COMPLIANCE VERIFICATION USING A JOINT MODEL OF OPEN WORKFLOW NET AND GLOBAL CALCULUS

Wenjie Sha¹, Li Zhang², Yue Ben², and Yahui Lu³
¹ Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
² School of Software, Tsinghua University, Beijing 100084, China
³ School of Software, Shenzhen University, Shenzhen, China

ABSTRACT
In a collaborative workflow environment, interaction protocol is adopted as a contract, which every peer process must comply with. In order to verify the compliance, we develop a joint model of Open Workflow Net (oWFN) and Global Calculus by establishing a connection between their grammars, and denote the joint model Dynamic oWFN (DoWFN). We construct the Reachability Graph (RG) of DoWFN, and we prove that by analyzing RG of DoWFN, we can verify the compliance of peer process and interaction protocol. We adopt 13 oWFN reduction rules to shrink the size of Reachability Graph, thus reducing the time complexity of the compliance verification. In order to validate our method, we implement a tool tValidator and test it on practical scenarios.

KEYWORDS
Collaborative workflow, compliance, verification, reduction rule, interaction protocol

1. INTRODUCTION
E-commerce works in a distributed environment (Chen and Hsu, 2000, Chiu et al., 2004), where multiple parties with heterogeneous technology participate in a certain predefined order. Business-to-business (B2B) requires the collaborative infrastructure to be correct in execution, and expandable to new participants. Correctness is fault-proofed by a mechanism which validates and executes processes. A fault in execution time is much less tolerable than one in design time. A B2B environment is composed of different organizations, where change of one participant is transparent to others. All influences are contained within the boundary of interaction protocol (Schroeder and Mayer, 2008), e.g. a contract. One longstanding problem is how to identify whether the peer process complies with the protocol (Schroeder and Mayer, 2008, Beyer et al., 2005). Effective verification is necessary to detect potential failure in collaboration in design time, therefore avoiding possible litigation and significant economic loss.

In order to verify whether a peer process complies with the interaction protocol, we develop a novel model-Dynamic oWFN (DoWFN), a joint model of Open Workflow Net (oWFN) and Global Calculus (GC). We discover three causal relations between transitions in oWFN and GC. Based on rule1-3, we further develop Reachability Graph (RG) on DoWFN, and verify compliance using RG analysis. In order to reduce the time complexity of computing RG, we suggest 13 oWFN reduction rules.

Our contributions include:
- The development of DoWFN and its Reachability Graph to verify compliance between peer process and interaction protocol.
- 13 reduction rules to dramatically reduce the time complexity of verification.

The rest of this paper is organized as follows. Section 2 explains the nature of collaborative business process management and presents different modeling languages to describe interaction protocol. Section 3 refers to various validation techniques and introduces our solution. Section 4 explains the design and use of our validation tool tValidator. Section 5 concludes the work and discusses the future work.
2. RELATED WORK

E-Commerce prospers in a collaborative environment (Chiu et al., 2004, Chen and Hsu, 2000), where companies collaborate with each other across the organizational boundary. Some data and process structures are kept private to the hosting organization. In order to fulfill this purpose, different architectures have been proposed. Two general approaches are taken, i.e. orchestration and choreography. Through orchestration approach, based on SOA infrastructure, WS-BPEL (OASIS, 2007) composes web services and communicates with its partner by exchanging messages. On the other hand, choreography defines interaction protocol, which sets up the message exchanging order between collaborating processes. Such protocol or language is sometimes called Choreography Description Language (CDL) (group, 2004). CDL is a form of contract or rule in a collaborative environment. WS-CDL is a CDL based on web service developed by W3C’s WS-CDL Working Group, an xml-based protocol. Partner Interface Process (PIP) (RosettaNet, 2010) defines the message exchange pattern between participating processes and supports a few industrial business protocols yet. Global Calculus (GC) (Marco et al., 2007) is an algebraic form of interaction protocol originated from WS-CDL, which has a good correspondence with industrialized interaction protocols.

Collaboration of involved organizations is usually implemented by composition of WS-BPEL or peer processes. Such collaboration must meet the requirement of behavioral compliance to perform seamlessly (Zhao et al., 2005). In order to ensure such compliance, 5 categories of verification methods are proposed. The first category neglects interaction protocol and composite orchestrations using formal model. Compliance verification is performed upon the integrated model. (Martens et al., 2006) converted BPEL4WS’s to Petri Nets, combines them and checked compliance on Petri Net; however, this approach reveals private business process, thus inappropriate for collaborative environment. (Sun and Yang, 2008) Proposed CoBTx-Net is based upon HCPN to model business collaboration. Such approach is only suitable for intra-organization environment, where verification can be performed centrally.

The second decode an orchestration into an operation manual for its partner to check the compliance against. In this approach, there is no choreography involved either. (Lohmann et al., 2007) produces an operation guideline based on automata and open Workflow Net for WS-BPEL. (Sha et al., 2009) produces a collaboration guideline based on Pi calculus and Behavioral Tree for general peer process. These methods cannot be applied in multi-party-collaboration scenarios, because operation manual cannot be validated against more than one partner. And the resulted guideline is often large for user to compute with.

The third category deals with the case that multiple orchestrations collaborate according to a predefined interaction protocol or choreography (Haller et al., 2009). This category of methods in some way, abstract interaction sequence from orchestration and use simulation relationship to verify the compliance against predefined interaction protocol. (Baldoni et al., 2006) represents individual process and protocol using finite state automata, and defines interoperability using conformant simulation. (Tasharofi and Sirjani, 2009) validated conformance by modeling WS-CDL in Reo, subsequently into CASM and define simulation relation between orchestration CASM and choreography CASM. These approaches cannot deduce interaction sequence from complicated interaction protocol and some undesired collaboration property such as live lock is hard to examine. And their mathematical foundation is just too complex to prove easily or understand.

The fourth category approaches the case by converting both orchestration and choreography into the same model, e.g. finite state automata or LTS. They then integrate the orchestration and choreography into a unified model, thus performing verification upon the intermediate result. (Beyer et al., 2005) checks compatibility and refinement by solving propositional constraints. (Schroeder and Mayer, 2008) uses UML4SOA to describe orchestration and UML protocol state machine to express choreography, and the author converts them to modal I/O LTS. (Yi and Kochut, 2004) specifies a CP-nets-based process composition model to unify both conversation protocol and composition and verify the interaction. These methods lead to information loss during the model translation, because orchestration and choreography are intrinsically different, the conversion might not work well for the variety of industrial process and protocol languages.

The fifth category approaches the case by incorporating the models of orchestration and choreography into a unified one, thus leaving most well-formed syntaxes intact. (Fei and Li, 2007) added trigger syntax to Pi calculus as CDL, and link communication transition in Petri Net with channel in Pi, but require all participants’ process to perform verification thus inappropriate for collaborative environment.
Our approach lies in the fifth category, because we use oWFN as the orchestration modeling language, Global Calculus as the choreography description language, and establish the link between them, therefore incorporating them into a unified model. Through this mean, information contained in oWFN and Global Calculus are not lost, besides, oWFN and Global Calculus are both easy to understand and well researched for its respective purpose as orchestration and choreography modeling language. Our approach is also superior because of the verification capability in multi-party collaboration circumstances, and it can be performed by each peer process separately to distribute the computational load and avoid information leaking.

3. PRELIMINARY

In Collaborative Workflow environment, each peer process is modeled independently of its counterpart, but it must comply with the interaction protocol (Beyer et al., 2005, Schroeder and Mayer, 2008). We use Global Calculus (Marco et al., 2007) as our interaction protocol description language, which is concise, descriptive and is originated from WS-CDL (group, 2004). As for peer process, there are various types of modeling languages such as EPC, YAWL, Petri Net, Pi Calculus, End-point Calculus, and etc. A lot of research has been undertaken on the formalization of Petri Net, and Petri Net is comparatively simple, yet powerful to describe collaborative workflow.

3.1 Open Workflow Net (oWFN)

An Open Workflow Net (oWFN) (Lohmann et al., 2007) is a liberal version of WF-net (Workflow Net), a type of Petri Net with communication place. There are two types of communication places, namely incoming place and outgoing place, representing channel to receive message and channel to send message to external partners.

Communication place should be linked either inward or outward to only one transition (task) and the linked task is a communication task, meaning it interacts with the environment.

An oWFN can be formally defined as following.

- An oWFN= (P, T, F). P is the set of all places (states). T is the set of all transitions (tasks or actions). And F indicates the set of information and data flow between places and transitions.
- Two sets in, out ⊆ P, such that for all transitions t ∈ T holds: if p ∈ in (p ∈ out) then (t,p) ∈ F((p,t) ∈ F), and there is bijection set of communication tasks with in, out.
- A distinguished marking Minit called the initial marking of oWFN.
- A distinguished marking Mfinal called the final marking of oWFN.

The definition of soundness of oWFN can be defined as following:

**Definition 1** For any oWFN, we can get its degraded WF-net by removing all communication places and attached edges. An oWFN is sound if its degraded WF-net is sound.

An oWFN of Seller process in the basic example scenario in Figure 1 is depicted in Figure 2.

![Figure 1 Graphical representation of BSH protocol](image-url)
3.2 Interaction protocol

Global Calculus (GC) (Marco et al., 2007) is an algebraic form of interaction protocol originated from WS-CDL. Its power of describing the rule of collaboration in a simple and accurate way offers a good basis for us to model choreography (group, 2004) and base research on. A Buyer-Seller-Shipper (BSH) protocol in Figure 1 Graphical representation of BSH protocol for purchasing situation among a buy, a seller and a shipper is demonstrated in Figure 3. A thorough specification of GC can be found in (Marco et al., 2007). We can depict state space of GC following its reduction rule.

4. COMPLIANCE VERIFICATION

The oWFN describing peer process uses communication places to represent interaction. Incoming communication place or outgoing communication place has the property that it is linked with only one transition. Linked transition is the action that needs to be performed interactively with partners. In BSH example, place 1 (id) represents the “quote message” channel and there is an edge directing to transition “receive quote”. Place 1 together with transition “receive quote” is the projected counterpart in oWFN of action “Buyer->Seller: B2Sch<QuoteRequest>” in GC.
The action set of interaction protocols and communication place set in peer process has a bijection relationship which is shown in Table 1. However, there is one special case. The “Buyer->Seller: InitB2S<B2Sch>” action corresponds to the starting place of process Seller. The symbol “0” means that the interaction protocol finishes successfully.

Table 1 Bijection of action in GC and communication place in oWFN

<table>
<thead>
<tr>
<th>Action in interaction protocol</th>
<th>Communication place ID</th>
<th>Method (+/- token)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buyer-&gt;Seller: InitB2S (B2Sch)</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>Buyer-&gt;Seller: B2Sch&lt;QuoteRequest&gt;</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td>Seller-&gt;Buyer: B2Sch&lt;QuoteResponse,vquote,xquote&gt;</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>{Buyer-&gt;Seller:B2Sch&lt;QuoteAccept&gt;}</td>
<td>3</td>
<td>+</td>
</tr>
<tr>
<td>Seller-&gt;Buyer: B2Sch&lt;OrderConfirmation&gt;</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>Seller-&gt;Shipper: InitS2H (S2Hch)</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td>Seller-&gt;Shipper: S2Hch&lt;RequestDeliveryDetails&gt;</td>
<td>6</td>
<td>-</td>
</tr>
<tr>
<td>Shipper-&gt;Seller: S2Hch&lt;DeliveryDetails,vdetails,xdetails&gt;</td>
<td>7</td>
<td>+</td>
</tr>
<tr>
<td>Seller-&gt;Buyer:B2Sch&lt;DeliveryDetails,xdetails,ydetails&gt;.0</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>+ {Buyer-&gt;Seller:B2Sch&lt;QuoteReject&gt;.0}</td>
<td>4</td>
<td>+</td>
</tr>
</tbody>
</table>

When a request for quote arrives, a token will be passed from Buyer’s side of the process into place 1. In fact, when validating the compliance of Seller’s process in Figure 2 with the interaction protocol above, we “manually” put a token in place 1 (true simulation of the Buyer->Seller:B2Sch<QuoteRequest> action or the virtual Buyer. Each Buyer->Seller: B2Sch<action> action causes a token being put into correspondent communication place immediately. Whereas, Seller->Buyer: B2Sch<action> executes and affects the state of oWFN reversely. Seller->Buyer:B2Sch<action> can take place if and only if there are tokens in corresponding communication place, and the effect is that a token will be removed from that place. We already know that state transition of oWFN can be depicted in Reachability Graph (RG), and RG is used to check the soundness of oWFN.

We formally define three rules as the mapping between transitions in oWFN and GC

- **Rule 1 Receiving action** of GC results in a token put in corresponding communication place.
- **Rule 2 Sending action** results in a token removed from corresponding communication place.
- **Rule 3** The preconditions of sending action is that there are more than one token in corresponding communication place, so that one token can be consumed by the sending action.

Receiving action is of the form: partner-> process and sending action is of the form: process->partner.

The three rules say that state transition in GC state space results in simultaneous state transition in oWFN state space, thus leading to a state transition in the joint state space, DoWFN (we define the joint model later in this section). We already know how to produce the state space of oWFN, and we can also produce the state space of GC. Thus, we can define a joint model of oWFN and GC, and its state space as following.

**Definition 3** Denote the joint model of oWFN and GC Dynamic oWFN (DoWFN), and the state space of DoWFN $P_D$. State of $P_D$ is a tuple $(M,G)$, which means that at this point, peer process is in state $M$ and interaction protocol is executed to stage $G$.

$$P_D \equiv P_{oWFN} \times P_G$$

$(M_{0},G_{0})$: The initial state of DoWFN; $(M_{final},0)$ : the final state of DoWFN.

Any other state is an intermediate state, which means either the peer process or the interaction hasn’t completed yet. $P_D$ is a digraph, whose node is tuple of state in $P_{oWFN}$ and $P_G$. We indicate the transition of state in $P_D$ in the following theorem.

**Theorem 2** State transition in $P_D$ obeys the following rule. For every condition met, there is one outgoing edge.

For any $(G_i, M_s) \in P_D$, if

- rule 3 permits $G_i \rightarrow G_j$, and $M_s \xrightarrow{m1,m2} M_i$, then $(M_s,G_i) \rightarrow (M_i,G_j)$
- $M_s \rightarrow M_j$, then $(M_s,G_i) \rightarrow (M_j,G_i)$
So, following the above theorem, a tuple-form state in P_D might have lots of or no subsequent state. Task execution in P_{oWFN} will not result in transition in P_G; however, it might make P_G reducible in the next step.

Figure 4 Reachability Graph of DoWFN of Seller and BSH protocol

From Figure 4, we see there is only one sinking state, which means that every execution path of peer process Seller in BSH protocol-based environment will complete successfully, and the peer process complies with the BSH interaction protocol. Therefore, as long as any Buyer and Shipper also comply with the protocol, Buyer, Seller, and Shipper can collaborate well.

We can use RG of DoWFN to analyze compliance of any complicated process and interaction protocol, as long as they can be transformed to oWFN and Global calculus. To tackle ill-formed peer process, we put bounding constraints on DoWFN. If there is more than one token in any place at any time, the verification process halts and report an error.

We know from above that a DoWFN is composed of oWFN and Global Calculus respectively describing peer process and interaction protocol. So, now we present the most important theorem in this work as following.

**Theorem 3** If there is a path, on the RG, from every state of the DoWFN to the final state, then the peer process complies with the interaction protocol and there is no live lock in the interaction protocol-based collaboration. Otherwise, they either don’t comply or incur live lock.

**Proof**

**A → B**: The existence of path from every state to the final state indicates there is execution path from both oWFN part and GC part of each state to respective final state, which is a dictator of non-existence of live lock or deadlock; therefore, the process complies with the interaction protocol.

**B → A**: There is no such DoWFN state, in which the process can no longer evolve or transit to the final state. Thus for each DoWFN state, a reduction path to the final DoWFN state exists.

Theorem 3 is the most important result we present in this work. It indicates a straightforward approach to verify the compliance between any peer process and interaction protocol.

5. REDUCTION

The oWFN modeling practical scenario often contains a huge amount of tasks. And Reachability Graph, a capable tool, therefore becomes a time consuming analysis to perform. (Lee-Kwang et al., 1985, van Dongen et al., 2005) studied reduction rules based on WF-net or generalized Petri Net. We propose 13 reduction rules for oWFN (presented in Table 2), which preserve the communication places and the compliance of oWFN with GC. By performing reduction, we can reduce the time complexity of verification process dramatically.

<table>
<thead>
<tr>
<th>ID</th>
<th>FIGURE</th>
<th>CONTENT</th>
<th>ID</th>
<th>FIGURE</th>
<th>CONTENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td><img src="image" alt="Diagram" /></td>
<td>(combination of serial transitions)</td>
<td>R7</td>
<td><img src="image" alt="Diagram" /></td>
<td>(combination of OR-Split)</td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Diagram" /></td>
<td>This rule is only applicable when the place is an internal place.</td>
<td></td>
<td><img src="image" alt="Diagram" /></td>
<td>Applicable only when place P is an internal place.</td>
</tr>
</tbody>
</table>
### Reduction Rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
<th>Applicability</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2</td>
<td>(combination of serial places)</td>
<td>Applicable only as both the two places are internal places.</td>
</tr>
<tr>
<td>R3</td>
<td>(combination of parallel transitions)</td>
<td>All the transitions in parallel can be merged into one. Always applicable.</td>
</tr>
<tr>
<td>R4</td>
<td>(combination of parallel places)</td>
<td>All internal places can be merged into one. Leave interface places unchanged.</td>
</tr>
<tr>
<td>R5</td>
<td>(combination of overlapped transitions)</td>
<td>Always applicable.</td>
</tr>
<tr>
<td>R6</td>
<td>(combination of overlapped places)</td>
<td>All internal places can be merged into one.</td>
</tr>
<tr>
<td>R8</td>
<td>(combination of AND- Split)</td>
<td>Applicable only as all the places P₂, P₃ and P₄ are internal places.</td>
</tr>
<tr>
<td>R9</td>
<td>(combination of OR-Join)</td>
<td>Applicable only when place P₁ is an internal place.</td>
</tr>
<tr>
<td>R10</td>
<td>(combination of AND-Join)</td>
<td>Applicable only as all the places P₂, P₃, and P₄ are internal places.</td>
</tr>
<tr>
<td>R11</td>
<td>(deletion of self-loop place)</td>
<td>Applicable only as the place is internal.</td>
</tr>
<tr>
<td>R12</td>
<td>(deletion of self-loop transition)</td>
<td>Applicable only as the place is internal.</td>
</tr>
<tr>
<td>R13</td>
<td>(fusion of loop-circuit)</td>
<td>If places P₁ and P₂ are both inner places, they two can be merged into one place as P₁₂, similar to reduction rule R2. Then if place P₃ is also an inner place, P₁₂ and P₃ can be merged into one as shown on the second arrow of R13 sketch graph. In other situations R13 will not be applicable.</td>
</tr>
</tbody>
</table>

The communication task involved should reside, after the reduction. Moreover, starting place and sinking place of oWFN should not be affected during reduction, or in other words, not removed.

All these rules concluded above are compliance preserving. By compliance preserving, we mean that if the pre-reduced oWFN is compliant with the GC, then the pro-reduced oWFN is also compliant with the GC, and vice versa.

We use these reduction rules over the oWFN in Figure 2 and obtain the reduced oWFN in Figure 55.

Recursive appliance of reduction rule to the oWFN in Figure 2 decreases the number of places from 27 to 18, a reduction rate of 30%, and the number of transition from 16 to 9, a rate of 40%. Since the Reachability Graph construction is an exponential complexity process, this reduction in oWFN size can result in tremendous time saving in RG construction and consequently verification process.
6. TVVALIDATOR

We have developed a validation tool tValidator implementing the verification and reduction techniques described above. The tValidator comprises five parts: importer of oWFN, oWFN reduction component, GC importer and parser, compliance checker, and visualization. An additional component to pinpoint execution path that lead to interaction failure is helpful and we plan to add this functionality into the tool in the near future.

We customize ProM (van Dongen et al., 2005), an open-source workflow modeling framework to implement our components. A module is developed to implement the oWFN reduction rules, and we implement a functionality to construct the RG from oWFN and Global Calculus. We then verify the compliance based on the constructed RG.

Using our tValidator, we verify the compliance between Seller process and BSH interaction protocol. We also successfully test our tool over many actual business processes, which justify our theory.
7. CONCLUSION

In this paper, we have developed a joint model DoWFN, which combines the characteristics of oWFN and Global Calculus. Based on this model, we have proposed a novel approach to verify the compliance of peer process and interaction protocol. We prove that if there is only one sinking state in RG, which is also a final state of DoWFN, and the peer process complies with the interaction protocol. Furthermore, we have introduced 13 reduction rules to reduce the complexity of our approach. In order to validate the practicability and performance in actual scenarios, a tool tValidator is implemented and tested.

In the future, we will integrate our tool tValidator with agent-based infrastructures or collaborative workflow management systems to facilitate the model checking of collaborative systems. Our idea can also be further extended to check compliance between WS-BPEL and WS-CDL, which are becoming the de facto standards in distributed computing and collaborative workflow environments.

ACKNOWLEDGEMENT

We thank Dr. Yingbo, Liu for his constructive suggestions. The work is partially supported by the National Natural Science Foundation of China (Grant no. 60973100, 60673008 and 60903114)
REFERENCES

Journal


Standard


Conference paper or contributed volume


