Overview on Decentralized Establishment of Consistent Multi-lateral Collaborations based on Asynchronous Communication

Andreas Wombacher
University of Twente,
7500 AE Enschede, The Netherlands
a.wombacher@utwente.nl

Peter Fankhauser
Fraunhofer Gesellschaft IPSI,
64293 Darmstadt, Germany
peter.fankhauser@ipsi.fraunhofer.de

Karl Aberer
School of Computer and Communication Sciences, EPFL,
1015 Lausanne, Switzerland
karl.aberer@epfl.ch

Abstract

Service oriented architectures facilitate loosely coupled collaborations, which are established in a decentralized way. One challenge for such collaborations is to guarantee consistency, that is, fulfillment of all constraints of individual services and deadlock-freeness. This paper introduces a decentralized approach to consistency checking, which utilizes only bilateral views of the collaboration.

1. Introduction

Service Oriented Architectures (SOA) facilitate loosely coupled collaborations between companies (B2B Commerce) as well as between company divisions (Enterprise Application Integration, EAI). The main benefits of SOAs are dynamicity and flexibility: The individual services of companies can dynamically evolve and be flexibly combined to form a multi-lateral collaboration without the straight jacket of a preorchestrated global workflow.

However, these benefits only hold, if the resulting collaborations are consistent, that is, if the constraints of the individual services are fulfilled and no deadlocks can occur. To check this consistency v.d.Aalst [12] has introduced a centralized approach based on an explicit global workflow constructed from the individual local workflows based on asynchronous communication. As stated above, the flexibility of SOAs requires a more dynamic approach without instantiating a global workflow. This paper introduces a decentralized approach to consistency checking, which utilizes only bilateral views of the (implicit) global workflow. Thereby, loosely coupled collaborations can enjoy the same consistency guarantees as collaborations controlled by a global workflow.

The remainder of this paper is organized as follows. In Section 2, we introduce by way of example Workflow Nets, which model individual workflows and their constraints. In Section 3, we briefly review consistency checking for multi-lateral collaborations on the basis of a global workflow. In Section 4, we describe how to represent the multi-lateral collaboration by means of bilateral views, and how these views can be used for decentralized consistency checking. In Section 5, we discuss related work, and in Section 6 we discuss future work.

2. Example

The example used throughout this paper is a simple procurement workflow within a virtual enterprise comprising a buyer, an accounting department, and a logistics department (Figure 1, see also [14]). Accounting checks orders (order message) of buyers and forwards them to logistics (deliver message) to deliver the requested goods. Logistics confirms the receipt (deliver_conf message), which is forwarded by accounting to the buyer (delivery message). Further, the buyer may perform parcel tracking (get_status and status messages) as sometimes offered by logistics companies, after accounting has authenticated the buyer (auth message).

Figure 1 represents the global relationships between the individual services, but not the underlying local workflows. In this paper we use Workflow Nets (WF-Net) [11, 12] to model these workflows. However, our approach could also be applied to other models like for example, Petri Nets [5],

* This work has been done during the employment at Fraunhofer IPSI.
or statecharts [2]. A Workflow Net (WF-Net) consists of places (circles) representing business tasks and transitions (rectangles) connecting places representing a message exchange. The transitions are labeled with $\text{msg}$ representing sender $s$ and receiver $r$ of the message $msg$. Messages may in addition contain parameters annotated in brackets. WF-Nets contain a single final place represented by a circle with a solid line. In contrast to general Petri Nets WF-Nets require a single initial and final place.

The execution of a workflow is realized by pushing tokens through the WF-Net, which are depicted as a dot within a place. A transition is enabled if all input places of a transition contain a token. If a transition is enabled, it may fire, which removes tokens from incoming places and inserts new tokens to all outgoing places of the transition. The current distribution of tokens over the places represents the state of the workflow and is called marking. The firing of a transition can be further constrained by a logical formula on the parameters of the message, which is noted in square brackets. \footnote{These constraints are also known as colour extension (see e.g. [5]).}

The local workflows of the parties involved are depicted in Figure 2. The process is started by buyer $B$ sending a $\text{B\#Order}$ message to accounting $A$ with the parameters item number $it$, price $p$, and amount $a$, which is restricted to be below 100. Accounting $A$ informs logistics $L$ via a $\text{A\#L\#Deliver}$ message to deliver the ordered goods without forwarding the price parameter $p$ of the order. $L$ accepts the request from $A$ if the amount $a$ is below 100, and confirms it with a $\text{L\#A\#Deliver\_Conf}$ message providing an additional tracking number $tn$ parameter. $A$ forwards the delivery details of the order ($\text{A\#B\#Deliver}$ message) to $B$. Afterwards, $B$ can track parcels directly with $L$ by sending a $\text{B\#L\#Get\_Status}$ message containing a tracking number parameter $tn$ answered by a $\text{L\#B\#Get\_Status}$ message with an additional status parameter $st$. While $B$ must have performed the $\text{A\#B\#Deliver}$ message before tracking parcels $L$ allows parcel tracking at any time after receiving an authentication message from $A$ ($\text{A\#L\#Auth}$ message). Finally, $B$ terminates the buyer and logistics process by sending a $\text{B\#L\#Terminate}$ message.

### 3. Centralized Workflow Consistency

The global workflow for these local workflows can be constructed as described in [12] by relating two transitions with the same message and a matching sender and receiver pair via an asynchronous channel. Each channel is represented by an additional place connected by an incoming arc with the "sending" transition of a message and an outgoing arc to the "receiving" transition. A token located in a newly introduced place can be interpreted as a message contained in the channel waiting for being received by the corresponding party. Finally, new initial and final places are connected by new transitions labeled $t_{initial}$ and $t_{final}$ to the local initial and final places.

Applying this approach to the example workflow in Figure 2 leads to the global workflow in Figure 3. Note, that one parcel tracking option of the logistics department has been discarded because no corresponding sender transition labeled $\text{A\#L\#Auth}$ exists at the accounting workflow, thus it is never used.

Such a global workflow can be analyzed for consistency by constructing the so called occurrence graph. The vertices of the occurrence graph represent all possible markings of a WF-Net, and the directed edges represent the transitions leading from one marking to the next. The WF-Net is consistent, if all vertices in its occurrence graph have either at least one outgoing transition or are a final marking. On this basis, it can be shown that this global workflow is consistent, that is, it fulfills all local constraints on parameters and
is deadlock free. However, the decision on the consistency of the multi-lateral collaboration is centralized based on the constructed global workflow. As a consequence, at least one party needs to know the local workflows of all parties involved, which is too restrictive for SOAs.

4. Decentralized Workflow Consistency

In contrast to the centralized consistency checking based on a global workflow, a decentralized consistency checking must be entirely based on each party’s local knowledge: the own local workflow and the bilateral communication of the party with its partners. For example the buyer should only need to know about its local workflow and its individual interaction with logistics and accounting, but not about the entire local workflows of logistics and accounting and not about their possible interaction.

In the following we will first show how to construct a WF-Net representing the bilateral communication between two local workflows, resolve cycles in these WF-Nets, and then show how consistency of the multi-lateral collaboration can be decided on the basis of bilateral acyclic WF-Nets.

4.1. Bilateral WF-Nets

A bilateral WF-Net is a workflow representing the bilateral communication between two trading partners. This can be constructed in two steps: first both local workflows are abstracted by substituting all transitions that do not directly involve the two trading partners with silent transitions $\tau_i$. In the second step the abstracted local workflows are combined in the same way as described in Section 3.

Figure 4 depicts the bilateral WF-Net for buyer $B$ and accounting $A$ representing all transitions that do not involve $B$ and $A$ as silent transitions $\tau$. In the buyer workflow $\tau_9$ represents the sequence of transitions labeled $B\#L\#deliver\_conf$, $L\#A\#deliver\_conf$, $A\#B\#delivery\_conf$, and $B\#A\#order\_conf$, while in the accounting workflow the transitions labeled $A\#L\#deliver\_conf$ and $L\#A\#deliver\_conf$ are relabeled by $\tau_6$ and $\tau_7$ respectively. The bilateral WF-Net representing the communication between logistics and accounting is depicted in Figure 5.

4.2. Resolving Cycles

As will be illustrated in Section 4.4 cycles in the bilateral WF-Net need to be resolved, because otherwise parameters would not be guaranteed to be immutable, but could be changed due to the cycle after their initial assignment. To construct these we need to resolve the cycles in the underlying local workflows. Cycles may be loops as contained in the buyer and logistics workflows depicted in Figure 2 or recursions. In the following loops are discussed in detail; a similar approach for recursion exists. A loop can be resolved by simulating it as a sequence of at most $N$ repetitions of a loop step:

- The transitions forming the loop are replaced by two subsequent transitions labeled $\tau_{loop}(1)$ and $\tau_{loop}(N+1)$, where the first transition initiates the loop and the second one represents the end of the loop.
Transitions in a loop step are encapsulated by silent transitions $\tau_{\text{loop}}(i)$ and $\tau_{\text{loop}}(i+1)$ representing the start and end of the $i$-th step of the loop. To execute the loop not all $N$ steps have to be performed, thus an additional transition $\tau$ is added to shortcut a single step.

- Parameters of transitions within a loop step $i$ are made unique by adding the suffix $i$ to each parameter.

- The output places of transition $\tau_{\text{loop}}(i)$ and the input places of transition $\tau_{\text{loop}}(i+1)$ share a single state.

Applying this construction to the buyer workflow results in the acyclic WF-Net depicted in Figure 6. All steps of the parcel tracking loop are represented by equivalent WF-Subnets except for the varying parameter indicating the number of the step. To reduce the complexity of this WF-Net a shorthand notation of the repetition is introduced, where only a single step is contained and the maximum number of iterations $N$ is annotated (lower half of Figure 7).

![Figure 6. Acyclic Buyer WF-Net](image)

The loop in the logistics workflow can be simulated in a similar way (upper half of Figure 7). Based on the constructed acyclic local workflows of the buyer and the logistics the bilateral WF-Net depicted in Figure 7 can be constructed.

### 4.3. Decentralized Consistency Checking

Deciding consistency of a multi-lateral collaboration in a decentralized fashion proceeds in three steps:

1. **Propagation**:
   Parameter constraints and occurrence graph constraints about already performed transitions are made available to all parties involved in the multi-lateral collaboration. This comprises:
   
   (a) Propagation of constraints within the WF-Nets representing the local workflows.
   
   (b) Propagation of constraints within bilateral WF-Nets.
   
   (c) Repeat the above steps until a fixpoint has been reached, that is, no further constraints can be propagated.

2. **Decentralized Consistency Checking**:
   Each party checks consistency of its bilateral WF-Nets and the WF-Net representing the local workflow. If all WF-Nets are consistent, then the party considers the multi-lateral collaboration to be consistent until any other party falsifies this decision by considering the multi-lateral collaboration to be inconsistent.

3. **Consensus Making**
   A protocol is required to decentrally check whether all parties consider their local WF-Nets as consistent, and to inform all parties about the final consensus. One possible approach for this is to determine a leader election algorithm. The coordinator starts a minimal spanning tree algorithm setting up a hierarchical structure of the parties. Based on this structure a classical 2-Phase-Commit protocol can be used to collect intermediate results of the partners, deriving a decision and
informing all parties on the result. This step is not further elaborated in this paper.

Step 1 is required because the bilateral WF-Nets hide all parameter and occurrence graph constraints that is not immediately seen by the two involved parties. Without propagating this information two of the bilateral WF-Nets in our example would be inconsistent:

The bilateral WF-Net for logistics and accounting (Figure 5) is inconsistent, because a message $A#L#deliver(it,a)$ with an amount $a$ greater than 100 violates the constraint of the “receiving” transition $A#L#deliver$. However, the guarantee of this constraint is only provided by the bilateral WF-Net for accounting and buyer, which is not seen by the bilateral WF-Net for logistics and buyer.

The bilateral WF-Net for buyer and logistics (Figure 7) is inconsistent, because the sequence $B#L#get_status - L#B#status$ results in a deadlock. The two “receiving” transitions $B#L#get_status(1,2)$ are both enabled after the corresponding ”sending” transition in the buyer’s workflow has been fired. Since the selection of the enabled transition is non-deterministic, both options must be considered for consistency checking. Taking transition (1) for direct parcel tracking leads to a deadlock since the $B#L#terminate$ transition can not be fired afterwards, thus, the final place can not be reached. Thus, the WF-Net is considered to be inconsistent although it is guaranteed by the accounting workflow that the transition (1) that causes the deadlock will never be fired, because no transition $A#L#auth$ will be sent.

Since the global WF-Net is consistent, the decentralized consistency checking without propagation of constraints is too restrictive. In the following we will show how to propagate constraints.

4.4. Propagation of Parameter Constraints

The goal of parameter constraint propagation is to make sure that all parameter constraints can be met, even though they may not immediately be visible in a bilateral WF-Net. The parameters of transitions are assumed to be immutable, that is, after they have been set initially they can not be changed. As a consequence, a parameter constraint holds for all transitions following the transition at which it has been specified. On these grounds parameter constraints can be propagated to all following transitions within a workflow as well as to the workflow of the partner.

The result of propagating the constraint on amount $a$ annotated to the transition labeled $B#A#order(it,p,a)$ within the bilateral WF-Net for buyer and accounting (Figure 4) is depicted in Figure 8. The result of propagating this constraint to the accounting workflow (Figure 2) is depicted in Figure 9. By further propagation of the constraint to the bilateral WF-Net for logistics and accounting (Figure 5) results in Figure 10.

Due to the propagation of the parameter constraints the deadlock discussed in Section 4.3 can no longer occur. As a consequence the bilateral WF-Net is consistent.

4.5. Propagation of Occurrence Graph Constraints

The goal of propagating occurrence graph constraints is to discard all those transitions, which cause a deadlock in a bilateral WF-Net but will never fire due to constraints imposed by the invisible part of the global workflow. An example for such a transition is the $B#L#get_status$ transition.
(1) in the bilateral WF-Net between logistics and buyer in Figure 7.

When the global workflow is known, such spurious transitions can be discarded by analyzing the occurrence graph of the global WF-Net as described in Section 3. A transition can be discarded, if it does not occur in the occurrence graph.

In the following we will describe how the occurrence graph can be constructed only on the basis of the bilateral workflows without explicating the entire global workflow. The approach consists of two steps. In the first step for each bilateral WF-Net occurrence graphs are constructed and used to discard spurious transitions. In the second step these transitions are also discarded in the other bilateral WF-Nets. As a consequence, the occurrence graph constraints of one bilateral WF-Net are propagated to the other bilateral WF-Nets. These two steps are repeated until a fixpoint is reached, that is, no further transition can be discarded.

In the example, the "receiving" transition \(A#L\#auth\) within the bilateral WF-Net for accounting and logistics (Figure 5) does not appear in the occurrence graph, since the corresponding "sending" transition does not exist. Thus the transition can be discarded (Figure 10). Further, the same transition occurring in the bilateral WF-Net for logistics and buyer (Figure 7) labeled \(\tau_{14}\) can also be discarded resulting in Figure 11. Next, the occurrence graphs of the modified bilateral WF-Nets involving logistics have to be recalculated. Analyzing the occurrence graph of the modified bilateral WF-Net between buyer and logistics reveals that the transition \(B#L\#get\_status\) (1) and the subsequent transition \(L#B\#status\) of the logistics workflow will never be fired, thus it can be discarded resulting in the final bilateral WF-Net for buyer and logistics depicted in Figure 12. The resulting bilateral WF-Nets (Figures 8, 10, and 12) represent a fixpoint.

The removal of the \(A#L\#auth\) transition of the bilateral WF-Net for logistics and accounting has been propagated to the bilateral WF-Net for logistics and buyer by removing the corresponding transition labeled \(\tau_{14}\). Due to the propagation of the occurrence graph constraints the deadlock discussed in Section 4.3 can no longer occur. As a consequence the bilateral WF-Net is consistent.

![Figure 11. Shorthand Notion of the Bilateral WF-Net for Buyer and Logistics after discarding transition \(A#L\#auth\)](image)

In the example, the "receiving" transition \(A#L\#auth\) within the bilateral WF-Net for accounting and logistics (Figure 5) does not appear in the occurrence graph, since the corresponding "sending" transition does not exist. Thus the transition can be discarded (Figure 10). Further, the same transition occurring in the bilateral WF-Net for logistics and buyer (Figure 7) labeled \(\tau_{14}\) can also be discarded resulting in Figure 11. Next, the occurrence graphs of the modified bilateral WF-Nets involving logistics have to be recalculated. Analyzing the occurrence graph of the modified bilateral WF-Net between buyer and logistics reveals that the transition \(B#L\#get\_status\) (1) and the subsequent transition \(L#B\#status\) of the logistics workflow will never be fired, thus it can be discarded resulting in the final bilateral WF-Net for buyer and logistics depicted in Figure 12. The resulting bilateral WF-Nets (Figures 8, 10, and 12) represent a fixpoint.

The removal of the \(A#L\#auth\) transition of the bilateral WF-Net for logistics and accounting has been propagated to the bilateral WF-Net for logistics and buyer by removing the corresponding transition labeled \(\tau_{14}\). Due to the propagation of the occurrence graph constraints the deadlock discussed in Section 4.3 can no longer occur. As a consequence the bilateral WF-Net is consistent.

![Figure 12. Extended Bilateral WF-Net for Buyer and Logistics](image)

5. Related Work

Multi-lateral collaborations have been investigated in the fields of workflow management theory, logic based approaches, and coordination theory. All these approaches have in common that they are limited to centralized multi-lateral collaboration establishment.

### Workflow Management Theory

Models for representing workflows known from workflow theory are finite state automata [4], Petri Nets [5], or statecharts [2]. They allow to represent control flows, but in many cases introduce a notion of data flow. In particular, all these approaches are based on structural representations, similar to the one presented in this paper, which have been applied to centralized collaboration establishment so far. The same holds for technology specific language proposals based on these approaches like for example BPEL re-
lated to web services or the ebXML process specification language BPSS, since they are related to the before mentioned formal models.

Alternatively, the workflow community has addressed the issue of direct coordination between workflow engines [3] rather than implementing coordination based on bilateral communication as presented in this paper. Corresponding approaches require a centralized coordination checking the consistent execution of the distributed workflow engines, thus do not support a decentralized decision and execution of multi-lateral collaborations.

Logic Based Approaches
Dynamic deontic logic [8] is an approach where a transition named action represents a change from one propositional world to another. In addition, deontic operations express permission, prohibition and obligation of actions, that is corresponding transitions. Another logic based approach is Courteous Logic Programs [1] being a non-monotonic logic, that allows to change predicate truth assignments. Both approaches have been used to represent workflows, but have never been applied to decentralized collaboration establishment. In particular, the propagation as introduced in this paper can be represented by these logics, if all transitions in all workflows are deterministic. Non-determinism can not be handled due to insufficient support of backtracking.

Coordination Theory
Within [13] coordination is identified as a critical principle in the study of workflow within organizations. [6] defined "coordination as managing dependencies between activities" and examined all sorts of scientific domains like social, psychological, economical and computer science to identify relevant activities and their dependencies. Most of the approaches are based on a general indirect, anonymous, undirected and asynchronous communication model, where data can be inserted, read and withdrawn from a shared multi-set.

Coordination theory has also been applied to workflow coordination [9] being data-driven rather than control-driven. An example of such an approach are place transition nets modeling state changes by events. The WorkSpace [10] approach is based on the notion of steps representing a transformation of one or several data elements. These approaches are all centralized, thus, can not be applied to decentralized decision making for multi-lateral collaborations establishment.

6. Conclusion and Future Work

In this paper, we have illustrated the decentralized consistency of multi-lateral collaborations, where consistency means the fulfillment of all local constraints and deadlock freeness. Opposed to centralized approaches as e.g. proposed by Aalst [12] and Martens [7] the decentralized approach provides higher flexibility to facilitate Service Oriented Architectures. The presented approach is based on propagation of local constraints until a fixed point has been reached. Future work will formalize the presented approach and provide a formal proof. Further, the specified form of the consensus problem raised in Section 4.3 will be discussed in more detail and a concrete solution will be provided.

References