E3value to BPMN Model Transformation

Hassan Fatemi, Marten Sinderen, Roel Wieringa

E^3value to BPMN Model Transformation

Hassan Fatemi, Marten van Sinderen, and Roel Wieringa

Information Systems (IS) Research Group, EEMCS Department,
University of Twente, Enschede, The Netherlands
{h.fatemi, m.j.vansinderen, r.j.wieringa}@utwente.nl

Abstract. Business value and coordination process perspectives need to be taken into consideration while modeling business collaborations. The need for these two models stems from the importance of separating the how from the what concerns. A business value model shows what is offered by whom to whom while a coordination process model shows how these offerings are fulfilled operationally. This case study addresses the model transformation between e^3value and BPMN, commonly used for modeling business collaborations from value and coordination perspectives respectively.

Keywords: Value model, coordination process model, model transformation.

1 Introduction

Value models support business design decisions, while coordination models support IT design decisions. The motivations for business decisions are commercial and strategic, while the motivations for IT decisions are technical. The business decisions must be made by business managers and the IT decisions by IT managers. Value and coordination models represent different aspects of an e-business network but they have obvious consistency relations that enable partial automated support for designing one of these models based on the other. This is not only useful for e-business design, but it also helps us to understand the similarities and differences between value and coordination models and improves our insight into the logic of the decisions that must be made in e-business network design. These insights are independent from notation.

The purpose of this case study is to define a model transformation between two languages commonly used for modeling business collaborations: the e^3value methodology [1] and the Business Process Modeling Notation (BPMN) [2].

A business web is a collection of enterprises designed to jointly satisfy a consumer need [3]. E^3value is a notation to model a business web from a value point of view. It shows the creation, distribution, and consumption of goods or services of economic value in a business web. The main goal of value modeling is to reach agreement amongst profit-and-loss responsible stakeholders regarding the question "Who is offering what of value to whom and expects what of value in return?" It also enables the stakeholders to assess their potential profitability in the business web and therefore develop an insight into the economical viability and sustainability of the whole business web. The target users of this notation are business stakeholders.

BPMN, on the other hand, is a standard notation for modeling business processes for the purposes of business analysis and its target users are business analysts.

Business value models have different goals and concepts compared to process models [4]. Nevertheless they should be consistent with each other because they both refer to the same system. A lot of researches have been done regarding generating one of the models based on the other [5-9] and checking their consistency [10-12].

In [9] we have proposed a stepwise and pattern-based method for generating a coordination model from a value model. In our transformation method, we start by finding value patterns in the value model and add their counterpart coordination patterns to the coordination model. This paper discusses the automation of this transformation process by modeling the value model in a graph and applying transformation rules on it using a graph transformation tool (Groove) [13].

2 (Business) Value Models

An e^3value model consists of a graphic part and a computational part. The graphic part is a diagram and the computational part is a spreadsheet with algorithms that can do Net Present Value (NPV) estimations for the stakeholders involved in the diagram. In e^3value we model a business web as a graph in which the nodes represent economic actors and the edges represent economic transactions. An e^3value model also shows how a consumer need is met by a set of economic transactions between actors[1].

Consider the e^3value model (Fig. 1) in which Buyer gives Money to Seller and receives Good in return and the Seller gives Money to the Transporter and receives Transport. This simple model illustrates the following modeling constructs of e^3value:

- **Contract Period.** A value model describes economic transactions during a specific period of time. It should be specified in supporting documentation.
- **Actor.** An actor is an independent economic (and often also legal) entity with a specific interest in the collaboration (making profit, increasing utility, earning experience ...). Actors in Fig. 1 are Buyer, Seller and Transporter. The actor for whom the business web is made to satisfy his needs is called the consumer. We represent the consumer need by a bullet placed inside this actor (Buyer in Fig. 1).
- **Market Segment.** A market segment is a set of actors that assign economic value to objects equally. They are shown as overlapping rectangles.
- **Value Object.** A value object is a service, good, money, or experience, that is of economic value to at least one actor and that is exchanged between actors. In our example value objects are Money, Good, Money and Transport.
- **Value Port.** An actor uses a value port to provide/request value objects to/from other actors. A value port is a conceptual construct indicating that during the contract period, an actor is capable of giving or receiving a value object. Value ports are represented by tiny triangles on the edge of the shapes depicting actors.

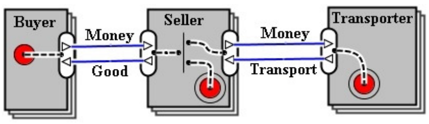


Fig. 1. A simple value model.

- **Value Interface.** Value interfaces group value ports and indicate atomicity: if one value port in the interface is triggered in the contract period, all of them are triggered in that period (however the model makes no statement about when this will happen: this has to be specified in a corresponding coordination model). Value interfaces are represented by oval shapes surrounding the value ports.
- **Value Transfer.** Value transfers link value ports of different actors, implying that the actors are willing to transfer value objects in the indicated direction.
- **Value Transaction.** Value transfers should come in economic reciprocal pairs, which are called value transactions.
- **Dependency Path.** A dependency path connects value interfaces of the same actor together, meaning that if one of the value interfaces is triggered the connected value interfaces also must be triggered [1]. It consists of dependency nodes and connections. A dependency node is a consumer need, an AND-fork (the sign in the actor Seller) or AND-join, an OR-fork or OR-join, or a boundary element (Bull's eye sign). A consumer need is the trigger for the transfer of value objects. A boundary element indicates that no more value transfers can be triggered. A dependency is represented by a dashed line. After estimating the frequencies and values of the transactions in the computational part of the value model profitability estimations can be done by tracing the dependency path of transactions that are triggered by each occurrence of the consumer need.

2.1 Value Model Example

We take an example that handles clearing Intellectual Property Rights (IPR). It has two steps: 1) collecting fees from IPR users (owners of radio stations, bars, discotheque, etc.) who play music in public spaces to get money from it, and 2) repartitioning the collected fees to Right Owners (artists, producers, publishers, etc.). IPR fee collection is currently done based on statistical evidence, but SENA (http://www.sena.nl/), one of the main IPR societies in the Netherlands, is interested in a business model in which fees are collected on a pay-per-play basis, where for each music track, a track-specific business web of clearing organizations is composed.

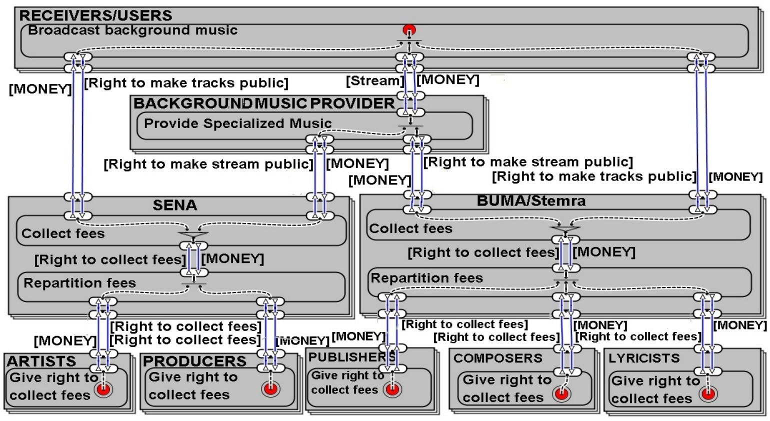


Fig. 2. Value model of providing music by Streaming.

The diagram (Fig. 2) shows a number of actors that are engaged in commercial transactions. Receivers of music (bars, restaurants, supermarkets, etc.) receive music from specialized companies that provide background music. The receivers as well as the background music providers have to pay for the rights of those involved in creating this music. In the Netherlands these rights are collected by two organizations, SENA and BUMA/Stemra each responsible for collecting rights on behalf of some specific right owners. All transactions, including music distribution, are done on-line.

3 From Value Model to Coordination Model

While there are satisfactory solutions for transforming models to text, this is not the case for transforming models to models. Graph-based-transformation approaches are inspired by theoretical work in graph rewriting. These approaches are powerful and declarative, but also complex. The complexity stems from the non-determinism in scheduling and application strategy, which requires careful consideration of termination of the transformation process and the rule application ordering.

We model the value model shown in Fig. 2 as a graph in Groove (Fig. 3). The AND/OR nodes connected to stakeholder's node by an edge labeled with 'r' indicate the logical relation (AND/OR) of the incoming/outgoing edges to/from a node. Each stakeholder node and the AND and OR nodes connected to it by an edge labeled with ‘r’ represent one integrated conceptual node. In this way, the model indicates that Receiver has three value exchanges with SENA, BMP and BUMA with an AND relation between them i.e. they make a unit of exchange which means either all of them occur or none. SENA has two value exchanges with Receiver and BMP with an OR relation between them. Upon receiving money either from Receiver or BMP, SENA should distribute it to the appropriate right owners. This graph is the starting graph in the model transformation process.

Two transformation rules are shown in Fig. 4. The rule in Fig. 4(a) indicates that if there is a pair of edges (value transfers), namely 'x' and 'y', between two distinct nodes (stakeholders), delete those two edges and add four nodes labeled with 'Send' and 'Request' connected to those two nodes by edges labeled with 'x' and 'y' as shown in Fig. 4(a). To prevent the recursive application of this rule on the newly added nodes, we need to specify in the rule that the two main nodes should be neither Send nor Request. Similarly, the rule in Fig. 4(b) indicates that if there is a single edge (value transfer), namely 'x', between two distinct nodes, delete that edge and add a node labeled with 'Send' connected to those two nodes by edges labeled with 'x' as shown in Fig. 4(b). To prevent application of this rule on the fake edges labeled with 'r' it is stipulated that the edge between the two nodes should not be labeled with 'r'. Again, to prevent the recursive application of this rule on the newly added node, we need to specify in the rule that the two main nodes should be neither Send nor Request.

The first rule has a higher priority, so Groove applies it first until it has no more matches. Then the second rule will be applied. If we don't specify priority for these two rules they conflict with each other and in case of having a pair of value exchanges between two stakeholders, instead of applying the first rule, the second rule may apply twice.

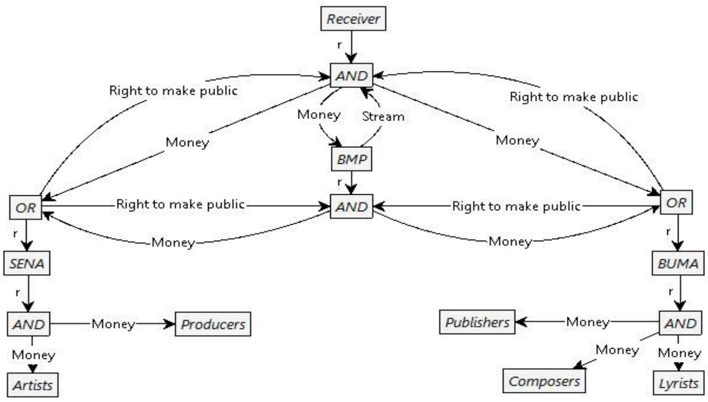
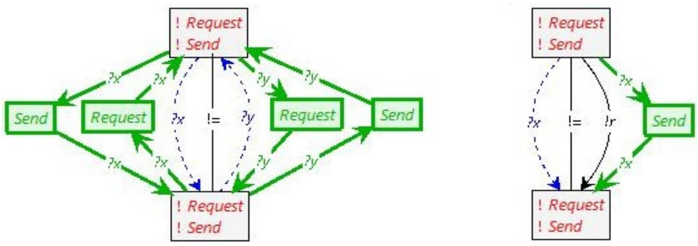


Fig. 3. Start graph for value model of providing music by streaming (Fig. 2).



(a) A pair of related value transfers (x,y) **(b) A single value transfer (x)**

Fig. 4. Two transformation rules modeled in Groove

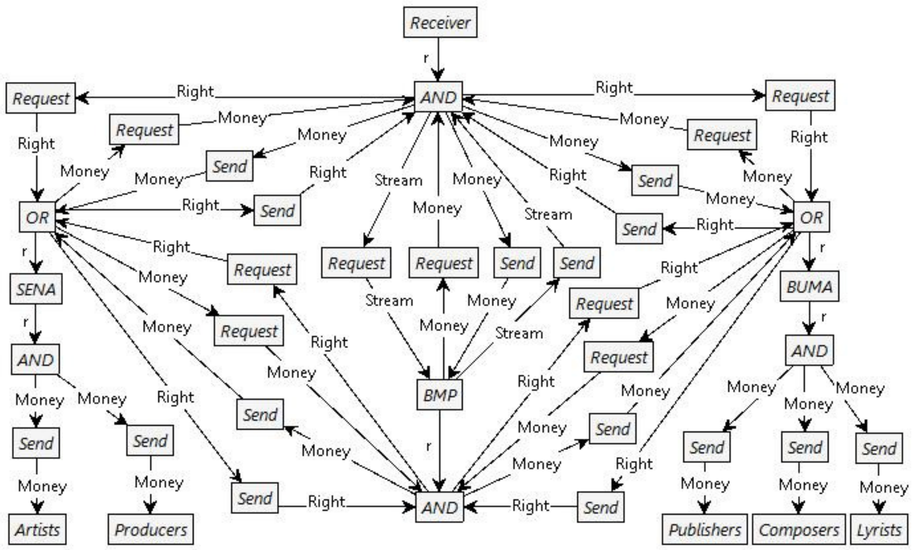


Fig. 5. Final graph after applying the above two transformation rules.

Fig. 5 shows the final graph resulting from applying the transformation rules shown in Fig. 4 on the start graph. It shows all the necessary message exchanges between the stakeholders and the logical relation between the incoming/outgoing messages to/from a stakeholder's node. Note that there is no temporal ordering between the messages in this graph. Hence, the resulting graph is an interaction/communication model rather than a coordination model.

3.1 Making the Coordination Model

As we mentioned above, the result of the transformation process (Fig. 3) is not a coordination model but rather an interaction/communication model which shows all the necessary message interactions between the stakeholders and the logical relation between the incoming/outgoing messages to/from a stakeholder's node. To make a coordination model we need to add temporal ordering to the model, i.e. indicating the order in which the messages are exchanged between the actors.

The temporal meaning of the dependency path is merely that if the consumer need is triggered in the contract period, then the connected transactions are also triggered in the contract period. The dependency path actually represents the structure of the profitability computations not a process that coordinates the transactions. Many different coordination processes are compatible with a single value model.

Putting messages in a correct and meaningful order needs expert intervention and cannot be automated using the value model only. We need to supplement the value model with more information. Some of the messages indicate the transfer of value from one actor to the other, for example the Send message labeled with 'Money' from Receiver to SENA and the Send message labeled with 'Right' from SENA to Receiver imply the transfer of money and right respectively. The order in which these value objects are exchanged between stakeholders depends among others on the trust relations between the stakeholders. If, initially, Receiver sends the money to SENA in the hope that SENA acts reciprocally by returning the requested rights, it means that Receiver trusts SENA. Otherwise, if SENA, before receiving the money, sends the requested right to Receiver in the hope that Receiver will pay later, it means that SENA trusts Receiver. The third scenario, in which both SENA and Receiver need to trust each other at some point in time, is the case in which, initially, Receiver pays not all the money but part of it to SENA and then SENA sends the requested right to Receiver and finally Receiver sends SENA the remaining of the payment.

If we had the trust relations between the stakeholders or the way in which they want to transfer the value objects, then we could accordingly determine the temporal ordering of messages in the coordination model. For further discussions on the issue of message ordering using trust relations, we refer the reader to our recent work in [14].

We derived a coordination model from the model produced by the tool, but there is no space to show the resulting coordination model (which also is not of interest for this paper anyway). Ways to support this last part of transformation with automated solutions (e.g., providing the expert with decision support, or applying constraints implied by trust relations automatically) are for further study.

4 Discussion

According to [4], business value models and coordination process models can have different actors. This happens only when the two models have different granularities. By abstracting from the sub-actors and the internal activities inside the actors, both models can always be designed using the same actors. In what follows we assume that this has been done.

One of the main remaining differences between business value models and coordination process models is type of objects exchanged by actors [4, 9]. In a value model every object should be of value to at least one partner, but in a coordination model objects are not included necessarily because they are of economic value to a partner. They can also be included because they help coordinating the activities of the partners. The application of the transformation rules covers this difference by adding all the necessary interaction messages to the model regardless of whether they are of economic value to a partner. In this paper we use a pattern with minimal interaction messages (a pair of 'Request', 'Send' messages) to realize a value transfer. In general, patterns which include more detailed pre and post interaction messages can be used.
As a difference between business value models and coordination process models, [4] mentions a special case in which there is a value exchange (e.g. experience, entertainment, pleasure …) that has no associated direct physical or information flow in the coordination model. We think all value transfers, and all objects in a coordination model, are physical. In a coordination model, we abstract from the physical world and represent the manipulation and transfer of information, but the information is of course implemented physically. But this physical implementation (on paper, as electric signals, dots on a screen etc.) is not relevant and not insightful, so we abstract from it. In a value model, all transfers are physical too: on paper (money, books, pictures etc.), as sound (music, spoken information), etc. therefore we think this special kind of value exchanges are not real value exchanges. Instead they are the cause or effect of another value exchange. For example, assume that there is an online radio station that charges you if you want to listen to its music. In this case there is one obvious value transfer from the user/listener to the radio station which is 'Money'. However the reciprocal value object may be modeled as 'entertainment' in the value model. If so, applying our pattern does not yield an appropriate result. Nevertheless we think 'entertainment' is not a value transfer but it is the effect of a value transfer, namely 'Broadcasting Music'. Therefore, in this case 'Request'and 'Send' messages are indicating asking and giving permission.

The other main difference between business value models and coordination process models mentioned by [4, 9] is the notion of temporal ordering. In an e^3value model there intentionally is no notion of time ordering at all [1]. Behavior and temporal order are beyond the value perspective and are part of the coordination perspective.

E^3value models have a value reciprocity concept which basically means every value transfer should have an associated value transfer in reverse direction. This concept has no associated counterpart in the coordination process model. E^3value also includes a computational part which enables the stakeholders to do profitability analysis which also has no associated counterpart in the coordination process model.

5 Conclusion

In this paper we address the automation of our business value model to coordination process model transformation method [9] using Groove. It turns out that we can automatically cover all the gaps caused by different factors except the one caused by the notion of time/temporal ordering, which is not present in business value models while it is a fundamental concept in coordination process models. The resulting model form the automated transformation of value model is an interaction/communication model rather than a coordination model. It needs expert intervention to add time/temporal ordering to the exchanges messages and make the final coordination process model.

References

1. Gordijn, J., Akkermans, H.: Value based requirements engineering: Exploring innovative e-commerce ideas. Requirements Engineering Journal 8, pp. 114--134 (2002)
2. OMG: Business Process Modeling Notation (BPMN) Version 1.0. OMG Final Adopted Specification. OMG, Available via http://www.bpmn.org/ (2006)
3. Tapscott, D., Ticoll, D., Lowy, A.: Digital Capital: Harnessing the Power of Business Webs. Harvard Business School Press, Boston (2000)
4. Gordijn, J., Akkermans, H., van Vliet, H.: Business modelling is not process modelling. In: ER Workshops, pp. 40--51 (2000)
5. Andersson, B., et al. : From business to process models - a chaining methodology. In: 8th Int. Conference on the Advanced Information Systems and Engineering (CAiSE'06), (2006)
6. Wieringa, R., Pijpers, V., Bodenstaff, L., Gordijn, J.: Value-driven coordination process design using physical delivery models, 27th International Conference on Conceptual Modeling ER 2008. Volume 5231 of LNCS., Springer, pp. 216--231 (2008)
7. Weigand, H., et al.: Value object analysis and the transformation from value model to process model. In: Enterprise Interoperability, Springer London, pp. 55--65 (2007)
8. Fatemi, H., van Sinderen, M.J., Wieringa, R.J.: From business value model to coordination process model. In: 2nd IFIP WG5.8 International Workshop on Enterprise Interoperability, IWEI 2009, Valencia, Spain. Volume 38 of LNBIP. Springer, pp. 94--106 (2009)
9. Fatemi, H., van Sinderen, M.J., Wieringa, R.J.: Value-oriented coordination process modeling. 8th Int. Con. on Business Process Management (BPM),LNCS,pp.162--177 (2010)
10. Zlatev, Z., Wombacher, A.: Consistency between e3-value models and activity diagrams in a multi-perspective development method, OTM Conferences (1). Volume 3760 of LNCS, Springer, pp. 520--538(2005)
11. Bodenstaff, L., Wombacher, A., Reichert, M.U.: On formal consistency between value and coordination models. Technical Report TR-CTIT-07-91, Enschede (2007)
12. Zarvi¢, N., Wieringa, R., van Eck, P.: Checking the alignment of value-based business models and it functionality. In: SAC '08: ACM symposium on Applied Computing, NY, USA, pp. 607--613 (2008)
13. Ghamarian, A.H, et al.: Modelling and analysis using GROOVE. International Journal on Software Tools for Technology Transfer (STTT), pp. 1--26, (2011)
14. Fatemi, H., van Sinderen, M.J., Wieringa, R.J.: Trust and Business Webs. In: The Fifteenth IEEE International EDOC Conference, 29 August – 2 September 2011, Helsinki, Finland. IEEE Computer Society (in press)