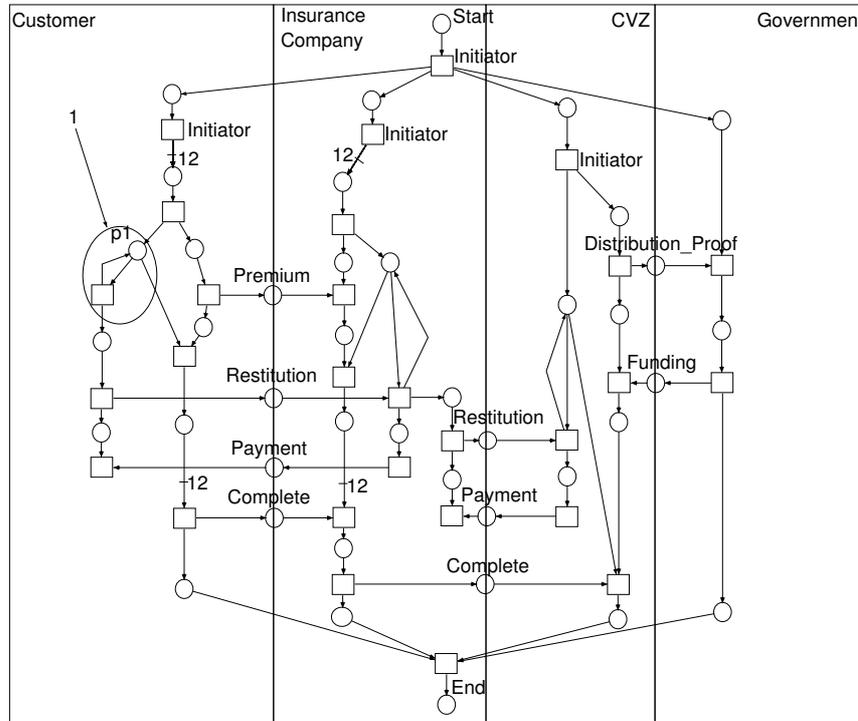


Fig. 4. Coordination model in terms of a Petri Net

4 Consistency between Value and Coordination Models

The value model and coordination model sketched in the previous sections describe the same system from different viewpoints. To ensure that both models indeed are related to the same system, we have to check whether these two viewpoints are consistent with each other. In [1] an intuitive definition of consistency between a value and a coordination model has been defined.

A value and coordination model are considered to be consistent if:

1. for every set of value transfers in the value model (dependency path in e^3 -value), there exists an execution sequence in the coordination model such that exactly the product/money value transfers contained in the set are exchanged in the execution sequence, and
2. for every execution sequence in the coordination model, there exists a set of value transfers in the value model (dependency path in e^3 -value) such that the message exchanges contained in the execution sequence represent product/money value transfers exchanged in the set of value transfers.

Note that the definition focusses on product and money value transfers, since experience and service value transfers are not instantiating an explicit message exchange. The definition is based on a relation between value transfers and message exchanges. In particular, a message exchange represents a product value transfer, if the sender and receiver of the message exchange equals the provider and the recipient of a value. With regard to the examples in Section 2.1 and 3.1, the *insurance* value transfer is a service, thus, can not be correlated with a message exchange. Further, both models contain the actors *consumer* and *insurance company*, which is a straight forward one-to-one mapping. The money value transfer *premium* in the value model is provided by the *customer* and received by the *insurance company*. A corresponding message exchange sent by the *customer* and received by the *insurance company* is also contained in the coordination model.

The definition is based on a comparison of sets of value transfers with execution sequences. The example in Section 2.1 contains a single set of value transfers similar to the example in Section 3.1 containing also only a single execution sequence. Since the elements of the set and sequence respectively can be related to each other and the number of their occurrences equals, the two models are considered consistent.

Applying this definition to the examples in Section 2.2 and 3.2, it turns out that the models are considered to be intuitively consistent although the operationalization as described in [1] determines inconsistency. This is due to the way the value model has been constructed, which is discussed in detail in Section 5.4. The derived inconsistency in this case illustrates one dimension of the consistency problem stated in this paper. Furthermore, the consistency definition mentioned so far focusses on value transfers and message exchanges only and completely

ignores the dynamics of the modelled system, resulting in estimations in the value model and observed behavior in the coordination model.

5 Research Issues

In this section we demonstrate the need for dynamic consistency checking by identifying major consistency issues that occur during runtime and could not be identified during design time. We identify two classes of issues that concern mismatches between the value and the coordination model.

The first class concerns a mismatch between the estimations made in the profitability analysis and the execution semantics of the coordination model. In this class differences in the purpose of the respective model causes issues. The purpose of the value model is to make an estimation on future revenue. This model is not used during runtime and will not become operational. As opposed to the value model, the purpose of the coordination model is to operationalize coordination of the different systems involved based on message exchanges specified in message sequences. This class represents the mismatch between the estimated number of occurrences of value transfers and choices between sets of value transfers in the value model and actual occurrences and choices of message exchanges in the coordination model.

The second class deals with the mismatch of different levels of granularity when modelling actors. It depends on the purpose of the model, which actors are represented and which actors are left out. Therefore, the chosen boundaries of the models can vary among models with different goals and different viewpoints although describing the same system. Also within a model different levels of granularity can occur, again dependent on the chosen model boundaries. This class covers mismatches of granularity differences between actors and value transfers in the value model itself as well as between the value and coordination model.

An overview of these issues is given in Table 1. Next, the two classes of issues are illustrated by the use of our example.

5.1 Issue 1: Number of Occurrences of a Value Transfer

For the fulfillment of a consumer need, value transfers have to be carried out between the different actors in the cooperation. For the fulfillment of

Class	Description	Issue
Estimations	Number of occurrences of a value transfer	Issue 1
	Choices in sets of value transfers	Issue 2
Granularity	Granularity difference between actors	Issue 3
	Granularity difference between value transfers	Issue 4

Table 1. Consistency Issues

one consumer need, a specific value transfer might occur several times. This number of occurrences may be fixed (e.g. one occurrence every month) or it can be an estimated average of occurrences of value transfers over periods of time and actors. When estimating the profitability of the cooperation in the value model, the number of expected occurrences of each value transfer compared to a single consumer need is estimated and the value of each transfer is quantified.

As an example, in e³-value the ratio between the consumer need and a value transfer as well as the ratio between two value transfers are represented by an explosion or implosion element. Regarding our example, the consumer need is ‘having a health insurance for one year’. Since the premium is paid on a monthly basis, the consumer need will be fulfilled if the premium is paid twelve consecutive months. In the value model this is modelled by adding an explosion element between the consumer need and the value transfer of the payment. As represented in Figure 5(a), this explosion element is considered in the profitability analysis by the ratio 1 : 12. We denote this ratio as a *fixed ratio* because it is the same with every customer and every case. Furthermore, a customer uses its insurance by asking one or more restitutions. This is again modelled as an explosion element with, in this example, an associated ratio of 1 : 1,5 in the profitability analysis. We denote this type of ratio as an *average ratio*.

The coordination model must contain a correspondence to the number of occurrences of value transfers as expressed in the value model. In the coordination model a value transfer with a fixed ratio is represented by forcing a fixed number of message exchanges to occur. In the case of an average ratio, a construction for enabling repetitions of tasks without a predetermined number of repetitions is used without modelling the average ratio of occurrences.

Using a Petri Net-based coordination model, the fixed ratio of monthly payments can be realized by adding an initiator transition for the cus-

customer in Figure 4. The initiator transition inserts twelve tokens, each one representing a month, for further processing of premiums and restitutions with the insurance company. This is again represented in the upper part of Figure 5(b) where this part of the coordination model is depicted. Furthermore, to allow claiming more than one restitution per month a recursive process, annotated with '1' in Figure 4, is used. In the coordination model there is no ratio represented between the monthly paid premium and the amount of restitutions as it is done in the profitability analysis of the value model. This part of the coordination model is highlighted in the lower part of Figure 5(b).

In case of having fixed ratios the consistency of the value and coordination model can be assured when designing the models by representing the ratio directly in the coordination model. When having average ratios the scenario is an example of the first class of the mismatch issue.

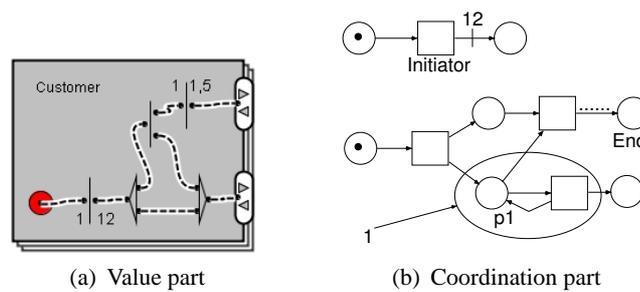


Fig. 5. Illustration of Issue 1

5.2 Issue 2: Choices in Sets of Value Transfers

If a consumer need can be fulfilled in multiple ways by carrying out different sets of value transfers, each of these sets is represented in the value model. Further, every set is associated with an expected percentage of consumer needs that will be fulfilled by using that specific set. This percentage can be estimated based on expectations or experiences (e.g. by analyzing audit trails). In the example (see Figure 2), the consumer need of having insurance can be fulfilled by either using the possibility of restitution during a monthly period or by not using this possibility and

only paying the premium. The ratio between using the restitution option and not using it is estimated by the insurance company based on all its customers and their restitution requests in previous years. This is again modelled in Figure 6(a) where the estimated ratio is modelled as 1 : 3. This average ratio is quantified in the profitability analysis.

In the coordination model the different ways of fulfilling a consumer need are represented as decisions between tasks. In our Petri Net (see Figure 4), for example, the decision point of requesting a restitution is place p1 in Figure 4 (see mark '1'). In Figure 6(b), this part of the Petri Net is again depicted where the ratio between arc a1 and a2 is not represented. Now the mismatch is that the profitability analysis of the value model is based on an average over a specific period of time while during runtime of the coordination model either restitutions occur or not. The average value can only be determined after the coordination process has been executed several times.

The issue of checking estimated ratios on the choices in sets of value transfers in the value model with the runtime instances of the coordination model, belongs to the first class of issues related to dynamic consistency.

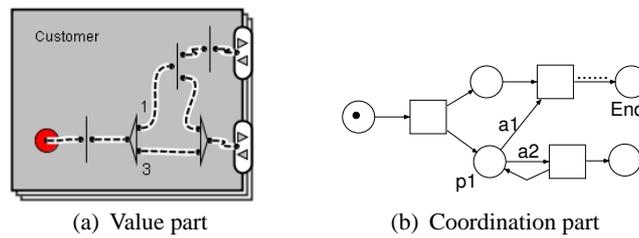


Fig. 6. Illustration of Issue 2

5.3 Issue 3: Granularity Difference between Actors

As another issue, calculation methods in value modelling use estimations on the number of occurrences of value transfers based on groups of actors. These averages are used for profitability analysis. However, in the coordination model every actor is modelled separately while several instances of actors are usually not modelled explicitly. Thus, the estimated

number of occurrences of value transfers based on an actor group in the value model cannot be directly related to the real time number of occurrences of message exchanges per actor as given in the coordination model.

In the value model, for example, the insurance company interacts with several customers who represent a market segment rather than a single actor. However, the coordination model represents the interaction between the insurance company and a single customer. Thus, the two models have different levels of granularity of actors. This is an issue for dynamic consistency checking because the average of restitutions in the value model can only be compared with the average value calculated over several instances associated to different actors of a coordination model.

A schematic example of this issue is given in Figure 7. Here, a part of the coordination and value model are depicted. The value model denotes a market segment for the customers and their relation with the insurance company. Furthermore, the coordination model denotes the interaction between one customer and the insurance company. The coordination model captures just a *fraction* of the market segment represented in the value model, i.e., one customer.

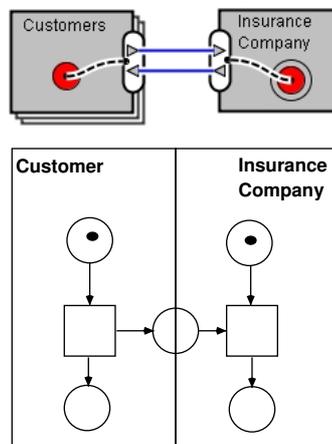


Fig. 7. Illustration of Issue 3

5.4 Issue 4: Granularity Difference between Value Transfers

Recall that the purpose of a model determines which information is represented in and which information is left out of a model. Thus, the purpose of the model influences its boundaries. If, due to the boundaries of the model, one value transfer represents the transfer of a value object concerning a market segment while another transfer represents the transfer of a value object concerning one specific actor of that market segment, a granularity difference between these value transfers occurs. The value transfer representing the single actor, concerns a fraction of the value transfer representing the market segment. Since a value transfer is atomic and therefore can not be partly executed, the relation between both value transfers cannot be straightforwardly calculated.

For example, for paying restitutions to all insurance companies, CVZ (the Dutch organization that distributes tax money from the government to the insurance companies) gets a fixed amount of funding from the government. This is a value transfer concerning the market segment of insurance companies. The purpose of the value model is to estimate the revenue of one specific actor of the market segment insurance companies. Therefore, only a fraction of the value transfer between the government and CVZ is relevant for our modelling purpose. This results in a different level of granularity between the value transfers between CVZ and the government, concerning the market segment, and the value transfers between CVZ and the particular insurance company. Therefore, the two interfaces of CVZ cannot be related through a dependency path and thus the dependency path is broken within actor CVZ as depicted in Figure 8.

In the coordination model this granularity difference is not present because every actor is modelled separately. Thus, there is a mismatch between the granularity in the value model affecting consistency with the coordination model .

Recall that the definition of static consistency is based upon matching the execution paths in the coordination model and the dependency paths in the value model (cf. Section 4). The separation of the dependency paths creates two independent paths which must be related to a single execution path in the coordination model. In the profitability analysis, however, there is a relation between both dependency paths. Thus, the estimations made in the profitability analysis about the relation of the

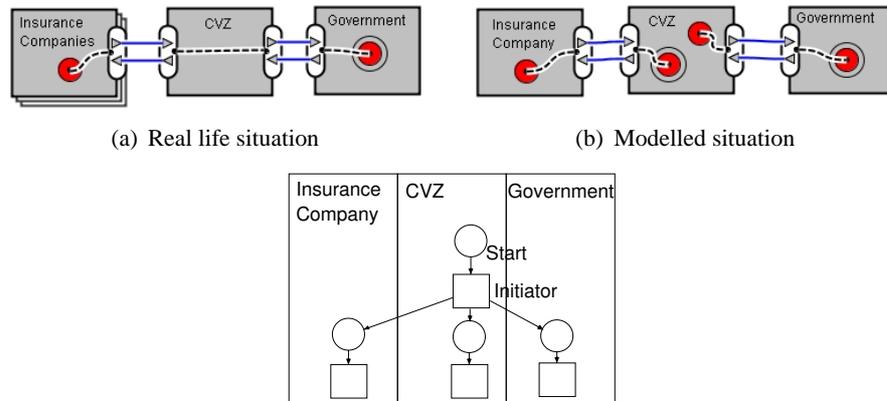


Fig. 8. Illustration of Issue 4

dependency paths have to be checked after runtime of the coordination model.

6 Related Work

Consistency between different viewpoints is an important issue addressed often in literature. In particular, there exist different ways of defining consistency within a single viewpoint as well as between different viewpoints. For instance in the workflow community different notions of consistency mainly based on deadlock-freeness have been defined on all kinds of workflow models, like e.g. Workflow Nets [18], guarded Finite State Automata [19], Coloured Place/Transition Nets [20], or statecharts [21]. Further there exist proposals to extend consistency between different models of the same viewpoint again focusing on deadlock-freeness like e.g. [18, 22, 23] for the different models.

Consistency between different viewpoints has been addressed on different levels of abstraction. An analysis on the conceptual level has been provided in [24] where the value and coordination viewpoints are compared based on the semantic concepts used in the different viewpoints have. However, this approach does not provide a means to compare two concrete models. A human intuitive consistency definition has been proposed in [25] which gives an understanding on what consistency means without explaining how to check it. This intuitive definition has been op-

erationalized in [1]. In particular, two models are consistent if for each set of value transfers there exists an associated execution sequence of message exchanges representing the value transfers, and vice versa. This consistency definition has been used as the intuitive consistency definition in Section 4. However, this consistency definition does not consider dynamic consistency as indicated in this paper.

Besides the above mentioned approaches on checking consistency between viewpoints, there exist constructive approaches guaranteeing consistency of the model derived from another model. For example in [26] an approach is proposed to use an intermediate model as a bridge between a business model and a process model. The approach is based on identifying tasks needed to accomplish the consumer need and to derive the interdependencies of these tasks. [27] propose a chaining method to derive from a business model a corresponding process model. The approach is based on associating different value transfer to off-the-shelf process patterns and combining these patterns. All these constructive approaches focus on static consistency and do not address the issues raised in this paper.

7 Summary and Outlook

In this paper we illustrate issues related to dynamic consistency checking between value and coordination models by the use of concrete examples. More specifically, we identify two classes of major consistency issues. The first class concerns mismatches between estimations made in the profitability analysis of the value model and the execution semantics of the coordination model. The second class deals with mismatches between different levels of granularity when modelling actors.

In this paper we demonstrated the need for dynamic consistency checking between value and coordination models. Furthermore, we illustrated the need for a dynamic consistency definition, since current consistency definitions, e.g. as defined in [1], cannot check consistency between two models for all aspects, i.e., dynamic as well as static aspects. The contribution of this paper is the identification and structuring of these issues. We continue this research by investigating a dynamic consistency definition to resolve the issues raised in this paper. Furthermore, these methods will be verified by means of a case study.

8 Acknowledgments

The authors would like to thank Roel Wieringa from the University of Twente and Jaap Gordijn from the Vrije Universiteit for participating in discussions and their constructive comments on earlier versions of this paper.

References

1. Zlatev, Z., Wombacher, A.: Consistency between e^3 -value models and activity diagrams in a multi-perspective development method. In: OTM Conferences (1). (2005) 520–538
2. Laudon, K., Laudon, J.: Essentials of Management Information Systems. 5 edn. Prentice Hall (2003)
3. Friedlob, G.T., Plewa Jr., F.J.: Understanding Return on Investment. John Wiley & Sons, Inc. (1996)
4. Benaroch, M.: Managing information technology investment risk: A real options perspective. *Journal of Management Information Systems* **19**(2) (2002) 43–84
5. Gordijn, J., Akkermans, J.M.: Value-based requirements engineering: Exploring innovative e-commerce ideas. *Requirements Engineering* **8**(2) (2003) 114–134
6. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison Wesley (2001)
7. Peterson, J.L.: Petri Net Theory and the Modeling of Systems. Prentice-Hall (1981)
8. Aalst, W., Hee, K.: Workflow Management - Models, Methods, and Systems. MIT Press (2002)
9. Union, I.T.: Message sequence charts. ITU-T Recommendation Z.120 (1999)
10. Harel, D.: Statecharts: A visual formalism for complex systems. *Science of Computer Programming* **8**(3) (1987) 231–274
11. Peron, A.: Statecharts, transition structures and transformations. In: Proceedings International Conference Colloquium on Trees in Algebra and Programming (CAAP- TAP-SOFT'95, Springer, LNCS 915). (1995) 454–468
12. Harel, D., Naamad, A.: The STATEMATE semantics of statecharts. *ACM Transactions on Software Engineering and Methodology* **5**(4) (1996) 293–333
13. OMG: OMG UML specification (2003) <http://www.omg.org/cgi-bin/doc?formal/03-03-01>.
14. Eshuis, R., Wieringa, R.: Tool support for verifying uml activity diagrams. *IEEE Transactions on Software Engineering* **30**(7) (2004) 437–447
15. Grefen, P., Aberer, K., Hoffner, Y., Ludwig, H.: CrossFlow: Cross-organizational workflow management in dynamic virtual enterprises. *International Journal of Computer Systems Science & Engineering* **15**(5) (2000) 277–290
16. Klingemann, J., Wäsch, J., Aberer, K.: Adaptive outsourcing in cross-organizational workflows. In: Proceedings of the the 11th Conference on Advanced Information Systems Engineering (CAiSE), Heidelberg, Germany (1999) 417–421
17. Jensen, K.: Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Springer (1997) Three Volumes.
18. Aalst, W., Weske, M.: The P2P approach to interorganizational workflows. In: Proceedings of 13. International Conference on Advanced Information Systems Engineering (CAISE), Interlaken, Switzerland (2001)
19. Fu, X., Bultan, T., Su, J.: Realizability of conversation protocols with message contents. In: Proc. IEEE Int'l. Conf. on Web Services (ICWS), IEEE Computer Society (2004) 96–103

20. Yi, X., Kochut, K.J.: Process composition of web services with complex conversation protocols: a colored petri nets based approach. In: Proc. of the Design, Analysis, and Simulation of Distributed Systems Symposium at Advanced Simulation Technology Conf. (2004) 141–148
21. Wodtke, D., Weikum, G.: A formal foundation for distributed workflow execution based on state charts. In Afrati, F.N., Kolaitis, P., eds.: Proceedings of the 6th International Conference on Database Theory (ICDT), Springer LNCS 1186 (1997) 230–246
22. Wombacher, A., Fankhauser, P., Aberer, K.: Overview on decentralized establishment of consistent multi-lateral collaborations based on asynchronous communication. In: Proc. IEEE Int'l. Conf. on e-Technology, e-Commerce and e-Service (EEE). (2005) 164–170
23. Kindler, E., Martens, A., Reisig, W.: Interoperability of workflow applications: Local criteria for global soundness. In: Business Process Management, Models, Techniques, and Empirical Studies. (2000) 235–253
24. Gordijn, J., Akkermans, J., van Vliet, J.: Business modelling is not process modelling. In: Conceptual Modeling for E-Business and the Web. Volume 1921., Springer LNCS (2000) 40–51
25. Wieringa, R.J., Gordijn, J.: Value-oriented design of service coordination processes: correctness and trust. In: Proceedings of the ACM Symposium on Applied Computing (SAC, New York, NY, USA, ACM Press (2005) 1320–1327
26. Andersson, B., Bergholtz, M., Edirisuriya, A., Ilayperuma, T., Johannesson, P.: A declarative foundation of process models. In Pastor, O., ao Falcão e Cunha, J., eds.: CAiSE2005: Proceedings of the Seventeenth International Conference on Advanced Information Systems Engineering, Springer Berlin Heidelberg (2005) 233–247
27. Andersson, B., Bergholtz, M., Grégoire, B., Johannesson, P., Schmitt, M., Zdravkovic, J.: From business to process models - a chaining methodology. In Latour, T., Petit, M., eds.: CAiSE2006: Proceedings of the Eighteenth International Conference on Advanced Information Systems Engineering, Luxembourg, Presses universitaires de Namur (Namur University Press) (2006) 211–218