

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 constitutes a crucial task that can be done from different viewpoints. Each viewpoint emphasizes an important aspect of the cooperation. In this paper two viewpoints are dealt with.

The *value viewpoint* gives an indication on the profitability of the cooperation. The value model describing this viewpoint models which objects of economic value 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 and abstracts from processes and object flow, i.e., it models *what* objects of value are exchanged 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. 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 consistency across viewpoints. To ensure both models indeed describe the same cooperation,

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

they have to be checked for consistency, i.e., we have to validate that the overlapping system specification contained in both viewpoints is not contradicting.

Our work will build on the approach presented in [1]. So far, this approach has solely considered consistency checking of static aspects, i.e., during design time, and 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. In this paper we refer to consistency of the static aspects as *static consistency* and consistency of the dynamic aspects will be referred to as *dynamic consistency*.

To illustrate relevant issues, we use a running example in which we abstract from details for the sake of simplicity. This example consists of a health insurance company which provides one-year insurance to its customers based on monthly paid premiums. 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 funding on an annual basis in exchange for a proof of proper distribution of tax money.

The paper is structured as follows: Section 2 and Section 3 explain in detail value and coordination modelling. After that consistency aspects are discussed in Section 4. Section 5 identifies research issues in dynamic consistency checking between the value and coordination models. 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. The expected revenue for every actor is calculated through a method of cost-benefit analysis (like e.g. Net Present Value (NPV) [2], Return on Investment [3] and Real Options Analysis [4]). In this paper we use e³-value [5] because of its graphical representation. However, the issues raised in this paper apply to value models in general. e³-value uses NPV for cost-benefit analysis.

We informally describe the semantics of basic e³-value concepts [5], based on Figure 1. It depicts our sample business case as explained in the introduction as an e³-value model. The example depicts four *actors* and eight *value transfers*. For example, one value object, *premium*, is transferred from the customer to the insurance company. Another value object, the *insurance* itself, is transferred from the insurance company to the customer. These two transfers are in Figure 1 annotated with an ‘F’. A combination of value transfers in one transaction is referred to as a *value exchange*. 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*.

The consumer need is “having a health insurance for one year”. This is represented by placing the *start stimulus* at the customer. 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. 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 only pays the monthly *premium*. When the customer claims a restitution, the insurance company claims compensation from CVZ. CVZ, in turn, gets its *funding* from the government. The health insurance company has multiple customers, represented as a *market segment* in the figure.

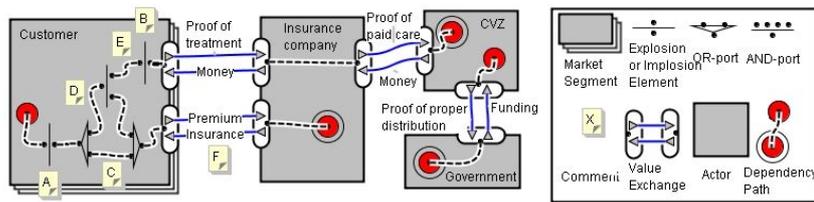


Fig. 1. e³-value model, business case

In Figure 1, the twelve monthly payments for fulfilling one consumer need are realized by adding an *explosion element*, annotated with ‘A’ in the figure, associated with ratio 1 : 12. The choice between the two options for fulfilling the consumer need is represented as an OR-split in the figure. After an OR-split only one of the dependency paths is chosen. When the customer has not received treatments that month, the path annotated with ‘C’ is chosen. The two resulting value transfers constitute the first set of transfers that can fulfill the consumer need. If the customer did receive treatment that month, the path annotated with ‘D’ is chosen. This path splits through an AND-split, representing a parallel occurrence of two or more dependency paths. In an AND-join all entering dependency paths share the continuation of the dependency path. Now, both value exchanges in the model between the insurance company and the customer occur. To enable more than one restitution claim per month another explosion element, annotated with ‘B’, is added. The insurance company claims restitution from CVZ. These value transfers together, are the second set of value transfers. The dependency path starting within CVZ represents the third set of value transfers. In Section 5.4 the reason for intermitting the dependency path is explained.

In the profitability sheets, associated with the graphical representation of the value model, the estimations are denoted. The market segment is quantified by estimating the number of customers and the ratio on the explosion elements and OR-split is set. For every monetary value transfer the quantification is denoted in

the profitability sheets. Now, the expected revenue for every actor in the model can be calculated.

3 Coordination Model

In a cooperation the messages between actors are exchanged in a particular order which is not represented in the value model. The set of ordered tasks and message exchanges is referred to as the *execution sequence*. The coordination model is important to determine conceptual problems of the cooperation at an early stage. Coordination model examples are e.g. Finite State Automata (FSA), Petri Nets [6], Workflow Nets and flowcharts.

In this paper we use Petri Nets [7] to represent coordination models because of its graphical representation, formal semantics and the variety of available tools. Although we use Petri Nets in this paper, the issues illustrated are modelling technique independent. In Figure 2 the sample business case as described in the introduction is represented as a coordination model in terms of a Petri Net. First, the basic concepts of a Petri Net are introduced after which the business case is explained in more detail.

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. Furthermore, the dynamic part of the Petri Net enables simulation of executions in the model. 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 incoming 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 outgoing arc. A Petri Net and a distribution of tokens over places is also referred to as an *instance of a coordination model*.

The 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. A recursive process, annotated with ‘1’ in Figure 2, is used to allow claiming more than one restitution per month. After the customer has paid all monthly premiums and received all restitution payments from the insurance company, he sends a *complete* message to the insurance company. For every request of *restitution* by the customer, 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

Note that the definition focusses on product and money value transfers, since experience and service value transfers are not instantiating an explicit message exchange. 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 example in Section 2 the *insurance* value transfer is a service, thus, can not be correlated with a message exchange. 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 consistency definition mentioned so far 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 value and coordination model.

The first class concerns a mismatch between the **estimations** made in the profitability analysis and the execution semantics of the coordination model. This class represents the mismatch between the estimated number of occurrences and choices between sets of 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**. Model boundaries can vary among models with different purposes although describing the same system. Furthermore, within a model different levels of granularity can occur. 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. Next, the issues are illustrated by the use of our example.

5.1 Issue 1: Number of Occurrences of a Value Transfer

For the fulfillment of one consumer need, a specific value transfer might occur several times. This number of occurrences may be fixed 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 as well as the value of each transfer is estimated.

As an example, in e^3 -value the ratio between the consumer need and a value transfer as well as the ratio between two value transfers is represented by an explosion or implosion element. Regarding our example, the consumer need will be fulfilled if the *premium* is paid twelve consecutive months. In the value model, Figure 3(a), this is modelled by adding an explosion element with 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. 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 is used.

Using a Petri Net-based coordination model, the fixed ratio of monthly payments can be realized by adding an initiator transition for the customer. The initiator transition inserts twelve tokens for further processing of premiums and restitutions, represented in the upper part of Figure 3(b). Furthermore, to allow claiming more than one restitution per month a recursive process, annotated with ‘1’ in Figure 2, 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, highlighted in the lower part of Figure 3(b).

In case of having fixed ratios consistency can be assured when designing the models while having average ratios is an example of the first class of issues.

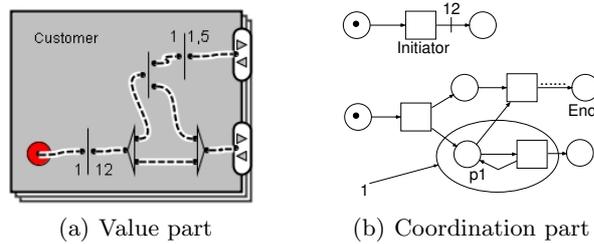


Fig. 3. Illustration of Issue 1

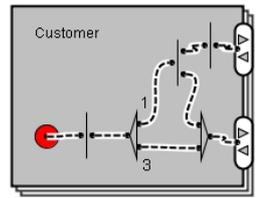
5.2 Issue 2: Choices in Sets of Value Transfers

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. In the example (see Figure 1), the consumer need can be fulfilled by either using the possibility of restitution during a monthly period or by not using this possibility. The ratio between these options is estimated by the insurance company based on all its customers and

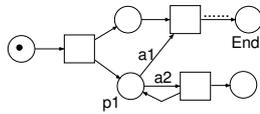
their restitution requests in previous years. This is again modelled in Figure 4(a) where the estimated ratio is modelled as 1 : 3.

In the coordination model the different ways of fulfilling a consumer need are represented as decisions between tasks. In our Petri Net (see Figure 2), for example, the decision point of requesting a restitution is place $p1$ (see mark '1'). In Figure 4(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.

Checking estimated ratios on choices in sets of value transfers with runtime instances of the coordination model, belongs to the first class of issues.

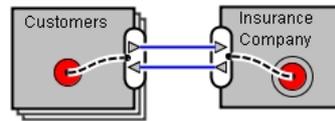


(a) Value part

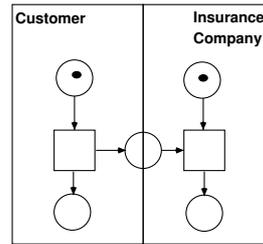


(b) Coordination part

Fig. 4. Illustration of Issue 2



(a) Value part



(b) Coordination part

Fig. 5. Illustration of Issue 3

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. However, in the coordination model every actor is modelled separately. 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 in the coordination model.

In the value model, for example, the insurance company interacts with several customers 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 5. The coordination model captures only a *fraction* of the market segment represented in the value model, i.e., one customer.

5.4 Issue 4: Granularity Difference between Value Transfers

Recall that the purpose of a model determines which information is represented in a model. 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. 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 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 Figures 6 and 7.

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.

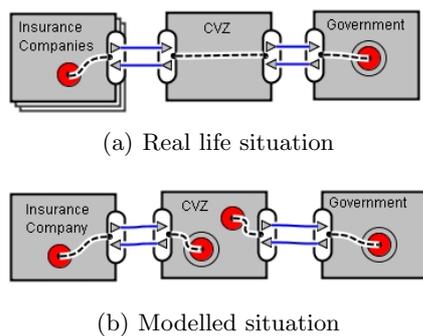


Fig. 6. Value model, Issue 4

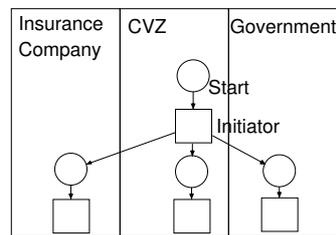


Fig. 7. Coordination model, Issue 4

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 on the relation of the 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, guarded Finite State Automata, Coloured Place/Transition Nets, or statecharts. Further there exist proposals to extend consistency between different models of the same viewpoint again focusing on deadlock-freeness like e.g. [8–10] 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 [11] where the value and coordination viewpoints are compared based on the semantic concepts used in the different viewpoints. A human intuitive consistency definition has been proposed in [12] which gives an understanding on what consistency means without explaining how to check it. This intuitive definition has been operationalized in [1]. However, this consistency definition does not consider dynamic consistency.

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 [13] an approach is proposed to use an intermediate model as a bridge between a business model and a process model. [14] 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. 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.

The authors thank Roel Wieringa and Jaap Gordijn for participating in discussions and giving their 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. Peterson, J.L.: Petri Net Theory and the Modeling of Systems. Prentice-Hall (1981)
7. Jensen, K.: Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Springer (1997) Three Volumes.
8. 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)
9. 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
10. 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
11. 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
12. 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
13. Andersson, B., Bergholtz, M., Edirisuriya, A., Ilayperuma, T., Johannesson, P.: A declarative foundation of process models. In: Proc. of the 17th International Conference on Advanced Information Systems Engineering. (2005) 233–247
14. Andersson, B., Bergholtz, M., Grégoire, B., Johannesson, P., Schmitt, M., Zdravkovic, J.: From business to process models - a chaining methodology. In: CAISE2006: Proceedings of the 18th International Conference on Advanced Information Systems Engineering, Luxembourg (2006) 211–218