Opleiding Informatica & Economie

Analysis of BPMN collaboration diagrams using Petri nets

Vijanti Ramautar - 1559060

Supervisors:
Dr. H.C.M. Kleijn & Drs. P.M. Kwantes

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

August 15, 2017

Abstract

Business Process Modelling Notation (BPMN) is a popular process modelling notation. BPMN allows the user to model interactions between processes with so called collaboration diagrams (CD). However these BPMN collaboration diagrams do not have a formal definition of soundness. We investigate methods for checking the soundness of BPMN collaboration diagrams. We use Petri nets as a formal semantical framework for BPMN collaboration diagrams because soundness can be well defined for Petri nets.

In order to execute an analysis of a BPMN collaboration diagram using Peti nets, we use a case study. The case study contains a BPMN CD modelling the settlement process in the financial sector. This collaboration diagram is considered not sound, since it contains a dead end and indeed the corresponding Petri net is not sound.

Starting from this Petri net we identify the problem and we propose five different solutions. Each one improving on the previous one. So, although we do not give a generic way to guarantee the soundness of a collaboration diagram using Petri nets, we present an analysis of the collaboration diagram of the settlement process and propose a Petri net solution for the soundness problem in the settlement process.

Contents

Chapter 1

Introduction

The case we use in order to analyse BPMN collaboration diagrams using Petri nets, describes the settlement process. It is an old process that describes the exchange of securities (stocks and bonds). These securities used to be exchanged physically, by exchanging paper certificates that stated the value of the securities [Bro14]. Nowadays these securities are exchanged electronically. In the settlement process many parties are involved. This thesis is inspired by a case describing this process. The case simplifies the process by only involving three parties, two investment firms and one custodian. The custodian is the safe keeper of the securities.

The process is modelled in [KGKR15] using Business Process Modelling Notation (BPMN). BPMN is a popular process modelling notation [Whi04]. BPMN is used to model processes. It also allows the user to model interactions between two processes with the use of collaboration diagrams (CD). It is mentioned in [KGKR15] that the collaboration diagram for our case is not considered sound because of a problem regarding the timing of receiving information from different parties. This may lead to a dead end in the collaboration.

In this thesis we will analyse this CD using Petri nets. Petri nets are a well-known mathematical modelling language [vdAvHtH+11]. We choose to use Petri nets because Petri nets have a precise formal semantics, in contrast with BPMN. First we will translate the CD to a Petri net. Next we will analyse the net in order to investigate which soundness criteria are met and which are not. After demonstrating how soundness is violated in the Petri net we will investigate possible solutions. The resulting Petri nets will be created and analysed using the open-source software WoPeD (Workflow Petri Net Designer) [wop].

One of the issues that our research had to face is the fact that there are no well established formal semantics for converting a BPMN CD to a Petri net. We decide to use the method for converting a BPMN CD to a Petri net from [DDO08], although this method does not guarantee a correct outcome, as will be explained in this thesis. Another issue is that some of our Petri net solutions do not correspond to the settlement process or are not be applicable in practice.

The best outcome of this project would have been if a Petri net solution for the timing problem within the settlement process, together with a generic way for the conversion of a BPMN CD to Petri nets had been found.

Still, even though we do not come up with a generic way to convert a BPMN CD to Petri nets, we propose five Petri net solutions, each net improving on the previous one. Two of the Petri nets that we developed meet all the criteria we defined in order for a net to provide us with the desired result.

This thesis is organised as follows. Chapter 2 provides a definition overview of BPMN process diagrams and collaboration diagrams, Petri nets, workflow nets and soundness. In Chapter 3 the workflow designing tool WoPeD is introduced. This chapter focuses on the analysing methods WoPeD provides. The formal semantics that are used in this thesis to convert BPMN to Petri nets are shown in Chapter 4. The case and corresponding collaboration diagram are explained in Chapter 5. Chapter 6 proposes and evaluates five different conversions of the collaboration diagram to Petri nets. Furthermore the best proposed solution will be discussed. This thesis concludes with a summary and discussion of the results in Chapter 7. The appendix gives some information about the soundness criteria in WoPeD.

Chapter 2

Preliminaries

This chapter provides background information about BPMN. Also the relevant BPMN elements will be explained. Moreover an explanation of Petri nets and workflow nets will be given, followed by a Petri net definition of soundness. The definition of deadlocks will be given.

2.1 BPMN

BPMN stands for Business Process Modelling Notation. It was developed by the Business Process Management Initiative (BPMI) and released in May 2004 [Whi04]. In February 2006 the BPMI was subsumed by the Object Management Group (OMG). Since then they have maintained BPMN [vRSvS15]. The latest version, BPMN 2.0 has been released in January 2011. BPMN has been developed as a revision of other modelling notations such as UML and IDEF [RIRG06].

A BPMN diagram exists of graphical elements connected by a flow. That makes BPMN easily understandable for all business users. This is a primary goal of BPMN [vRSvS15]. It has to be understandable for all business stakeholders without very specific knowledge about BPMN. Another advantage of BPMN is the possibility of modelling sub-processes as activities. BPMN diagrams can be used to describe processes, provide explanatory insight or prescribe process improvements. Every individual process is modelled as a BPMN process diagram using the BPMN elements. BPMN then allows the user to combine process diagrams and their interaction in collaboration diagrams.

2.1.1 Business Process Diagram

Business process diagrams (BPD) are modelled using BPMN. A process diagram shows an abstract representation of one process. Each process diagram is modelled in a pool. The number of pools is restricted to one in process diagrams. A process diagram consists of elements of different types and of arrows. The indegree of an element is the number of incoming arrows it should have. The outdegree is the number of outgoing arrows the element should have. When no explicit indegree or outdegree is given, the element under consideration does not have a fixed number of incoming or outgoing arrows, respectively.

2.1.2 BPMN Collaboration Diagram

A BPMN collaboration diagram (BPMN CD) models the way processes interact with each other. Each process diagram is viewed as a separate pool. Pools are connected by dotted arrows involving special elements. This is called the message flow. The message flow represents communication between processes at the message events. In our case, message events are only used in order to receive messages sent by tasks. All elements, used in this thesis are shown in Table 2.1. An example of a collaboration diagram is the diagram modelling the settlement process in Section 5.

Every BPD and every BPMN CD starts with a start event. The end of a process is indicated with an end event. We assume that every process within a collaboration diagram has its own end event. In between the start and end event elements like tasks and gateways can be added. Elements within a process diagram are connected by sequence flow. In CD interactions can be modelled by adding message events. The messages sent between processes are connected by message flow, running from the task to the message event.

2.2 Petri nets

Petri nets are an old and very well defined modelling language. There are many analysis techniques that can be used to analyse Petri nets. Petri nets are directed bipartite graphs consisting of places and transitions [vdA11]. A bipartite graph is a graph with vertices divided in two disjoint sets U and V. Every edge has one vertex in U and one in V. Petri nets vertices consist of the two disjoint sets transitions T and places P. The directed edges between the transitions and places are called arcs. In this thesis we use arcs without weights. A Petri net is static but there are tokens that flow through the net. The state of a net is indicated by the distribution of tokens over places, called a marking M. The distribution of the tokens over the places at the beginning of the process is called the initial marking M_{in}. Table 2.2 provides an overview of the four Petri net elements.

Element	Description
Place	Places indicate a state or a condition for a transition to be enabled. Places can contain tokens.
Transition	A transition models a process step. A transition can fire and then consume and produce tokens.
Arc	Arcs connect places and transitions and indicate the direction of the connection.
Token	Tokens distributed over places form a marking that indicates the state of the net.

Table 2.2: Petri net elements

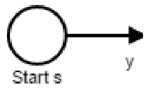
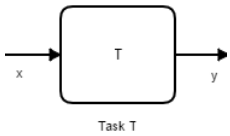
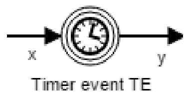
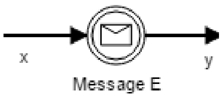
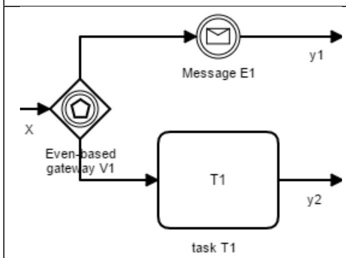
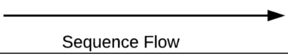
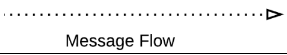
Element	Description	In PD or CD
	A start event indicates where a process will start. Indegree 0 and outdegree. 1	PD and CD
	A task represents a unit of work or an activity within a process. A task will be activated once one of its incoming sequence flows is triggered. All its outgoing sequence flows are triggered once the task is done [RPU+07].	PD and CD
	The timer event symbolises an amount of time or a delay.	PD and CD
	Communication between processes is modelled using message events. In our case the message events may receive messages from tasks. Unless the message event is also a start event.	PD and CD
	The event based gateway is triggered when a certain event occurs, indegree 1 and no explicit outdegree. It is activated when the incoming sequence flow is triggered. It triggers itself one of its outgoing arcs.	PD and CD
	The exclusive-or join is triggered when one of its incoming flows is triggered. Indegree 1 outdegree 1.	PD and CD
	The end event indicates when the process will end, outdegree 0 and indegree 1.	PD and CD
	Connects elements within a process diagram	PD and CD
	Message events and tasks are connected by a message flow between processes in a CD.	CD

Table 2.1: All relevant BPMN elements for our case

So, a Petri net is a triple $N = (P, T, F)$where:

P and T are finite sets of places and transitions, respectively with $P \cap T = \emptyset$

$F \subseteq (P \times T) \cup (T \times P)$ is the set of arcs of N such that:

for every $t \in T$ there exist $p, q \in P$ such that $(p, t), (q, t) \in F$ and

for every $t \in T$ and $p, q \in P$ if $(p, t), (t, q) \in F$, then $p \neq q$.

A place p is an input place of transition t if $(p, t) \in F$. Similarly if $(t, p) \in F$, then p is an output place of t, moreover if p is an input (output) place of a transition t, tgen t is an output (input, respectively) transition of p. By $p\bullet$ we denote the set of output transitions $t : (p, t) \in F$ of p; and by $\bullet p = t : (t, p) \in F$ its set of input transitions. The set of input places of a transition t is denoted by $\bullet t$; the set of output places is denoted by $t\bullet$. As stated before, tokens can flow through Petri nets. The distribution of the tokens over places indicates the state of a net. A marking is a function $M : P \to \mathbb{N}$, where $M(p)$ is the number of tokens in p assigned to p by M. The marking of a net can change when transitions occur ("fire"). Transitions can only fire when they are enabled. A transition is enabled once each of its input places contains at least one token. When a transition is enabled it removes one token from each of its input places and adds one token to every output place [vdA11].

So, let (N, M) be a marked Petri net i.e. a Petri net $N = (P, T, F)$ together with a marking M. Transition $t \in T$ is enabled, denoted $(N, M)[t >$, if and only if for all $p \in \bullet t$ we have $M(p) \geq 1$.

$(N, M)[t > (N, M')$ if t enabled at M and

$\forall p \in \bullet t : M'(p) = M(p) - 1$

$\forall p \in t\bullet : M'(p) = M(p) + 1$

$\forall p \notin \bullet t \cup t\bullet : M'(p) = M(p)$

Figure 2.1, 2.2 and 2.3 demonstrate the firing rule. Repeating the firing rule of transitions leads to a firing sequence. The reachability graph is a rooted edge-labelled directed graph with the initial marking at its root and a labelled edge (M, t, M') whenever $M[t > M'$.

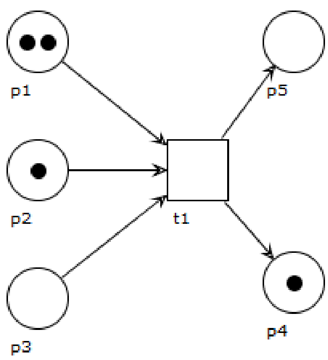


Figure 2.1: Transition t_1 is not enabled, because not all of its input places (p_1, p_2, p_3) contain a token

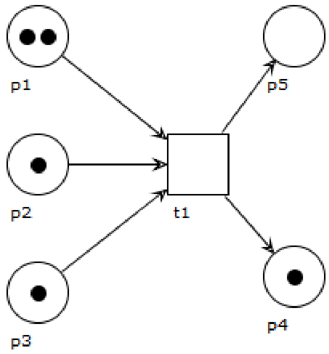


Figure 2.2: Transition t_1 is enabled, because all of its input places (p_1, p_2, p_3) contain a token

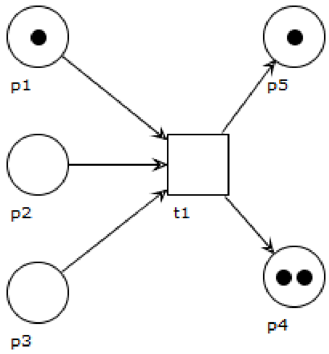


Figure 2.3: Transition t_1 has fired from the marking in Figure 2.2. t_1 has consumed one token from each input place (p_1, p_2, p_3). t_1 has fired tokens to all its output places (p_4, p_5)

2.2.1 Workflow net

A workflow net WN is a Petri net together with an intial marking M_{in}. The initial marking consists of one token in the initial place (p_{in}), with all other places being empty. A marked Petri net (N, M) is a workflow net if [vdA11]:

- N has an unique initial place p_{in}, with $\bullet p_{in} = \emptyset$ and N has an unique final place p_{fin}, with $p_{fin}\bullet = \emptyset$
- If a transition t^* is added to WN, connecting p_{in} and p_{fin}, WN is strongly connected. A graph is strongly connected if every vertex is reachable from every other vertex.

Soundness of a workflow net

In addition to the structural requirements, there are three soundness criteria for workflow nets [vdAvHtH+11]. Let $WN = (N, M_{in})$ with $N = (P, T, F)$ be a workflow net and let $R(WN) = \{M : \exists n \geq 0, t_1, ..., t_n \in T$ such that $M_{in}[t_1 > M_1 ... M_{n-1}[t_n > M_n = M\}$ be the set of reachable markings of WN.

- *Option to complete*: From every reachable marking $M \in R(WN)$, it is possible to reach a marking which marks the final place p_{fin}.

Proper completion: If the final place is marked it contains only one token and all other places are empty for a given case.

$M(p_{fin}) = 0 \Rightarrow M(p_{fin}) = 1$ and $M(p) = 0$ for $p \neq p_{fin}$. We refer to this marking as the final marking of WN. Figure 2.4 shows an example of a completed net meeting the criterion of proper completion.

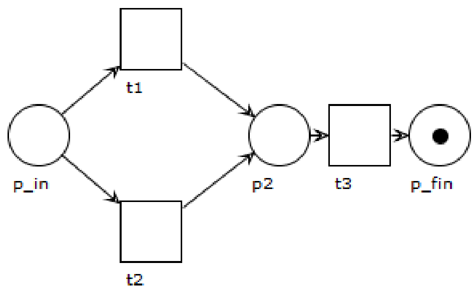


Figure 2.4: All places except the final place are empty and p_{fin} has one token

No useless transitions: it should be possible to execute any arbitrary transition by following an appropriate route through the WF-net. $\forall_{t \in T} \exists n > 0, t_1, .., t_n \in T$ such that $M_{in}[t_1 > M_1 ... [t_n > M_n$ and $M_n[t >$. Figure 2.5 shows an example of a net containing a dead transition.

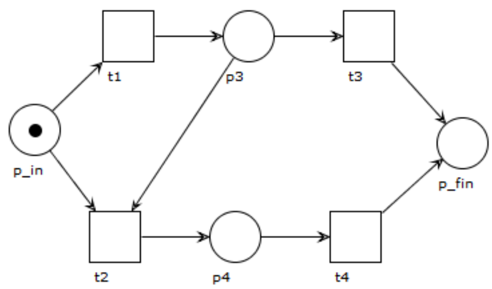


Figure 2.5: t2 is a useless transition

2.2.2 Deadlocks

A WF-net WN is deadlock free if for each reachable marking in which p_{fin} is empty, at least one transition is enabled. If WN is not deadlock free, there exists a reachable marking M, that is not final and for which there is no transition enabled at M and WN is not completed. This thesis is inspired by a deadlock in the settlement process.

Chapter 3

WoPeD

We created the workflow nets in the designing tool WoPeD. WoPeD also offers automatic analyses. The analyses we used in order to analyse our workflow nets will be explained in this chapter as well. The analyse criteria can be found in the Appendix.

3.1 About WoPeD

WoPeD (Workflow Petri Net Designer) is Java-based open-source software developed at the Cooperative State University Karlsruhe under the GNU Lesser General Public License (LGPL) [wop]. It is hosted on Sourceflog. Sourceflog is a distribution platform for open-source software. The development of WoPeD started in 2003. Since then the software has been altered continuously. WoPeD is well maintained and updated frequently. The latest version of WoPeD, version 3.6.0 has been released on 19-06-2017. The updates provided us with extra functionalities, which supported the execution of this thesis . WoPeD currently has 843 users and is mainly used for academic purposes [wop]. WoPeD provides a graphical editor for modelling Petri nets and workflow nets. PNML files can be imported and exported in WoPeD. WoPeD also exports image formats such as PNG and JPEG. Moreover, WoPeD contains several methods for analysing workflow nets. WoPeD has built-in algorithms to check the soundness, process metrics and quantitive simulation of a net [FS14]. Furthermore WoPeD allows the user to play the token game, meaning the user can simulate a firing sequence.

3.2 Used analysis methods

For analysing the workflow nets, simulating the settlement process, the semantical analysis is used. The semantical analysis consists of a wizard mode and an expert mode. Since the expert mode provides more information concerning the analysis of the net we use the expert mode. The criteria that the expert mode uses to analyse workflow nets can be found in Appendix A. When all the soundness criteria are met we consider a workflow net sound. For our research we do not deem it necessary that all the structural criteria are met (Appendix A).

Chapter 4

BPMN vs PN

For the conversion of BPMN diagrams to Petri nets the formal semantics of [DDO08] are used. The BPMN specification contains some ambiguities. The relevant ambiguities will be discussed in this chapter. Moreover we use two different translations of the message flow. These translations and the motivation behind these translations will be explained as well.

4.1 Formal semantics

In order to convert the BPMN collaboration diagram in Figure 5.1 we use the formal semantics proposed by [DDO08]. Table 4.1 shows the Petri net translation of the BPMN elements that are relevant for our case. The two different ways of translating the message flow are shown in Table 4.2 and 4.3. The paper [DDO08] states that the BPMN specification contains several ambiguities, therefore not every BPMN process or collaboration diagram conversion results in a workflow net. These formal semantics fail to convert a BPMN CD to a workflow net when a process model has multiple start events. This is not relevant for our case, since the collaboration diagram in [KGKR15] only has one starting event. The next ambiguity, on the other hand is relevant for our case, since it considers process instance completion. BPMN does not specify when an instance of a process model is completed [DDO08]. The approach described in [DDO08] considers a process instance to be completed when at least one end task has been executed and no other task for that process instance is enabled. The reason for this is that a workflow net should only have one final place, so a process diagram that is completed only when two end tasks are executed will not translate to a proper workflow net. Yet our case requires two end tasks to be executed in order for the collaboration diagram to be completed. We came up with a solution that allows both end tasks to be executed and still meets the workflow net properties. This solution is explained in Chapter 6. The other two ambiguities consider exception handling and OR-join gateways. Both flaws are irrelevant to our case.

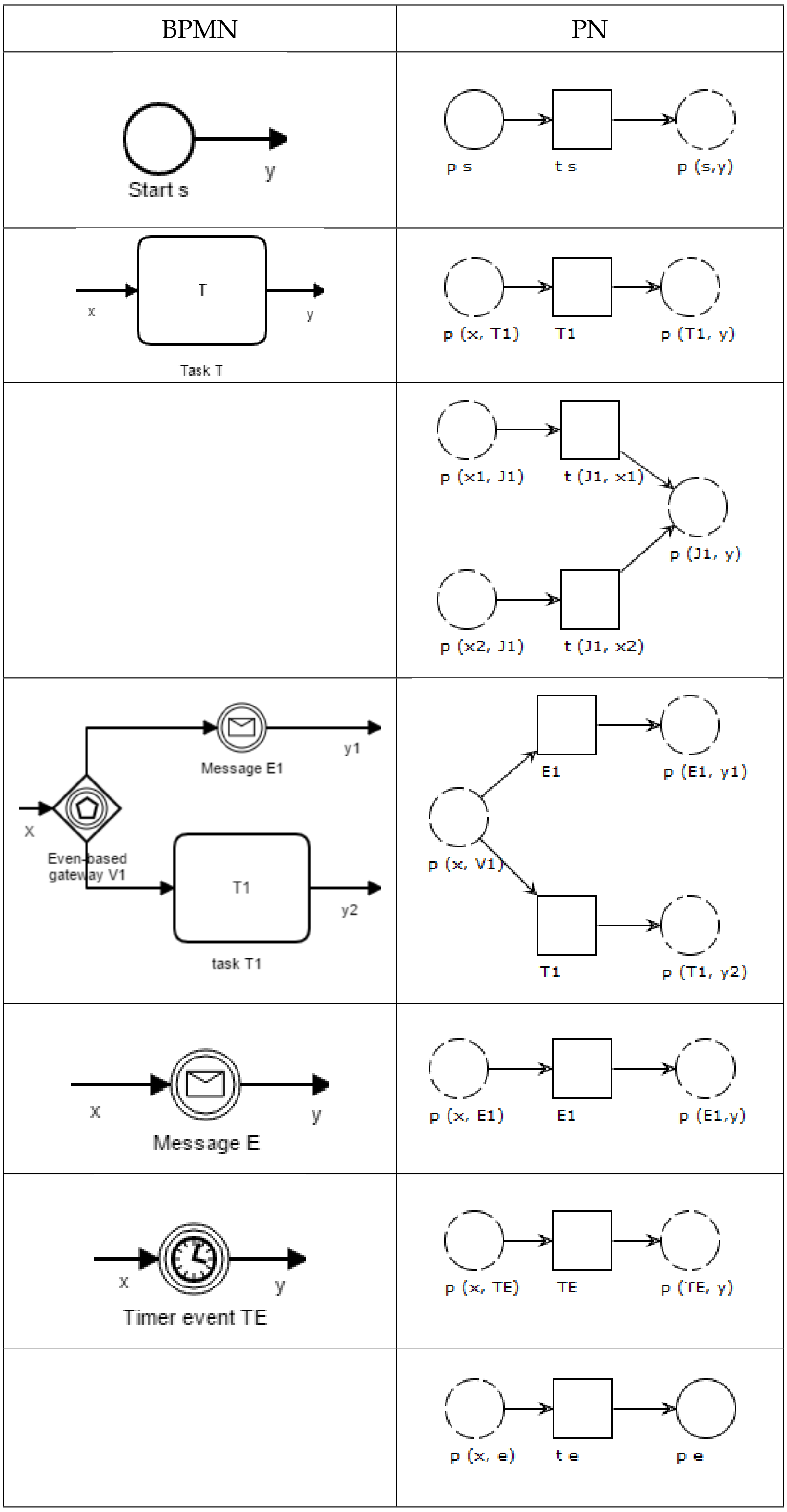


Table 4.1: BPMN elements converted to Petri nets using the formal semantics of [DDO08]

The dashed borders of some of the Petri net places mean that a place is not unique to a specific element. The sequence flow, connecting elements, is translated in arcs and places. It should be noted that the exclusive OR gateway, results in an non-exclusive OR in the Petri net conversion. No problem here, because the XOR-join in our case follows the split structure as created by the event-based gateway in Table 4.1.

The place corresponding to the start event is named "p $name_{start\_event}$". The place corresponding to the end event is named "p $name_{end\_event}$". The places with dashed borders of the start event, message event, end event and task are named according to the format "p($input, output$)". The input is defined as the name of the transition that provides the arc directed at the place. The output is defined as the name of the transition the arc runs towards. Their transitions are named after their corresponding BPMN element. For example, when a task is named $T1$ the corresponding transition is called $T1$ as well.

The naming of the PN elements representing the gateways differs from the naming described above. This is due to the fact that these gateways are not translated to one transition. Instead a special construction is created in order to mimic the behaviour of the gateways. In the case of the XOR join (J) the transitions are named according to the format $t(J, element_{input})$ The input places are named according to the format $p(input, J)$. The output place is named $p(J, output)$.

In case of the event-based gateway (V) the first place is named $p(input, V)$. The transitions are named after the BPMN elements connected to the gateway. The output places of the transitions are named according to the format $p(input, output)$.

In collaboration diagrams communication exchange is modelled by message flow. In case of the collaboration diagram we use in this thesis, there are two forms of message flow. The first message flow is sent by a task and received by a message start event and the second scenario is when the message is sent by a task and received by a message event, that is not a start event. For the latter, we came up with two ways of translating this to a Petri net.

In order for messages to be received directly we introduce the message flow conversion of Table 4.2. The message flow is translated by a single arc. The transition corresponding to the task sending the message is connected to the place leading to the transition corresponding to the message event, receiving the message. This enables the transition that symbolises the message event. However, this method makes it impossible to determine which action has enabled a transition t when the input place p has multiple input transitions t^0.

The message flow conversion from [DDO08] is shown in Table 4.3. The first row shows the translation of a task-to-start event. This is the case when a task execution in process A triggers the start event of process B. The message flow is translated by a single arc. The second row demonstrates the task-to-message conversion. The message flow is translated by adding a place between the two transitions that correspond to the task and message event. The tasks are connected to the place in the same direction as in the collaboration diagram. The disadvantage of this approach is that messages are not transferred directly. Messages can get stuck at the places that translate as message flow. The results of the different message flow conversions will be discussed in Chapter 6.

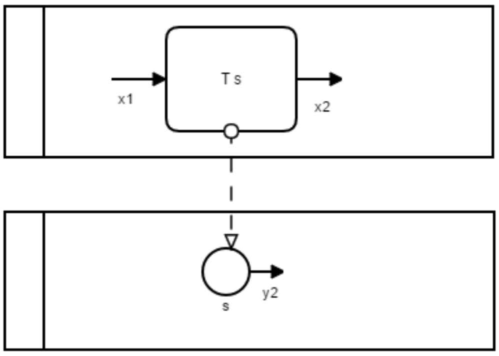
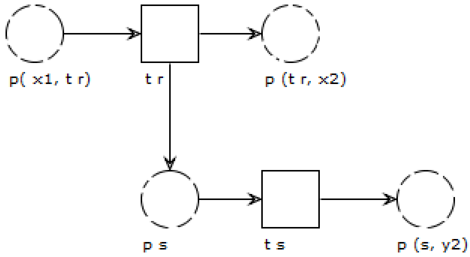
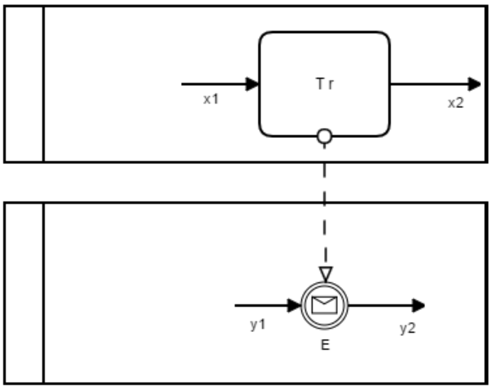
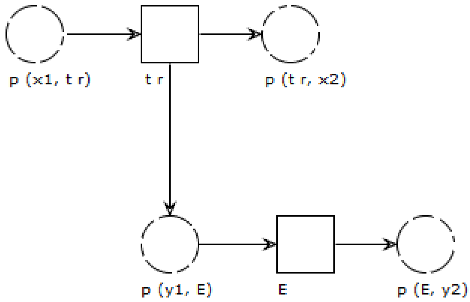
BPMN	PN	
Task-to-start		
Task-to-message		

Table 4.2: Message flow converted to Petri nets using our initial approach

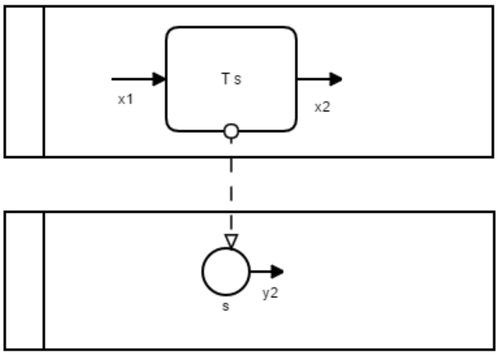
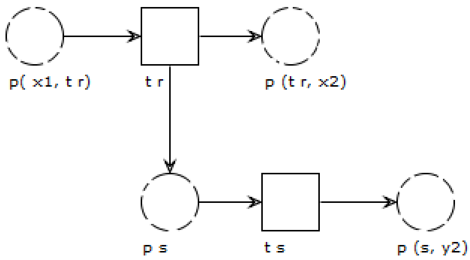
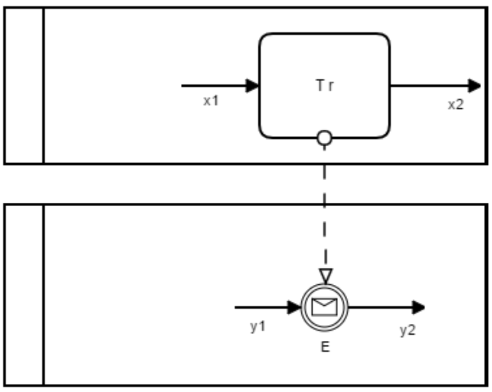
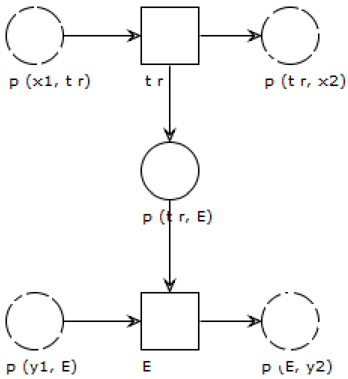
BPMN	PN	
Task-to-start		
Task-to-message		

Table 4.3: Message flow converted to Petri nets using the formal semantics of [DDO08]

Chapter 5

The case

In this chapter the case-study from [KGKR15] is described to illustrate some of the problems that might occur when modelling concurrent processes with BPMN Collaboration diagrams, which will be analyzed using Petri nets in Chapter 6. The case study describes a BPMN collaboration diagram of the settlement process. The settlement process is the final stage of the administrative process to complete a (securities) transaction executed on a secondary capital market. The first stage is the trading stage in which the conditions for exchange, such as price and quantity, are agreed between the trading parties. The second stage is the clearing stage in which the accountability for the exchange of funds and financial assets are determined. This might, for instance, involve the confirmation between the trading parties and the preparation of the details needed for creating settlement instructions. The settlement process is the third stage involving the actual exchange of funds and assets between the parties to the transaction. [GKM+02]. The safekeeping of securities is typically entrusted to a specialized financial institution: a Custodian. In the settlement process described in [KGKR15], three parties are involved: two investment firms, that have entered into a securities transaction, and the custodian entrusted with the safekeeping of these securities. To initiate settlement of the securities transaction, each investment firm sends an instruction, that is complementary to the other, to the custodian. When the custodian receives two matching instructions there is a legally binding commitment to transfer the securities, which can not be cancelled by the parties involved. The actual transfer is confirmed by the custodian to both investment firms.

The BPMN Collaboration Diagram from [KGKR15] shown in Figure 5.1. is a simplification of the settlement process as described above. First of all, only one investment firm is modelled. Secondly the matching of the two instructions sent by the investment firms is not explicitly modelled. Furthermore it is assumed that there are always two matching settlement instructions (SSI), but only one is modelled.

The diagram shows two pools. The first pool models the actions of the investment firm and the second pool models the actions of the custodian. The communication between these two parties is modelled with the message events, connected by the message flow.

In the collaboration diagram the process starts when the investment firm sends a settlement instruction (SSI) to the custodian. The custodian receives this instruction at the message event gateway ($S2$). After some time has passed ($TE2$) the custodian sends a settlement confirmation (SSC) to the investment firm. The intermediate time event ($TE2$) represents many in practice executed actions, including the matching. Once the investment firm has received the settlement confirmation (RSC) the end tasks ($E1$ and $E2$) of both processes can be executed. A settlement is reached.

The investment firm can also choose to send a cancellation (SC), after some time. This is modelled by the intermediate time event $TE1$. The cancellation is received by the custodian at RC. Upon receiving this message the custodian sends a cancellation confirmation (SCC). This cancellation confirmation is received by the investment firm at RCC. The end tasks of both processes can be executed. There is no collaboration completed without a settlement.

The collaboration diagram, however allows a third scenario, which results in a deadlock. In this scenario the investment firm sends a cancellation (SC) before receiving a message at RSC and the custodian sends a settlement confirmation SSC before receiving a message at RC. In this case a deadlock occurs, because the investment firm does not receive a message at RCC, so $E1$ cannot be executed. The custodian can execute its end event $E2$.

In the next chapter the collaboration diagram will be analysed using Petri nets. Five alternatives will be presented, after which the best solution will be chosen and discussed.

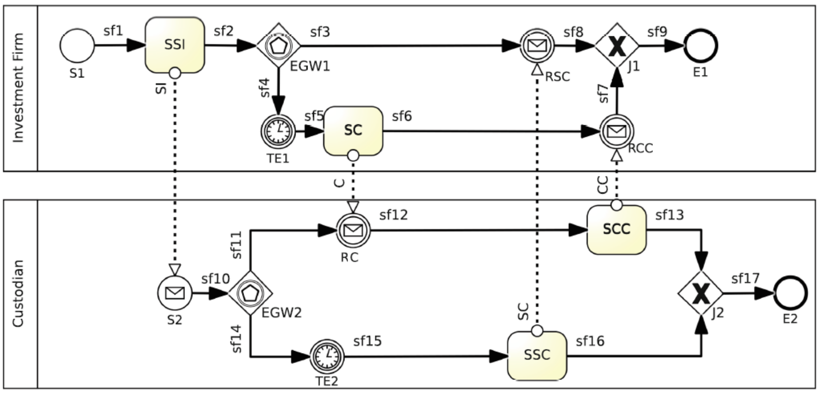


Figure 5.1: The collaboration diagram describing the settlement process [KGKR15]

Chapter 6

Conversion to Petri net

This chapter presents five Petri net conversions of the BPMN collaboration diagram in Figure 5.1. The BPMN elements are converted using the formal semantics proposed by [DDO08]. The first three Petri nets use the conversion of the message flow as explained in Table 4.2. The other two Petri nets use the message flow conversion from [DDO08] illustrated in Table 4.3. All nets will be analysed according to the same criteria. When for each method a solution was found that met all criteria we stopped creating new nets. The nets have been modelled and analysed using WoPeD (Section 3). Based on the analysis the best net is chosen.

6.1 Correctness criteria

A marked Petri net $WN = (N, M_{in})$ with $N = (P, T, F)$, has to meet the two workflow net properties and the three soundness criteria (from Section 2.2.1). The two workflow net properties are:

P.1 N has an unique initial place p_{in}, such that $\bullet p_{in} = ?$. N has an unique final place p_{fin}, such that $p_{fin}\bullet = ?$. Moreover $M_{in}(p_{in}) = 1$ and $M_{in}(p) = 0$ for all $p \neq p_{in}$

P.2 If a transition t^* is added to WN, connecting p_{in} and p_{fin}, WN is strongly connected.

A workflow net is sound if the three soundness criteria from Section 2.2.1 are met. The criteria are:

S.1 *Option to complete*: From every reachable marking $M \in R(WN)$, it is possible to reach a marking which marks the final place p_{fin}.

S.2 *Proper completion*: If the final place is marked it contains only one token and all other places are empty for a given case.

$M(p_{fin}) \neq 0 \implies M(p_{fin}) = 1$ and $M(p) = 0$ for $p \neq p_{fin}$. We refer to this marking as the final marking of WN.

S.3 *No useless transitions*: It should be possible to execute any arbitrary transition by following an appropriate route through the WF-net. $8_t\ 2\ T9M\ 2\ R(WN)$ such that $M[t >$.

Furthermore the workflow net has to meet the practical criteria. This means that the workflow net should only allow the scenarios that are allowed according to the settlement process. This means the investment firm can only cancel as long as the custodian has not received two matching instructions. In the collaboration diagram (from Figure 5.1) this happens at $TE2$. The other only allowed scenario is when the custodian has received two matching settlement instructions and then sends a settlement confirmation to the investment firm.

In order to convert these two scenarios to a workflow net, certain transitions have to be enabled at the right time. The scenarios have to be enforced in this way, because Petri nets do not have an option to incorporate timing.

Table 6.1 shows the transitions that have to be enabled when the investment firm makes the first choice whether to cancel or not. Table 6.2 shows which transitions have to be enabled to enforce the scenario where the custodian first decides whether matching has occurred or not.

Investment firm	Custodian	Allowed
RSC	TE2	No
RSC	RC	No
TE1	TE2	No
TE1	RC	Yes

Table 6.1: Allowed scenarios when the investment firm chooses first

Custodian	Investment firm	Allowed
TE2	RSC	Yes
TE2	TE1	No
RC	RSC	No
RC	TE1	No

Table 6.2: Allowed scenarios when the custodian chooses first

6.2 Process diagrams

In order to convert the BPMN collaboration diagram of Figure 5.1 we first converted the individual processes of the investment firm and the custodian to two separate workflow nets. The BPMN process diagrams of the investment firm and of the custodian can be found in Figure 6.1 and 6.2.

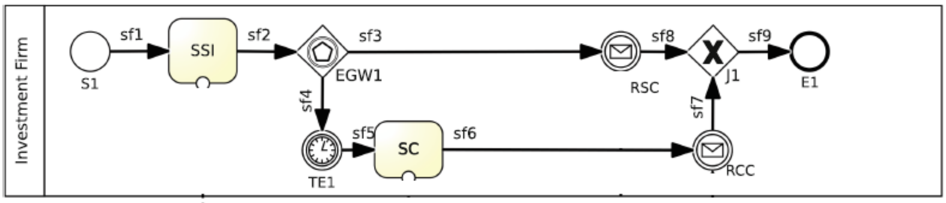


Figure 6.1: The BPMN process diagram of the investment firm [KGKR15]

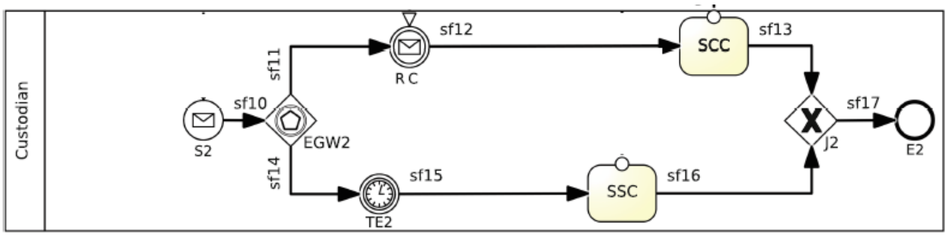


Figure 6.2: The BPMN process diagram of the custodian [KGKR15]

We convert these two process diagrams to workflow nets using the semantics of [DDO08], as explained in Chapter 4. The resulting workflow nets are shown in Figure 6.3 and 6.4. Analysis using WoPeD (Chapter 3) shows that the workflow nets are sound.

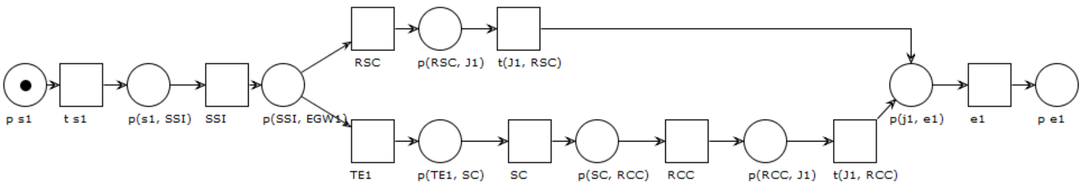


Figure 6.3: The process of the investment firm from 6.1 converted to a workflow net using the semantics of [DDO08]

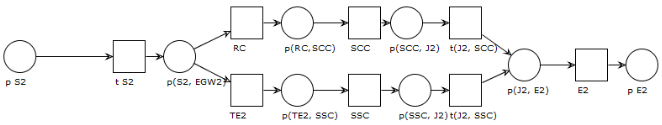


Figure 6.4: The process of the custodian from 6.2 converted to a workflow net using the semantics of [DDO08]

In order to investigate the effect of the message events we used two different approaches for converting the message flow. The first approach involves translating the message flow as a single arc between the transition that corresponds to the task sending the message and place before the transition that represents the receiving message event. The second approach used the translation as found in [DDO08]. The message flows were translated by adding a place between the transitions that correspond with the task and message event. The translation of the message flows is thoroughly explained in Chapter 4. The analysis and results of both methods will be discussed in this chapter.

6.3 First approach - first workflow net

The first workflow net using the first approach for the translation of the message events, connects both workflow nets as seen in Figure 6.3 and 6.4 by adding arcs between the transition that represents the task sending the message and the place before the task that represents the receiving message event. This method is demonstrated in Table 4.2. However, when we apply this translation strictly, the messages sent by SC and SSC have to connect to $p(S2, EGW2)$ and $p(SSI, EGW1)$, respectively. The big disadvantage of this is that these places are part of the construction mimicking the behaviour of the event based gateways. We predict this will at least cause problems, regarding the execution of the scenarios. Nevertheless we follow this approach as described in Chapter 4, in order to investigate whether our predictions were correct. All added arcs representing message flow are highlighted in Figure 6.5.

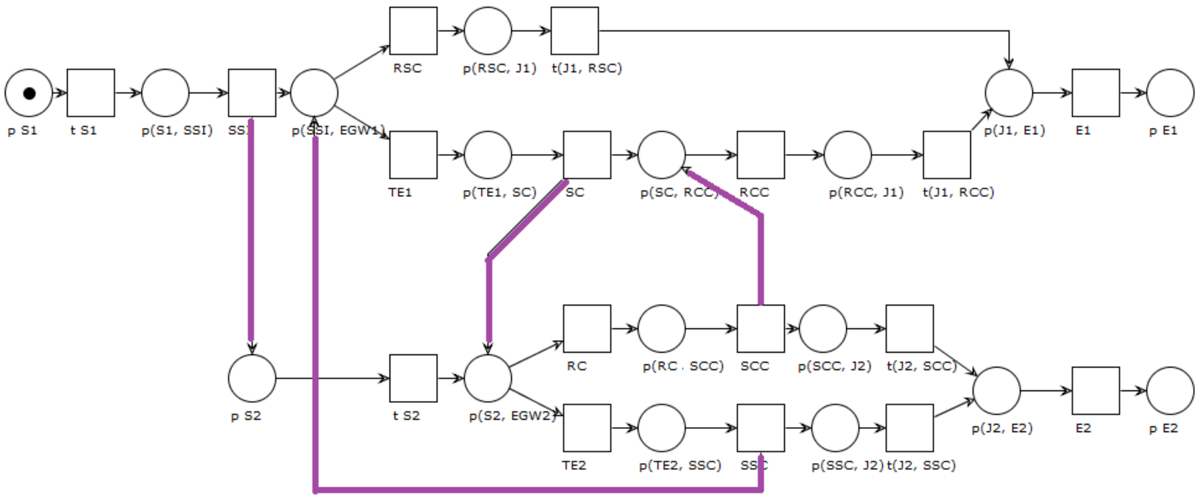


Figure 6.5: The workflow nets from Figure 6.3 and 6.4 combined by adding arcs according to the method found in Table 4.2

6.3.1 Analysis

The resulting net is not a workflow net. There are two final places, so workflow properties **P.1** and **P.2** are not met. Since the net is not a workfow net, it does not meet the soundness criteria **S.2** and **S.1** either. However all transitions are useful, so criteria **S.3** is met.

The biggest problem of this net is that it allows all possible scenarios to happen, not only the wanted scenarios as shown in Tables 6.1 and 6.2. Since the number of tokens in $p(SSI, EGW1)$ can flow arbitrarily large, the reachability graph is infinite. The net even allows multiple scenarios to be executed within one process cycle. This is due to the arcs connected to the places $p(SSI, EGW1)$ and $p(S2, EGW2)$, as we predicted. The translation for the message events is not suited for this CD.

For the next net we will prioritise meeting the workflow net properties and confirm that it is indeed best to eliminate the initial message flow and replace these with other constructions that provide interactions between

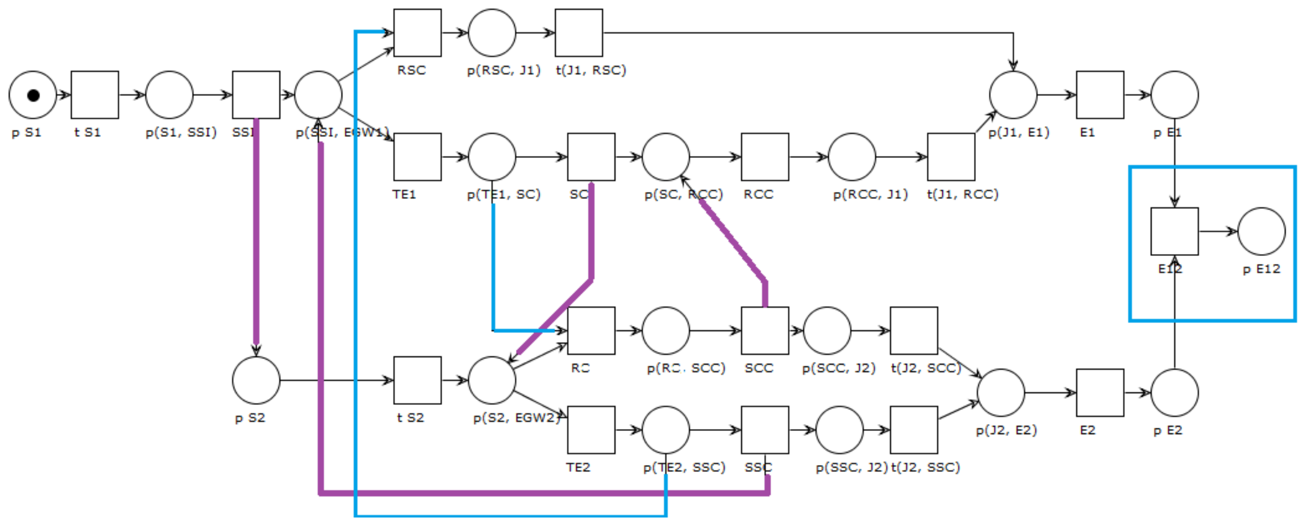


Figure 6.6: The improved version of the workflow net in Figure 6.5 satisfies the workflow net properties

the investment firm and custodian. We will not eliminate these in the second workflow net yet, since we want to investigate the effect of this translation on a net that meets the workflow net properties.

6.4 First approach - second workflow net

The second workflow net, shown in Figure 6.6 meets the workflow properties **P.1** and **P.2**. In order to satisfy these properties we have created a new final place. Since we consider a process instance to be completed when both end tasks are executed, both initial final places ($pE1$ and $pE2$) were connected to a new transition $tE12$ with output place $pE12$. Since transition $tE12$ is only enabled when both, $pE1$ and $pE2$, contain a token, it is ensured that both end tasks have been executed before the workflow net is completed. Moreover transition RSC is to be enabled once the place $p(TE2, SSC)$ contains a token. This means that the investment firm can only settle once the custodian has established a settlement. Moreover the transition RC is only enabled when the place $p(TE1, SC)$ contains a token. This restriction models the fact that the custodian can only send a cancellation confirmation, once the investment firm has sent a cancellation message. The initial message events and the additions to the previous net are highlighted.

6.4.1 Analysis

The workflow net from Figure 6.6 is an improvement compared to the previous net, since the workflow net properties are met. By using WoPeD, we observe that the net comes in a deadlock firing the sequence $pS1[tS1 > p(S1, SSI)[SSI > p(SSI, EGW1)pS2[tS2 > p(SSI, EGW1)p(S2, EGW2)[TE2 > p(SSI, EGW1)p(TE2, SSC)[RSC > p(RSC, J1)[t(J1, RSC) > p(J1, E1)[E1 > pE1$. All three soundness criteria are violated.

Even though we restricted the choices of both parties by enabling the transitions RSC and RC at the right time, the translation of the message events allows the net to execute another scenario without the restrictions the first cycle faces. This confirms our observation that the translation of the message events contradicts with the practical criteria of the settlement process. The next net will leave the first approach of message flow translation and construct a third workflow net without the initial message events.

6.5 Leaving the first approach - Third workflow net

The third workflow net as shown in Figure 6.7 uses the second workflow net as a base. As we concluded in the previous analysis, the initial translations of the message flow caused conflicts from a practical point of view. The first initial message event running from SSI to $pS2$ is maintained, since this initiates the process of the custodian. In order for the net to meet the soundness criteria we eliminate the deadlock. This is achieved by ensuring that the net can only execute firing sequences corresponding to the allowed scenarios. The allowed combinations of enabled transitions from Tables 6.1 and 6.2 were implemented. This was done by adding two extra places $p(SSI, TE)$ and $p(TE, RSC)$. These places enabled the transitions $TE1$, $TE2$, RSC and RC at the right time. The transition $TE1$ should only be enabled as long as transition $TE2$ has not fired yet, since this represents the delay passing and the custodian establishing a settlement. When $TE2$ has fired only transition RSC has to be enabled since the investment firm cannot settle anymore. Once the transition $TE1$ has fired, the delay of the investment firm has passed and the investment decides to cancel. In that case the transition $TE2$ should not be enabled anymore and only then the transition RSC_c for the custodian should be enabled, since the custodian cannot cancel before receiving a cancellation from the investment firm. In order for the net to function correctly an additional arc from $TE2$ to the final place $pE12$ is added. The additions in comparison to the previous nets have been highlighted.

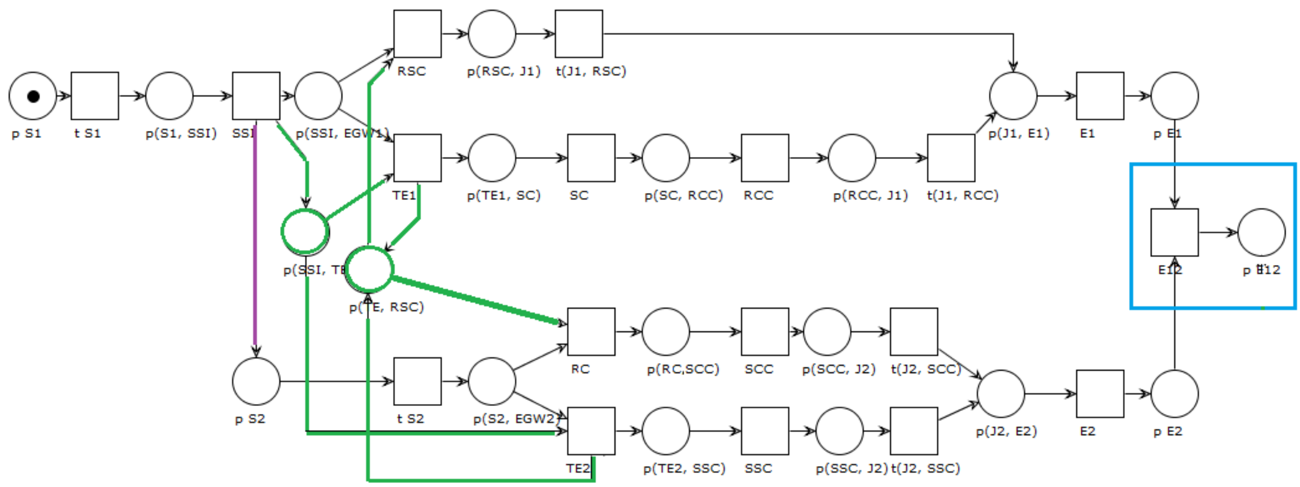


Figure 6.7: The improved version of the workflow net in Figure 6.6 meets all the correctness criteria

6.5.1 Analysis

The third workflow net still meets the workflow net properties. Moreover all three the soundness criteria as stated in Section 6.1 are met. The reachability graph (Figure 6.8) shows that there are no deadlocks and there is only one marking with a token in $pE12$. Also, only the allowed firing sequences can be executed. The entire net is empty and an allowed scenario has been executed. The initial message events, except for the one triggering the process of the custodian, do not provide any extra information. Moreover we observed that the reachability graph of the third net is no longer infinite in comparison to the reachability graphs of its predecessors. This indicates a reduction of redundancy and a reduction in unwanted firing executions. The only problem with this net is its structure. Due to the elimination of the message flow, the structure of the settlement process, as modelled in the CD, is lost.

Nodes in reachability graph	Places in workflow net
p6	p(SC, RCC)
p14	p(SCC, J2)
p16	p(SSC, J2)
p21	p(SSI, TE)
p23	p(TE, RSC)
p2	p(TE1, SC)
p13	p(RSC, SCC)
p12	p(S2, EGW2)
p15	p(TE2, SSC)
p3	p(SSI, EGW1)
p10	pE1
p18	pE2
p7	pS1
p11	pS2
p20	pE12
p9	p(J1, E1)
p17	p(J2, E2)
p8	p(RCC, J1)
p5	p(RSC, J1)
p4	p(S1, SSI)

Table 6.3: The name of the places in the workflow net in Figure 6.7 corresponding to the name of the nodes in the reachability graph in Figure 6.8

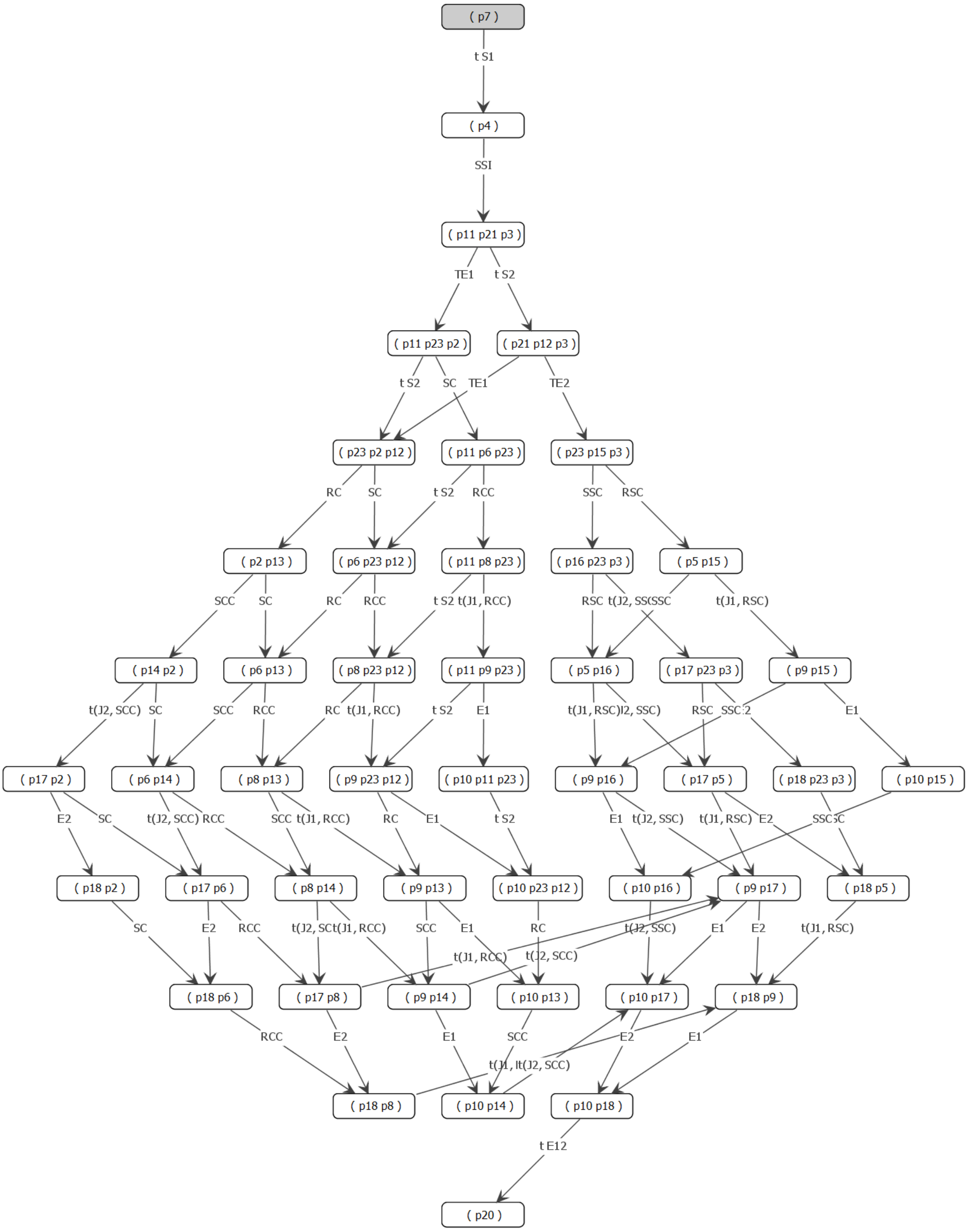


Figure 6.8: The reachability graph of the net in Figure 6.7

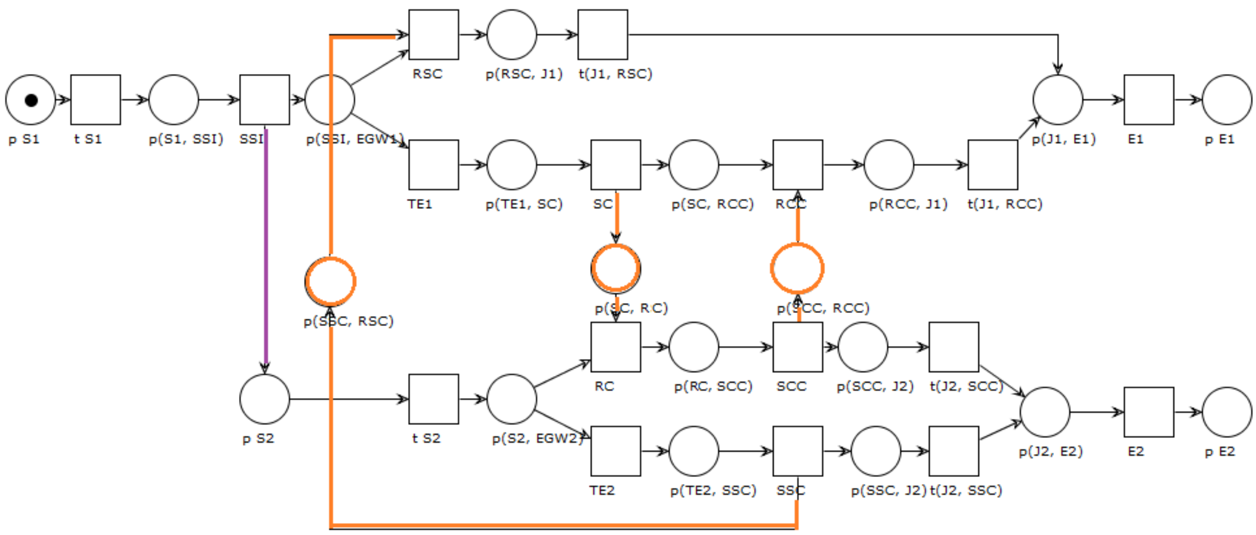


Figure 6.9: The combined workflow net using the semantics of [DDO08]

6.6 Second approach - fourth workflow net

For the first workflow net that uses the second approach the nets of Figure 6.3 and 6.4 are connected by using the translation of the message events of [DDO08], shown in Table 4.3. The first message flow (SI) was translated using the task-to-start-event translation (Table 4.3). This results in the arc between SSI and $pS2$. The three other message flows (C, SC, CC) were translated using the task-to-message translation (Table 4.3), leading to the additional places $p(SC, RSC)$, $p(SCC, RCC)$ and $p(SSC, RSC)$. The task-to-start message flow and the task-to-message flow are highlighted. The resulting net can be found in 6.9.

6.6.1 Analysis

The net does not meet the workflow net properties **P.1** and **P.2**, since there are two final places. The net also does not meet soundness criteria **S.1** and **S.3** since there is a deadlock when the custodian chooses to establishes a matching ($TE2$) and the investment firm decides to cancel ($TE1$). This deadlock also exists in the BPMN collaboration diagram. Moreover the allowed scenarios can be executed, but so can other forbidden scenarios that result in a deadlock. The deadlock can be observed in the reachability graph in Figure 6.10, when the firing sequence leading to the most left final marking is followed. One of the forbidden firing sequences is:

$pS1[tS1 > p(S1, SSI)[SSI > p(SSI, EGW1)pS2[tS2 > p(SSI, EGW1)p(S2, EGW2)[TE2 > p(SSI, EGW1)$
$p(TE2, SSC)[SSC > p(SSC, RSC)p(SSC, J2)p(SSI, EGW1)[TE1 > p(SSC, RSC)p(SSC, J2)p(TE1, SC)[SC >$
$p(SSC, RSC)p(SSC, J2)p(SC, RC)p(SC, RCC)$.

Immediately we see that the reachability graph is smaller than the reachability graphs of the first two nets, implying that the second translation method of the message flow enforces more restrictions than the first translation.

Nodes in reachability graph	Places in workflow net
p10	pE1
p18	pE2
p7	pS1
p11	pS2
p9	p(J1, E1)
p17	p(J2, E2)
p8	p(RCC, J1)
p5	p(RSC, J1)
p4	p(S1, SSI)
p6	p(SC, RCC)
p19	p(SC, RSC)
p14	p(SCC, J2)
p16	p(SSC, J2)
p2	p(TE1, SC)
p13	p(RSC, SCC)
p12	p(S2, EGW2)
p20	p(SCC, RCC)
p21	p(SSC, RSC)
p15	p(TE2, SSC)
p3	p(SSI, EGW1)

Table 6.4: The name of the places in the workflow net in Figure 6.9 corresponding to the name of the nodes in the reachability graph in Figure 6.10

6.7 Second approach - fifth workflow net

This workflow net uses the workflow net from Section 6.6 as a base. Since using the semantics as described in [DDO08] did not result in a workflow net corresponding to the workflow net properties **P.1** and **P.2** we first ensure that these properties are met. In order to meet property **P.1** we create one final place. This is done by connecting the two end places ($pE1$ and $pE2$), to one transition, which is input to final place $pE12$, as we did before (Section 6.6). Due to the proper construction property **P.2** is now met as well. In order to tackle the soundness issues we ensure that the scenario in which the custodian settles and the investment firm cancels can not be executed, which was in correspondence with the practical criteria. In order to solve this issue, transition $TE1$ may be enabled only as long as the custodian has not confirmed the settlement yet. Therefore place $p(S2, TE2)$ is added. Transition $tS2$ fires a token towards $p(S2, TE2)$, which is input place to both transitions $TE1$ and $TE2$. Only one of these transitions can consume the token in $p(S2, TE2)$. Therefore, the conflicting scenario that results in a deadlock cannot be executed anymore. The addition of the place

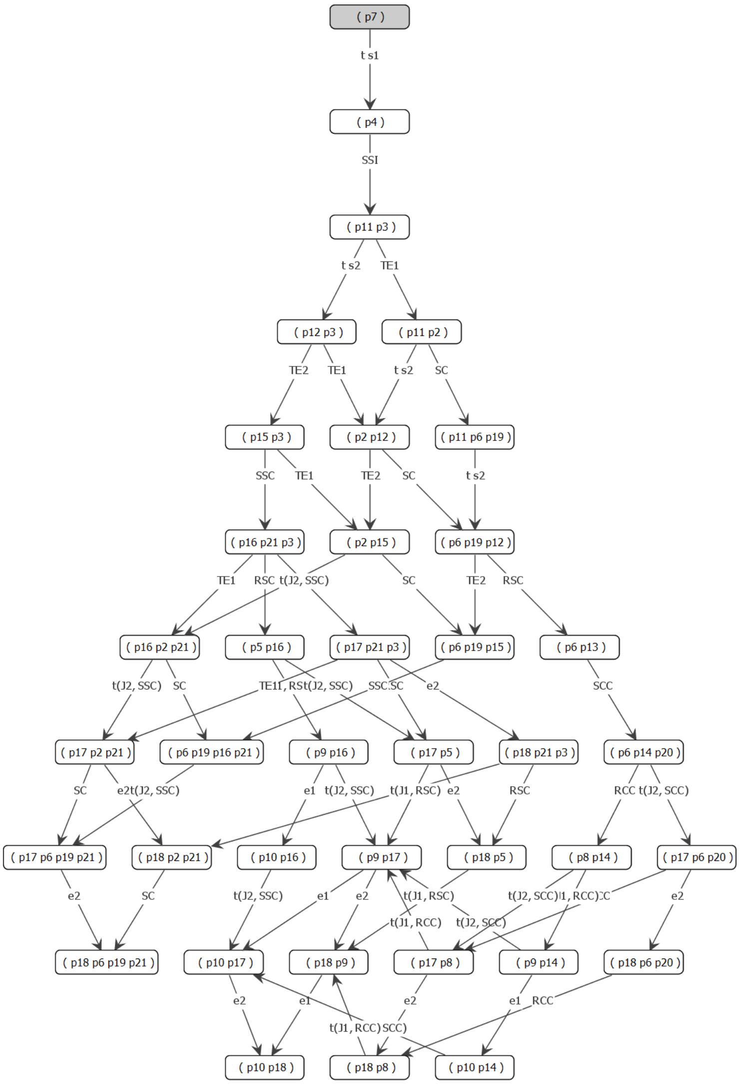


Figure 6.10: The reachability graph of the net in Figure 6.9

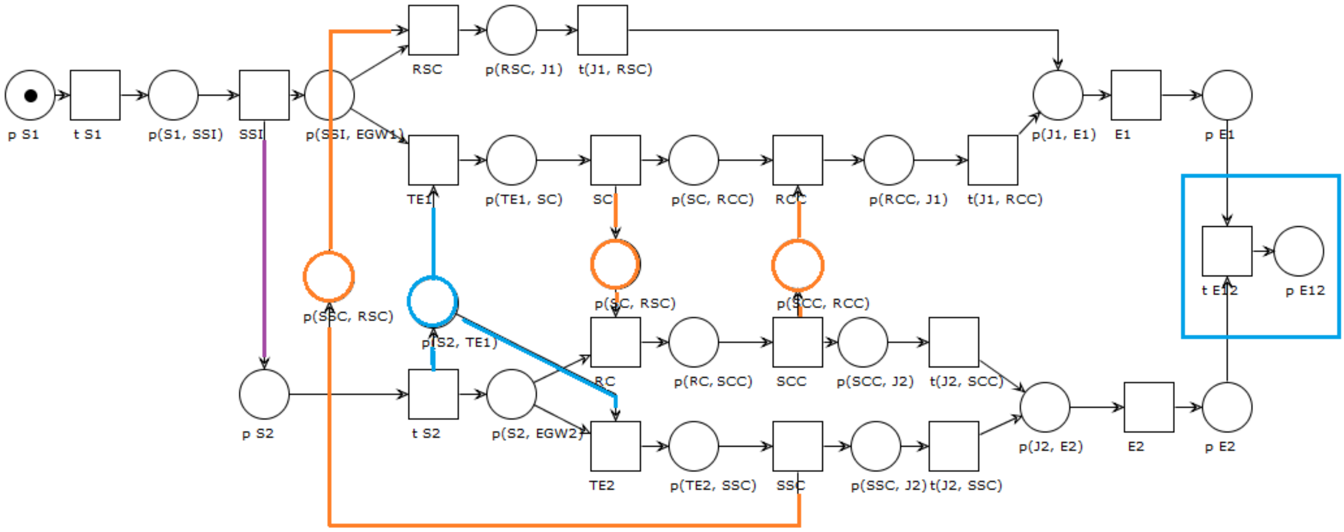


Figure 6.11: Improved version of the net in Figure 6.9 with new final place and additional place $p(S2, TE2)$

$p(S2, TE2)$ causes the net to only allow the scenarios as allowed by the settlement process. The net is shown in Figure 6.11. The addition in comparison to Figure 6.11 are highlighted.

6.7.1 Analysis

After the changes as described in Section 6.7, the workflow net meets all the workflow net properties and soundness criteria. The added place $p(S2, TE1)$ enables the allowed combination of transitions from Table 6.1 and 6.2. This causes the workflow net to only execute all allowed scenarios. The only issue that arises in this solution is the fact that the additional place $p(S2, TE2)$ does not symbolise any action within the settlement process. The reachability graph in Figure 6.12 shows that all reachable markings end with one token in the final place.

Nodes in reachability graph	Places in workflow net
p10	pE1
p18	pE2
p7	pS1
p11	pS2
p24	pE12
p9	p(J1, E1)
p17	p(J2, E2)
p8	p(RCC, J1)
p5	p(RSC, J1)
p4	p(S1, SSI)
p1	p(S2, TE1)
p6	p(SC, RCC)
p19	p(SC, RSC)
p14	p(SCC, J2)
p16	p(SSC, J2)
p2	p(TE1, SC)
p13	p(RSC, SCC)
p12	p(S2, EGW2)
p20	p(SCC, RCC)
p21	p(SSC, RSC)
p15	p(TE2, SSC)
p3	p(SSI, EGW1)

Table 6.5: The name of the places in the workflow net in Figure 6.11 corresponding to the name of the nodes in the reachability graph in Figure 6.12

6.7.2 Discussion

Eventually both methods for converting the message flow, resulted in Petri net solutions for the deadlock within the settlement process. Eventually the best solution turns out to be the fifth net (Figure 6.11). However third (Figure 6.7) net also corresponds to the outcome of the settlement process, while reducing redundancy. Nevertheless we prefer the fifth workflow net (6.11), due to the fact that it requires less additions from the original net in Figure 6.5 and the structure of the net corresponds more to the structure of the BPMN CD. The optimisation of the net resulting from the second method was relatively straight-forward due to the learning process we experienced when optimising the net resulting from the first approach. When we study the reachability graphs of the best solutions (Figure 6.8 and 6.12) we found that the second method has a smaller reachability graph, while the outcome is the same.

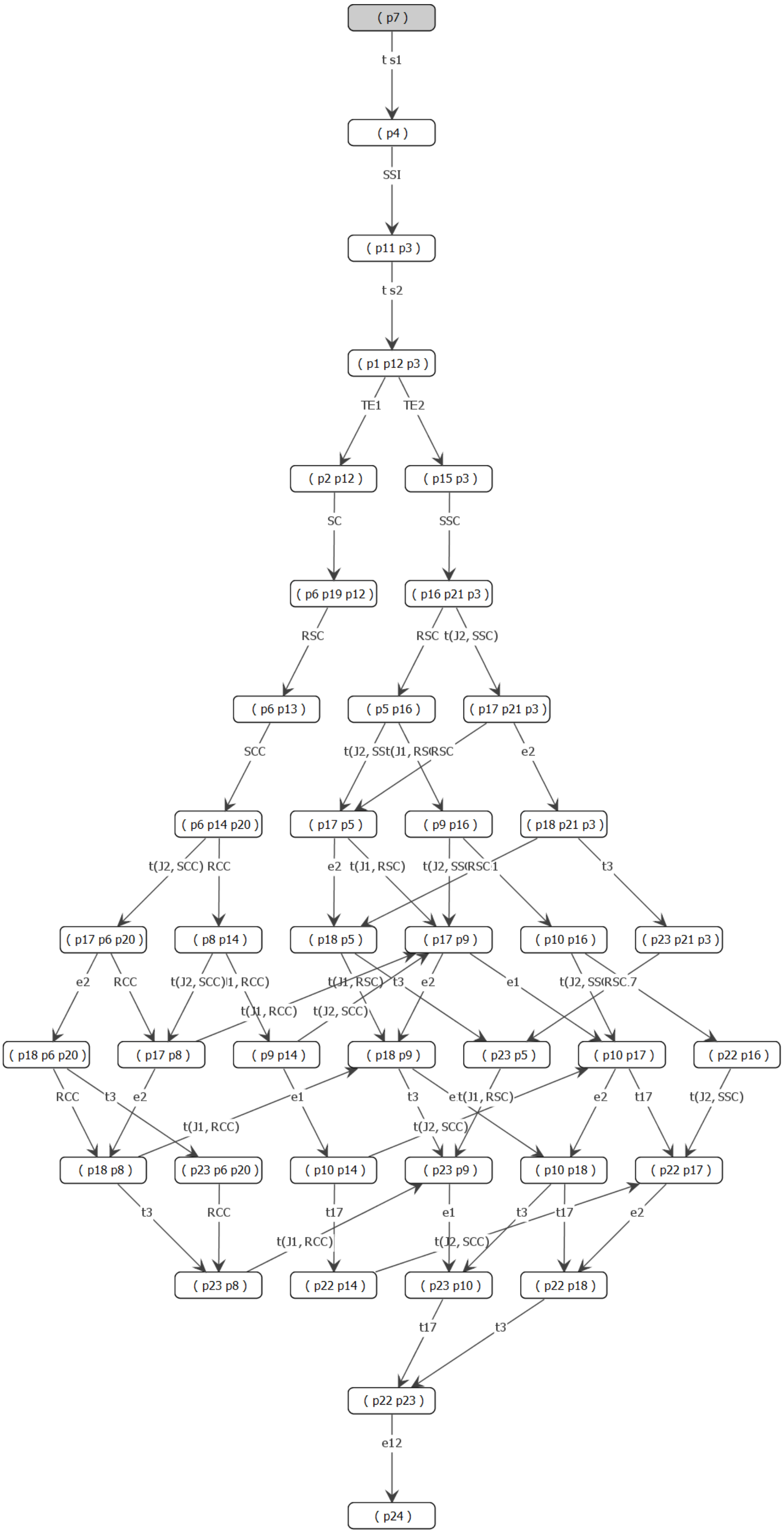


Figure 6.12: The reachability graph of the net in Figure 6.11

Since both nets have the same end results and meet the same criteria. However there is a difference in efficiency. The Petri net solution to the deadlock within the settlement process consists of enabling the transitions $TE1$, $TE2$, RSC and RC at the correct time. This solution corresponds to the concept of mutual exclusion. Mutual exclusion constraints are a way of timing the concurrent use of resources. In Petri nets a mutual conclusion constraint can be defined as a condition that limits a weighted sum of tokens contained in a subset of places [GDS92]. These places are called permission places, granting the transitions permission to be executed. In the case of our best solution (Figure 6.11) there is only one token in one involved in the permission place $p(S2, TE2)$. The permission place allows either the custodian to settle or the investment firm to cancel. In the net in Figure 6.7 the permission places are $p(SSI, TE)$ and $p(TE, RSC)$, granting either $TE1$ or $TE2$ permission or RSC or RC permission. In this case there is a double mutual exclusion. In BPMN there is not a direct way to restrict the concurrency between processes in a collaboration [RvdAtH16]. In practice these solutions guarantee that the two delays should not pass at the same time. The solution found in this thesis does not propose a practical solution for the issue within the settlement process, since the timer events do not really only represent an amount of time, but symbolise several other processes.

Future work could involve converting the Petri net solution to a BPMN collaboration diagram by implementing the mutual exclusion solution in an ad-hoc way. This conversion, however lays beyond the scope of this research. A workflow simulator that supports mutual exclusion is COSA [COS].

Chapter 7

Conclusion

This research provides more insight in the complexity of the coordination of the communication between processes. Different approaches for tackling the problem within the settlement process are executed, by constructing five different WF-net interpretations of the BPMN collaboration that models this process.

We thus arrived at the conclusion that a collaboration diagram with multiple end events, requires an adjusted construction in order to convert to a workflow net. Otherwise the resulting net has more than one final place. Although we kept to the traditional definition of soundness in this thesis, a workflow net can be analysed, using different criteria. In [vdAvHtH$^+$11] many adjusted notations of soundness for workflow nets are defined and analysed. It is also stated that different formal semantics can be used in order to analyse a BPMN diagrams, for example process algebras such as CSP.

In order of appearance each of the workflow nets met more criteria than the previous net for the method of translating the message flow. This allowed us to elaborate on the existing solution, leading us to our final solutions. The final solutions meet the full set of criteria. Nonetheless we prefer the solution that uses the message flow translation of [DDO08].

When observing both final solutions, we found that the deadlock can be solved by mutual exclusion [GDS92], no matter which method for translating message flow is used. One of the solutions uses two permission places and the other solution uses only one permission place. This finding points out the essence of the problem. Mutual exclusion of activities in different pools in a CD is not supported in BPMN. Therefore the issue within the settlement process might not be solvable in BPMN CD with mutual exclusion. However BPMN choreography diagrams could provide a solution, using mutual exclusion since these diagrams can capture more detailed communication.

Since the settlement process is such a complex process our solution will probably not provide an actual practical solution for the problem. Nor did we find a generic method to check the soundness for a BPMN CD because the formal semantics that we used for the conversion of BPMN CD to a workflow net do not always result in a workflow net, as we discovered during the conversion of the BPMN CD of the settlement process. In order to convert the BPMN CD given in [KGKR15], to a workflow net, domain specific knowledge regarding the completion of the settlement process was required.

The case used for this thesis is limited to actions of one investment firm. This study can be expanded by studying the settlement process from the perspective of the custodian and by adding a second investment firm. Furthermore, BPMN is a popular modelling notation and it would be easier to create a generic way of converting BPMN CD to Petri nets if BPMN was better specified, since ambiguities within BPMN cause confusion. The further development of formal semantics for BPMN is relevant.

Bibliography

[Bro14] Richard G. Brown. A simple explanation of how shares move around the securities settlement system. 2014.

[COS] *COSA website in July 2017*. http://www.cosa.nl/simulation.html.

[DDO08] Remco M. Dijkman, Marlon Dumas, and Chun Ouyang. Semantics and analysis of business process models in BPMN. *Information & Software Technology*, 50(12):1281–1294, 2008.

[FS14] Thomas Freytag and Martin Sänger. Woped - an educational tool for workflow nets. In *Proceedings of the BPM Demo Sessions 2014 Co-located with the 12th International Conference on Business Process Management (BPM 2014), Eindhoven, The Netherlands, September 10, 2014.*, page 31, 2014.

[GDS92] Alessandro Giua, Frank DiCesare, and Manuel Silva. Generalized mutual exclusion constraints on nets with uncontrollable transitions. *Systems, Man and Cybernetics*, page 975, 1992.

[KGKR15] Pieter M. Kwantes, Pieter Van Gorp, Jetty Kleijn, and Arend Rensink. Towards compliance verification between global and local process models. In *Graph Transformation - 8th International Conference, ICGT 2015, Held as Part of STAF 2015, L'Aquila, Italy, July 21-23, 2015. Proceedings*, pages 221–236, 2015.

[RIRG06] Jan Recker, Marta Indulska, Michael Rosemann, and Peter Green. HOW GOOD IS BPMN REALLY? INSIGHTS FROM THEORY AND PRACTICE. 2006.

[RPU+07] Ivo Raedts, Marija Petkovic, Yaroslav S. Usenko, Jan Martijn E. M. van der Werf, Jan Friso Groote, and Lou J. Somers. Transformation of BPMN models for behaviour analysis. In *Modelling, Simulation, Verification and Validation of Enterprise Information Systems, Proceedings of the 5th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems, MSVVEIS-2007, In conjunction with ICEIS 2007, Funchal, Madeira, Portugal, June 2007*, pages 126–137, 2007.

[RvdAtH16] Nick Russell, Wil M.P. van der Aalst, and Arthur H. M. ter Hofstede. *Workflow Patterns: The Definitive Guide*. MIT Press, 2016.

[vdA11] Wil M. P. van der Aalst. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.

[vdAvHtH+11] Wil M. P. van der Aalst, Kees M. van Hee, Arthur H. M. ter Hofstede, Natalia Sidorova, H. M. W. Verbeek, Marc Voorhoeve, and Moe Thandar Wynn. Soundness of workflow nets: classification, decidability, and analysis. *Formal Asp. Comput.*, 23(3):333–363, 2011.

[vRSvS15] Mark von Rosing, August-Wilhelm Scheer, and Henrik von Scheel. *The Complete Business Process Handbook*. 2015.

[Whi04] Stephan A. White. Introduction to BPMN, 2004.

[wop] *WoPeD website in March 2017*. http://woped.dhbw-karlsruhe.de/woped/.

Appendices

Appendix A

The expert mode provides a structural analysis and a soundness analysis. For this research we want our nets to pass at least the soundness analysis. The soundness criteria specified in WoPeD are:

- Workflow net properties
  - One source place
  - One sink place
  - Arcs with non-standard arc weight: 0
  - Transitions with empty preset: 0
  - Connected components: 1
  - Strongly connected components: 1
- Initial marking
  - Wrongly marked places: 0
- Boundedness
  - Unbounded places: 0
- Liveness
  - Dead transitions: 0
  - Non-live transitions: 0

When all of these criteria are met the net is considered sound. It should be noted that WoPeD includes the workflow net properties within the definition of soundness, while we distinguished the workflow net properties from the soundness criteria in Chapter 2. For our analysis in this research we will refer to the soundness without excluding the workflow net properties. These properties will be mentioned separately when we analyse the nets in Chapter 6. Even though the soundness criteria are defined by concepts such as boundedness and liveness, they correspond to the criteria as stated in Section 2.2.1.

The criteria of the structural analysis are as following:

- The net statistics
- The amount of wrongly used operators
- The amount of free-choice violations
- The amount of S-components
- The wellstructuredness of the net

However, we do not lay focus on these criteria and will not prioritise meeting these.

The latest version of WoPeD also automatically constructs the and reachability graph of a WF-net and optimizes the lay-out of the WF-net.