Meeting the Challenges of Collaborative Network Compliance – an exemplary view

Oyepeju Oyekola[1,1], Lai Xu[1], Paul de Vrrieze[1],

[1] Computing anf Informatics, Bournemouth University, Poole, BH12 5BB Bournemouth, United Kingdom
{ooyekola,lxu,pdvrieze }@bournemouth.ac.uk

Abstract. Ensuring the conformance of an organization's processes to certain rules and regulations has become a major issue in today's business world. As non-compliance with these regulations could cost organizations a considerable sum of money in fines or litigation or even loss of company reputation. Recently, the intelligent connectivity of collaborative networks of people, organization, machines, and smart things has become a high potential for value creation, and at the same time bring about some compliance challenges. Ensuring compliance in such a collaborative network environment (i.e., a dynamic and networked environment) is complicated due to its design principle for decentralized decision-making. To meet up with the various challenges of collaborative networks, this paper reviews an existing compliance approach, i.e., eCRG's (extended Compliance Rule Graph) specification language and its decomposition approach. A real-world collaborative case is used to examine which compliance properties can be checked by the existing approaches and which compliance properties cannot be checked yet. We further explore how to extend the approach to meet up with the identified challenges of collaborative network compliance, which are served as a base for supporting the automated compliance checking of Collaborative Process either at design time or run-time.

Keywords: Collaborative process, Collaborative network, Business compliance rules, Global compliance rule, Decomposition rule.

1 Introduction

Ensuring the conformance processes to certain rules and regulations has become a major issue in today's business world. As non-compliance with these regulations could cost organizations a considerable sum of money in fines or litigation or even loss of company reputation. Recently, the intelligent connectivity of collaborative networks of people, organization, machines, and smart things has become a high potential for value creation, and at the same time bring about some compliance challenges. For instance, Collaborative network/processes present a unique attribute

1
.

such as the need to conform to security and privacy requirements (i.e., keeping specific organization data private), the need to support the different process perspective, the need to conform with various regulatory requirement as a cross border organisations, as well as conforming to the constant changes in policies and regulations (e.g., COVID-19, BREXIT), all this presents a unique challenge. While several compliance verifications approaches have been proposed in the literature, these approaches still lack the full support for the automated compliance checking of collaborative processes.

In our previous paper [1], we identify some of the different challenges as a requirement needed to support the automated compliance checking of collaborative processes. We used a motivating use case of a collaborative process involving six partners adapted from [2], and one of the key challenges is checking the compliance of processes involving a high level of dependency and response between each partner activity. As well as detecting the imminent violation of instance execution and detecting the potential violators. Generally, Business Process Compliance lifecycle involves different compliance strategy involves the design-time (preventive approach) and run-time compliance checks (monitoring approach) [3]. Based on the literature, compliance monitoring has been identified as an important building block in the process lifecycle. The reality is that even if a business process has been checked during design time, there is no certainty that the corresponding running process instance will be compliant because of human and/or machine-related errors [1]. This implies that after designing a process model and the actual execution of a process is initiated, the running process instances need to be constantly monitored to detect any inconsistencies or violations early. As well as providing a reactive and proactive countermeasure i.e., recommending what next to do and predicting what will happen in the future instances of execution. Therefore, this research paper focuses on supporting the compliance of collaborative with the varied requirement from multiple process perspectives i.e., control flow, data flow, resource flow and time perspective. To support this functionality, the paper reviews an existing compliance i.e., decomposition approach [4],[5] using eCRG as a specification language. A real-world collaborative case was used to examine which compliance properties can be checked by the decomposition approach and which compliance properties cannot be checked yet. We further explore how to extend the approach to meet up with the identified challenges of collaborative network compliance [1].
The rest of the paper is structured as follows: Section 2 describes the case description used in identifying some of the challenges of collaborative process adapted from [2]. Section 3 explore and discusses the eCRG approach and their applicability to our use case. In section 4, a real-world collaborative case is used to examine which compliance properties can be checked by the decomposition approach and which compliance properties cannot be checked yet. Lastly, Section 5 gives the summary of the paper, we highlight the challenges of collaborative networks and discussed part of the solution that are partially met, as well as our future research.

2 Case Description

The A motivating use case of a collaborative process involving six partners adapted from [2]. The process starts with the policyholder who owns an insurance policy and reports any damage to the issued car. Euro Assist is the company that registers the claim received from the policyholder via the telephone and encourages approved garages. AGFIL is the insurance company that underwrites the car policy and decides whether the reported claim is valid or not. If the claim is valid, AGFIL will make payment to all parties involved. Lee Consulting Services (CS) works on behalf of AGFIL and manages the day-to-day emergency service operation. Lee CS access and determine whether the car requires an assessor after the assigned Garage estimated the repair cost, i.e., an assessor would be assigned to assess the damage of the car only when the repair cost exceeds a certain amount. They control how quickly garages will receive payment, as all invoices received from the Garage are sent through Lee CS, and further present the invoice to AGFIL to process the payment while ensuring that repair figures align with industry norms. The approved garages are then responsible for repairing the car after Lee CS has agreed upon the repair. The repair work must be carried out quickly and cost-effectively.

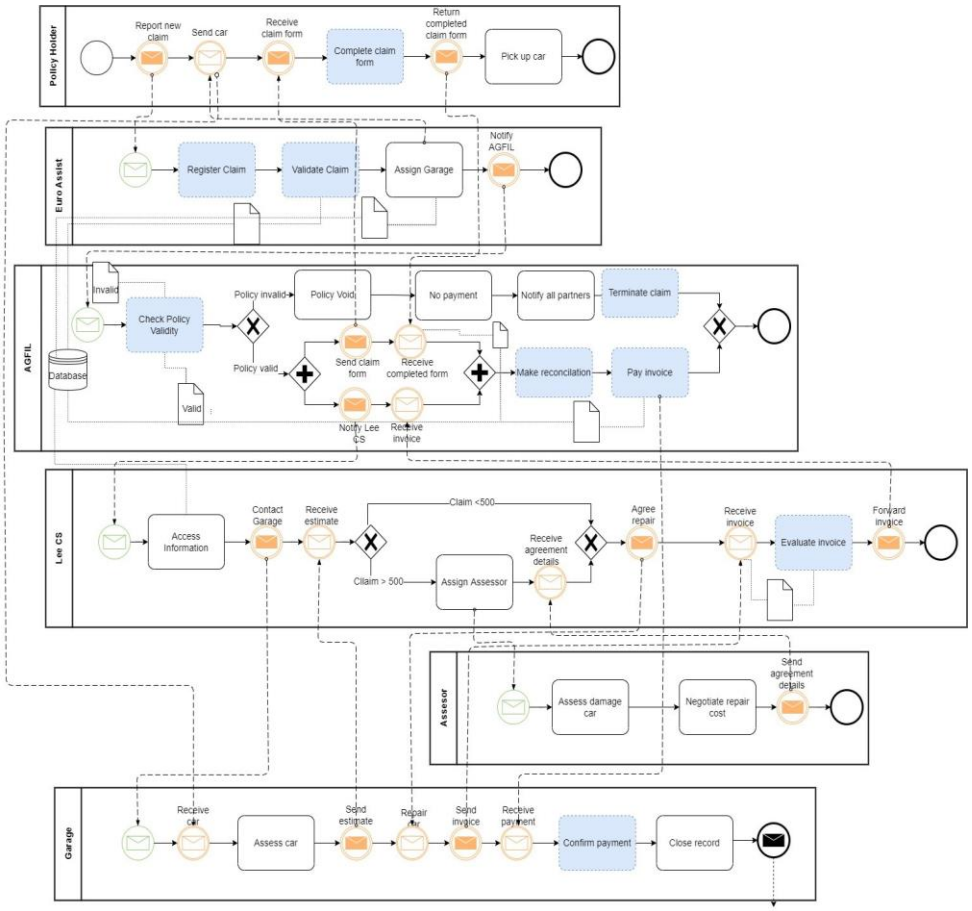


Fig. 1. Collaborative Model for Insurance Case [1] [2].

3 Compliance Rule Language

Compliance rules must be comprehensible and at the same time should have precise semantics to enable automated processing and avoid ambiguities [3]. Therefore, the identification of suitable compliance rule language that can support all multiple process perspectives i.e., support control flow, data, resource, time, and interactions with process partners remain important. Several approaches for the formal specification of compliance rules have been identified in literature using languages such as the FCL (Formal Contract Language), LTL (Linear Temporal Logic), CTL (Computation Tree Logic) or other text based languages, but since these formal languages are complex and error-prone, some researchers like [8,9,10,11] suggested the idea of specifying compliance rule using visual notation such as BPSL [13],

Compliance Rule Graph (CRG) [12], BPMN-Q [6] etc. The visual approach of compliance rule is flexible and aids comprehensibility for domain experts [7]. However, it is worth noting that most of these exiting visual languages do lack the full support of all the various process perspective as it solely focusses on the specification of control flow perspective and an aspect of the data flow. For instance, CRG support solely the control flow perspectives while BPMN-Q support control flow and data conditions.

3.1 eCRG (extended Compliance Rule Graph)

To support the specification of the different process perspectives, [7],[19] presents extended Compliance Rule Graph (eCRG) i.e., an extension of CRG to visually model compliance rule and to enable the full support of multiple process perspectives regarding the control flow, data, time, and resource perspectives as well as the interactions of a process involving different partners [4]. It allows detecting compliance violations at run-time, as well as visually highlighting their causes. Additionally, it allows providing recommendations to users to proactively ensure a compliant continuation of a running business process [4]. The specification of an eCRG consists of a pre-condition and a postcondition. The former specifies when the compliance rule shall be applied or triggered, and the latter needs to be met to satisfy the compliance rule. Accordingly, the edges and nodes of an eCRG are partitioned into an antecedence pattern (precondition), and a related consequence pattern (postcondition) [20]. The eCRG semantics is formally specified through a translation of eCRG's into FOL (First Order Logic) expressions based on completed process logs. The feasibility of eCRG was scientifically evaluated in [19],[7], using different approach such as proof of concept prototype, empirical evaluations, its applicability to real world cases, as well as systematic comparison with LTL and compliance patterns. Based on the benefits of eCRG and the scientific evidence of eCRG, this study supports the use of eCRG for its compliance rule specification language.

3.2 Collaborative Business Process Models in eCRG

Several studies like [14],[15],[16],[17] has adapted the use of eCRG language to specify their compliance rule. Most of these paper mostly focus on the compliance verification of single business process. Since the focus of this study is on Collaborative Business Process (CBP), then this section reviews few works of collaborative process that based their language specification on eCRG. Supporting cross-organizational business processes involving multiple partners with respect to GCR (Global Compliance rule) is addressed in [18] using eCRG to specify the asserted rules and GCRs (Global Compliance Rules). The paper checked the compliance rules that needed to be rechecked after a change in CBP. The algorithm developed was used to detect impact of CBP changes on GCR. In [4], the authors describe how global compliance rule of process choreographies could be verified in a decentralized manner by each partner in the process collaboration and deals with the restricted visibility of process activity. The approach uses a decomposition-based

approach i.e., the decomposition of GCR into a set of assertion that can be checked by each partner locally making sure the privacy of each partner is not violated. The work was further extended in [5], with more complex rules with multiple antecedence patterns, extensions of theorem proofing and illustrations as well as the extension of decomposition algorithm. The feasibility of the approach was based on a prototypical implementation, which includes the use of a model checker to verify the correctness of the decomposition.

3.3 Decomposition Approach

This section looked at the decomposition approach as described in [4],[5]. The decomposition approach is applied when a GCR involves a private activity of one or several partners in a process collaboration. In such situation, each partner's private activities remain invisible to other partner, and no information about when or how these activities are executed and, therefore, cannot identify the dependencies between each activity involved in the GCR. As such, the original GCR are split into a set of assertions which are checked locally by each partner and are combined to generate the behavior of the original GCR. The decomposition process is based on a well-grounded theorem, representing a decomposition of a given compliance pattern. A decomposition algorithm is presented using transitivity properties to break down the initial GCR into derived assertions. Once the GCR is decomposed and the corresponding assertions are derived, each partner locally checks its derived assertions at runtime.

4 Declarative Representation of GCR using Decomposition Approach

The applicability of the decomposition approach is demonstrated using the Car Insurance Case depicted in **Fig. 1.** Using the use case, we examine which compliance rules can be checked by using the decomposition approach and which compliance rules cannot be checked yet. The plan is to optimize this approach to fully support the automated compliance checking of collaborative process. In general, collaborative model involves the choreography model, public model, and private model**.** The choreography model describes the global view of interactions among the partners in the collaboration (see **Fig. 2**). The public model describes the message interaction between the collaborative partners. Lastly, the private model includes tasks that are not visible to others in the collaboration (see boxes depicted in blue in **Fig. 1**). The process model is subject to various global compliance rules stems from various policies and regulations as shown in **Table 1.**

Table 1. Global Compliance Rule for Car Insurance.

Global Compliance Rule	***GCR conditions***

GCR 1	Garage must receive and confirm payment from AGFIL within a specific period	AGFIL make reconciliation and make payment to the garage only when: • Policyholder assures that a completed form will be returned to AGFIL within a specific period • Lee CS assures invoice is forwarded to AGFIL within a specified period.
GCR 2	Once Euro assist notifies AGFIL of any claim, AGFIL assures a claim form is sent to Policyholder within a specified period if only the claim is valid.	The claim form will only be sent to the policyholder only when claim validity is checked, and the DATA OBJECT is and remain in state "Valid" or otherwise the process end.

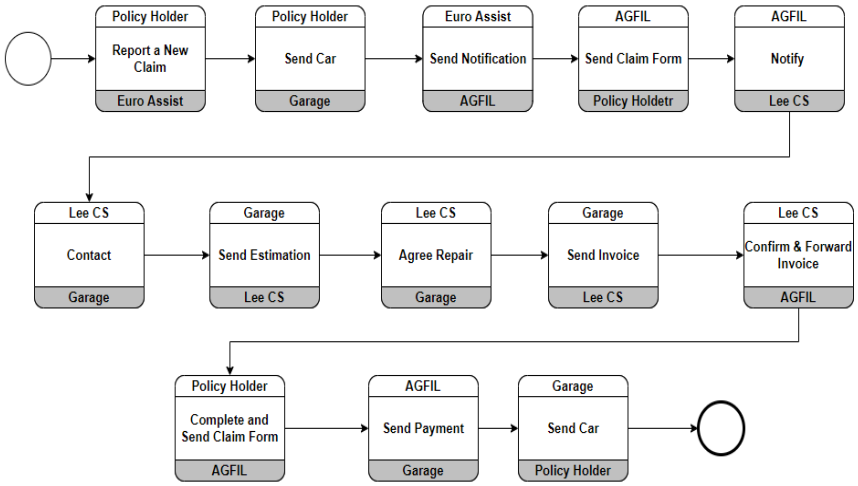


Fig. 2. Choreography Model for Insurance Case

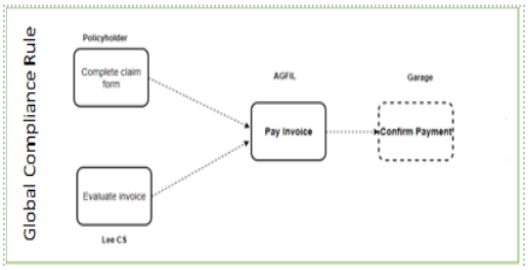
4.1 GCR 1

GCR 1 as shown in **Table 1,** involved the choreography, public as well as the private task of the partners involved in the GCR **(see Fig.1 and Fig. 2).** For instance, in **Fig.1**, **t**he task "pay Invoice" are invisible to the other partner and cannot have the idea of when the task is completed or not. However, there are few complexities as regards to this GCR as it involves some conditions that also needs to be verified first. The private activity "pay Invoice" involves a high level of dependency on the activity of one or several other partners in the collaboration making it difficult to decompose just the GCR 1 without the sub condition. These conditions need to be verified first to ensure compliance of GCR 1. And note that, the activity "complete claim form" for

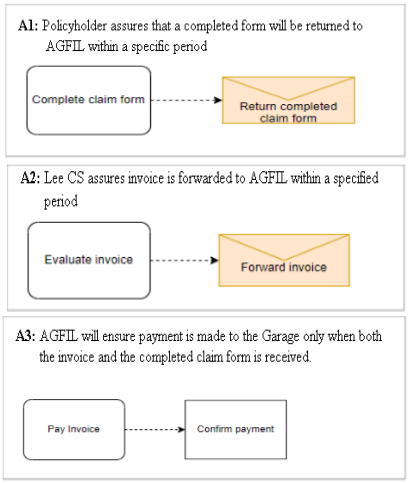
Policyholder and Evaluate Invoice from Lee CS are also private activities. Hence the need to apply the decomposition approach to ensure compliance.

GCR 1	Choreography task	Private task	Public task
Tasks/Activity	Send payment (Between AGFIL and Garage in **Fig.2**)	Pay invoice (AGFIL) Complete claim form (Policyholder) Evaluate invoice (Lee CS) Confirm payment (Garage) **(See blue boxes in Fig.1)**	Return completed claim form (Policyholder) Forward invoice (Lee CS) **(See Fig. 1: message interactions between partners)**

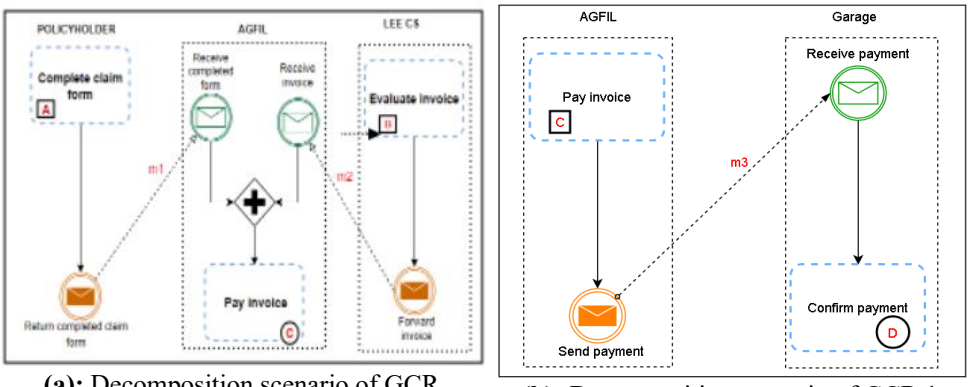
Global Compliance rule for GCR 1



Assertions



As stated earlier, we illustrate the decomposition process of GCR 1 using two scenarios to ensure simplicity and readability depicted in **Fig 3.** The decomposition of GCR1 include (a) the sub conditions that needed to be satisfied first and (b) rule that needed to be satisfied afterwards.



(a): Decomposition scenario of GCR condition for GCR1

(b): Decomposition scenario of GCR 1

Fig. 3. Decomposition of GCR 1

The scenario in (a) explains that if C is executed, then both A and B should have been executed before, and both m1 and m2 is successfully received by AGFIL.

$$\mathbf{A \dashrightarrow m1 \wedge \ B \dashrightarrow m2 \ \Rightarrow m1 \wedge m2 \dashrightarrow C}$$

It is worthy to note that the execution of just A and the successful receiver of m1 does not mean C will always be executed and the same with B. Though, there is no interaction between Lee CS and Policyholder.

In addition, the scenario in (b) explain that there is a message exchange between AGFIL and Garage which satisfies that the message m3 sent by AGFIL will be received correctly by Garage.

$$\mathbf{C \dashrightarrow m3 \text{ and } \ m3 \dashrightarrow D \ \Rightarrow C \dashrightarrow D}$$

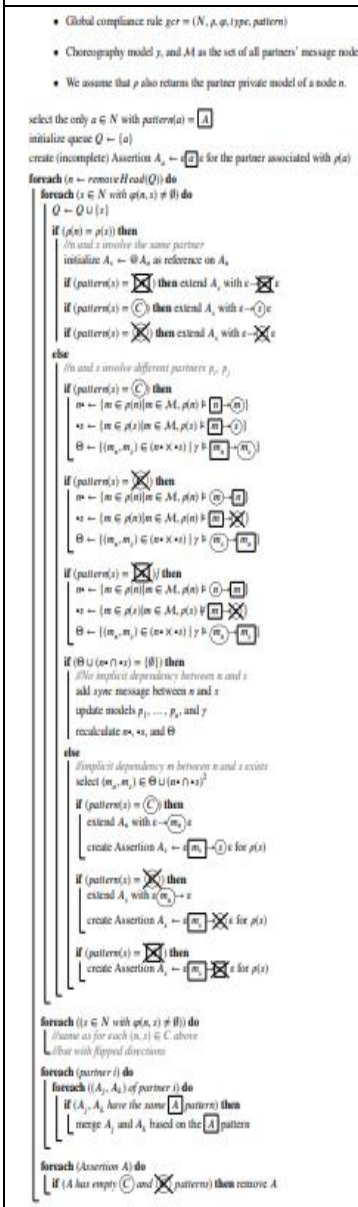
Which means that the execution of C and the successful receiver of m3, will lead to the execution of D. Once the decomposition as depicted in **a** and **b in** Fig 3 is verified, then we can ensure the correctness of GCR 1, that is:

$$\mathbf{A \wedge \ B \dashrightarrow C \text{ and } \ C \dashrightarrow D \Rightarrow A \wedge B \dashrightarrow D}$$

4.1.1 The Applicability of the Decomposition Algorithm to GCR 1

This section analyses the capabilities of the algorithm presented as Algorithm1 in [5], to the Car Insurance Case. Our aim is to check the applicability and complexity of the algorithm by analysing whether the algorithm can handle a complex collaborative process with high level of dependency among the partner's process and data conditions.

Applying the algorithm in [5] to GCR 1 in **Fig. 3** to derive the assertions using transitivity's is shown below:

Algorithm [21]	**Derivation of assertion for GCR 1**
	Let us assume that each node of GCR 1 is being assigned to Garage and the responsibilities include *p* (complete claim form) = policyholder, *p (*evaluate invoice) = Lee CS, *p* (pay invoice) = AGFIL, and *p* (confirm payment) = Garage. Then the algorithm walks through the nodes and start with node D i.e., Confirm payment and create an assertion for the Garage responsible for the task. Whenever the algorithm walks over a connector between two nodes n and s, which are assigned to different partners p{n) and p(s), the GCR is split at this position as the dependency cannot be evaluated by a single partner. At this point, since no other nodes of the GCR belongs to the Garage, the algorithm will then walk over a connector and cuts the respective connectors to create an assertion for AGFIL with the node ***Pay invoice.*** And since there are no nodes for AGFIL, hence, the algorithm cuts the connector but this time there are two different incoming connectors because of the AND gate. Therefore, two different assertions will be created for the respective partners involved. First, the algorithm will cut and create an assertion for policyholder with node ***Complete claim form.*** next, the second connector will be identified, and an assertion will be created for Lee Cs with the node ***Evaluate invoice.*** The algorithm tries to replicate the connector where the GCR was split through (transitive) message exchanges between the affected partners by applying the transitive relationships. Then, the algorithm calculates the sets of •n and •s and Θ, containing the messages that succeed or precede n and s. this time, a transitivity relationship will be used to replicate the connect where the GCR was split. However, this seems a bit challenging as the applicability of the algorithm could be easily applied to just (b) in **Fig 3.** without involving the GCR condition in (a). Secondly, the message exchange between the policyholder and Lee CS could also represent a data exchange among the

	partners which the present algorithm does not consider. At such, we propose the needs to optimize the algorithm to fully support sub conditions and Data perspectives.

Based on [5] explanation, the decomposition of the GCR into a set of assertions is subjected to a well-grounded theorem such that if a conjunction of hypothesis is true (i.e., the assertions), then the conclusion is true as well (i.e., GCR). Proving a set of theorems to ensure the correctness of the decomposition process for **Fig 3,** we realised that the eight proofed theorems (Transitivity, Zig-zag transitivity, Rightward chaining transitivity, Generic rightwards chaining transitivity, between pattern 1, Between pattern 2, Between pattern without loops, Requires transitivity) described in [5] cannot be applied to this scenario 1. Hence, we propose the need to extend the algorithm and provide a formal proof. We suggest the need to decompose (a) and (b) in Fig 3 separately for simplicity. That is decomposing Fig 3(a) first and followed by fig 3(b). To do that we propose the AND Chaining Transitivity.

Theorem 1 (AND Chaining Transitivity)
Let A, B and C be three activities or interactions such that $\mathbf{A} \wedge \mathbf{B} \dashrightarrow C$ (the antecedent of A and B will lead to the consequence of C): Both A and B must occur, for C to occur afterwards.
Let m1 and m2 be two interactions such as:

1. **A** ⇢ **M1 (i**.e., the antecedent of A lead to consequence of M1**)**
1. **B** ⇢ **M2 (**i.e., the antecedent of B lead to consequence of M2**)**
1. **M1 ∧ M2** → C (i.e., the antecedent of both M1 and M2 will eventually lead to the consequence of C)

So whenever (1) and (2) and (3) evaluates to true, then $\mathbf{A} \wedge \mathbf{B} \dashrightarrow \boldsymbol{C}$ is true as well

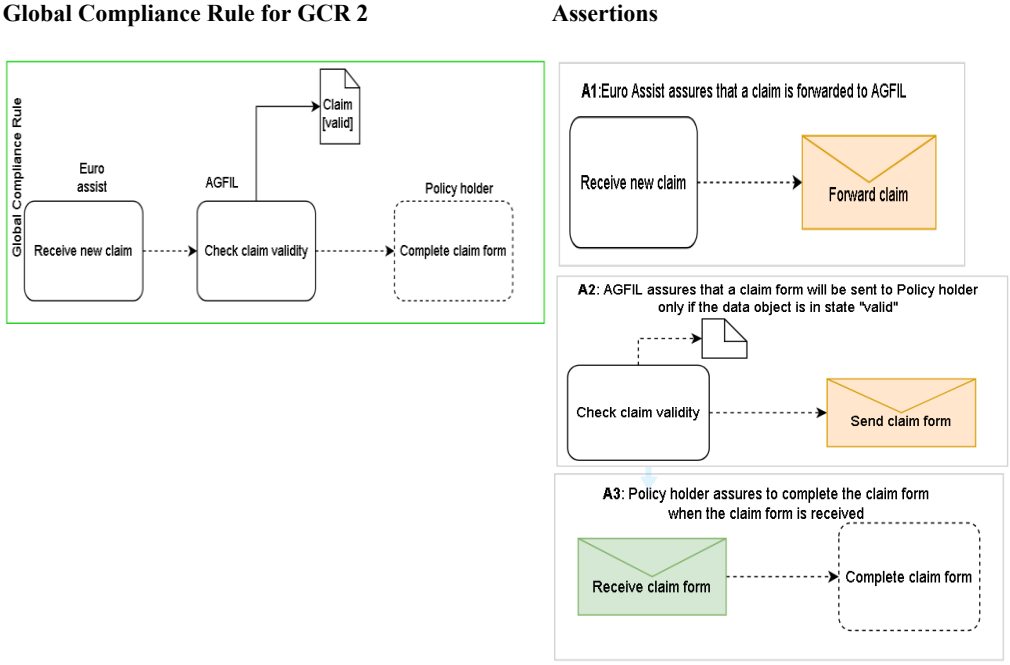
And for **Fig 3 (b),** the Rightwards transitivity holds such that:

$$\boldsymbol{A} \dashrightarrow \boldsymbol{B} \wedge \boldsymbol{B} \dashrightarrow \boldsymbol{C} \Rightarrow \boldsymbol{A} \dashrightarrow \mathbf{C}$$

4.2. GCR 2.

Expressing GCR 2 is also complicated because of the data condition associated with the rule. The rule refers to the choreography, public and private tasks of the partner as shown below:

GCR 2	**Choreography task**	**Private task**	**Public task**
Tasks/Activity	Send notification (see Fig. 2)	Check policy validity (see blue boxes in **Fig.1**)	Send claim form (see **Fig.1**)



Though, the GCR could be verified on the choreography model, but the condition also depends on the private activity of AGFIL, hence the reason to decompose the GCR. For this scenario, there is a message exchange between Euro Assist and AGFIL through m1, and the successful receiver of m1 will bring about the execution of B. when B is executed, we want to make sure that the state of the data object "Claim" will always be valid for m2 to be executed i.e., the if the data object is in state invalid then the process stops. Therefore, if C is executed, then B would have been executed before and the state of the data object "Claim" must always remain valid (see Fig 4). Once the decomposition in **Fig 4** is verified, then we can ensure the correctness of GCR 2.

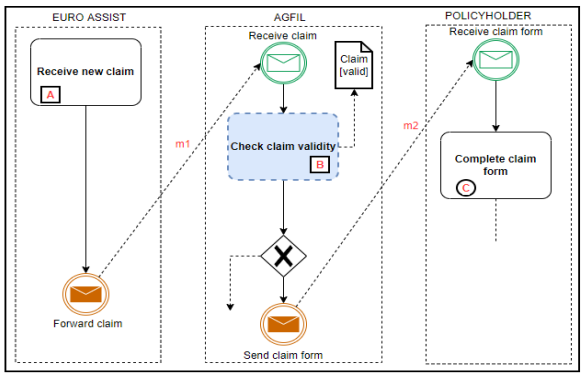


Fig 4. Decomposition scenario of GCR 2

4.2.1 The Applicability of the Decomposition Algorithm to GCR 2

Applying the algorithm in [21], the algorithm starts with the node **A** "Receive new claim" from Euro Assist and an assertion is created, them a connector is identified, and the algorithm cut the connector and create a new assertion for the node "check claim validity" for AGFIL. Expectedly, the algorithm will spot a connector to create a new assertion for Policyholder. However, this cannot happen as there is a data condition that needs to be satisfied before an assertion is created for Policyholder. As the data condition will determine the decision of the OR gateway. In case the data condition remains to be invalid as a result of the execution of **B**, no assertion is needed for the policyholder. However, when B is executed and the data object remains in the state "valid," then the algorithm can create an assertion for the node "Complete claim form" for policyholder. Hence, the algorithm is not applicable in such scenario. This scenario is common for a typical collaborative process. To fully support the compliance check of collaborative process, there is also a need to extend the approach in [5] and not just to support the structural compliance but also support the compliance patterns that deal with data flow and data conditions.
To prove a set of theorems that is required to ensure correctness of the decomposition method above, it is possible to apply the leftwards chaining Transitivity [5] to this scenario but with the data condition, such that the antecedent of A and B will eventually lead to C only when the data object is in state "valid".

$$A \rightarrow m1, m1 \rightarrow B \text{ and } B\ (Claim\ is\ in\ state\ "valid") \rightarrow m2, m2 \rightarrow C \Rightarrow A \dashrightarrow C$$
$$\forall a, b, c\ (a{=}b) \wedge (b{=}c) \Rightarrow a{=}c$$

Hence, there is a need to check whether data validation could be embedded into the different proofed theorems in [21] to support data condition and data flow in CBP. For instance, the Leftward chaining transitivity with data condition stated below:

Theorem 2 (Leftwards Chaining Transitivity with Data Condition)
Let A, B and C be activities or interactions such that A → B → C: if A and B occur, and B is in the state "valid," then C shall occur afterwards.
Let m1 and m2 be two interactions such as:

1) A → M1
2) M1 → B (and if data object is in state "valid")
3) B → M2
4) M2 → C

Whenever, (1) and (2) and (3) and (4) is true, then A → B ⇢ C is true as well.

It is worthy to note that messages interactions among partners in the collaboration could also represent a set of data object. For instance, in GCR 1, m1 and m2 are effectively data, the message exchange for m1 involves a completed claim form (data) being sent to AGFIL and m2 which include an invoice (which is also a data) sent to AGFIL. As a result, the compliance checking approach must be able to consider not just the message flow or interactions among partners but must consider messages as a valid data as well as the states the relevant data objects can adopt during the process

execution. Hence, for this current approach, there is an assumption that all messages are valid data. That is, we will treat all message interactions between partners as a valid data.
Based on the above two GCRs, the algorithm needs to be optimised for scenario 1 and improved for scenario 2. And in this instance, the research on commitment [2],[13],[20] and the use of BPMN-Q [6] could help to solve the identified gaps. It will further help to detect future violations as well as detecting any potential violator and provide the main cause of the violation. This is important because if something goes wrong, the whole business can become more complex. The high level of dependency as well as privacy issue, makes it quite difficult to identify who the potential violator is. For instance, for GCR 1, after Lee CS sent invoices, and the garage still does not get their payment. Based on the condition, AGFIL will pay the garage only after they receive the completed form from the policyholder and the invoices from Lee CS. With this, there is a great deal of potential violations from any of the partner i.e., Lee CS, Policyholder, AGFIL (refer to [1]).

5 Conclusion and Future works

This paper intends to review an existing compliance i.e., decomposition approach as described in [4],[5] to examine its applicability and complexity demonstrated using real-world collaborative case i.e., a car insurance case. We examine which compliance properties could be checked and cannot be checked yet using the approach. We further explore how the decomposition algorithm set out in [20],[21] could be applied to our case. Based on our analysis, compliance rule patterns that involves high level of dependency between more than two partners as well as the data flow pattern between partner processes, which could involve the private, public as well as the choreography model remains a challenge with this approach. And as a result, our future work plan to optimize their approach to support this limitation. This will help to add more complexity to the approach and meet up with the challenges of collaborative process identified in [1]. Lastly, we intend to use the application of the optimized algorithm to detect imminent violations and provide mechanisms for preventing violations as well as provide detailed feedback to end-user on the status of compliance.

Acknowledgments. This research is part of the FIRST project that has received funding from the European Union's Horizon 2020 research and innovation programme, the Marie Skłodowska-Curie grant agreement No. 734599.

References

1. Oyepeju, O., L. Xu. Compliance Checking of Collaborative Processes for Sustainable Collaborative Network, Working Conference on Virtual Enterprises. pp. 301-310. Springer. (2021).
2. Xu. L. A multi-party contract model. SIGecom Exchange. DOI: https://doi.org/10.1145/1120694.1120697. pp.13-23. (2004).
3. Oyepeju, O., L. Xu., Verification and Compliance in Collaborative Processes. In Boosting Collaborative Networks 4.0: 21st IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE 2020,.pp. 213-223. Springer, Valencia, Spain,(2020).
4. Fdhila, W., Rinderle-Ma, S., Knuplesch, D. and Reichert, M. Decomposition-based verification of global compliance in process choreographies. IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC) pp. 77-86. (2020).
5. Fdhila, W., Rinderle-Ma, S., Knuplesch, D. and Reichert, M. Verifying compliance in process choreographies: Foundations, algorithms, and implementation. 101983. (2022).
6. Awad, A., Weidlich, M. and Weske, M., Specification, verification, and explanation of violation for data-aware compliance rules. Service-oriented computing. pp.500- 515. Springer. (2009).
7. Knuplesch, D. and Reichert, M. A visual language for Modeling multiple perspectives of business process compliance rules. Software & Systems Modeling, 16 (3), pp.715-736. (2017).
8. Dwyer, M. B., Avrunin, G. S. and Corbett, J. C. Property specification patterns for finite-state verification, Proceedings of the second workshop on Formal methods in software practice. pp. 7-15. (1998).
9. Ramezani, E., Fahland, D. and van der Aalst, W. M. Where did I misbehave? Diagnostic information in compliance checking, international conference on business process management pp. 262-278. Springer. (2012).
10. Ramezani Taghiabadi, E., Fahland, D., Dongen, B. F. v. and van der Aalst, W. M. Diagnostic information for compliance checking of temporal compliance requirements, International Conference on Advanced Information Systems Engineering. pp. 304-320. Springer. (2013).
11. Turetken, O., Elgammal, A., van den Heuvel, W.-J. and Papazoglou, M. P. Capturing compliance requirements: A pattern-based approach. IEEE software, 29 (3), pp.28-36. (2012).
12. Ly, L. T., Rinderle-Ma, S. and Dadam, P. Design and verification of instantiable compliance rule graphs in process-aware information systems, International Conference on Advanced Information Systems Engineering. pp. 9-23. Springer. (2010).
13. 13. Montali, M. and Plebani, P. IoT-based compliance checking of multi-party business processes modeled with commitments, European Conference on Service-Oriented and Cloud Computing. pp. 179-195. Springer. (2017).
14. Semmelrodt, F., Knuplesch, D. and Reichert, M. Modeling the resource perspective of business process compliance rules with the extended

compliance rule graph. Enterprise, Business-Process, and Information Systems Modeling. pp.48-63. Springer. (2014).
15. Knuplesch, D. and Reichert, M. An operational semantics for the extended compliance rule graph language. (2017).
16. Knuplesch, D., Reichert, M. and Kumar, A. Towards visually monitoring multiple perspectives of business process compliance. (2015).
17. Knuplesch, D., Reichert, M., Ly, L. T., Kumar, A. and Rinderle-Ma, S. On the formal semantics of the extended compliance rule graph. (2013).
18. Knuplesch, D., Fdhila, W., Reichert, M. and Rinderle-Ma, S. Detecting the effects of changes on the compliance of cross-organizational business processes, International Conference on Conceptual Modeling. pp. 94-107. Springer. (2015).
19. Knuplesch, D., Reichert, M., Ly, L. T., Kumar, A. and Rinderle-Ma, S. Visual modeling of business process compliance rules with the support of multiple perspectives, international conference on conceptual modelling. pp. 106-120. Springer. (2013).
20. Telang, P. R. and Singh, M. P. J. I. T. o. S. C., 2011. Specifying and verifying cross organizational business models: An agent-oriented approach. 5 (3), pp.305-318. (2011).