Meeting the Challenges of Collaborative Network Compliance – An Exemplary View

Oyepeju Oyekola(✉), Lai Xu(✉), and Paul de Vrieze(✉)

Computing and Informatics, Bournemouth University, Poole BH12 5BB, Bournemouth, UK
{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, using a decomposition approach with eCRGs (extended Compliance Rule Graph) as a specification language. A real-world collaborative case is used to examine which compliance properties can be checked using 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, which served as a base for supporting the automated compliance checking of the 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 of 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 networks/processes present unique attributes such as the need to conform to security and privacy requirements, the need to conform to various regulatory requirements as a cross border organizations, as well as conforming to the constant changes in policies and regulations (e.g., COVID-19, BREXIT), presents a unique challenge.

Published by Springer Nature Switzerland AG 2022
L. M. Camarinha-Matos et al. (Eds.): PRO-VE 2022, IFIP AICT 662, pp. 406–419, 2022.
https://doi.org/10.1007/978-3-031-14844-6_33

While several compliance verification approaches have been addressed in the literature, these approaches still lack the support for the automated compliance checking of collaborative processes to the full extent. The compliance checking for the collaborative process should be designed to fully support the different process perspectives in terms of control flow, data flow, resource flow, and time perspective. 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 five partners adapted from [2], and a few of the key challenges is checking the compliance of processes involving a high level of dependency and response between each partner activity and data condition. As well as detecting the imminent violation of instance execution as well as the potential violators.

Generally, the Business Process Compliance lifecycle involves different compliance strategies: 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 [17]. The reality is that even if a business process has been checked during design time (before execution), there is no certainty that the corresponding running process instance will be compliant due to 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.

This research paper aims to support the compliance of the collaborative process 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 intends to follow the following process: (i) review an existing compliance approach for Collaborative processes i.e., decomposition approach [4, 5] using eCRG as a specification language. (ii) Use a real-world collaborative case to examine which compliance properties can be checked by the decomposition approach and which compliance properties cannot be checked yet, and (iii) explore how to extend the approach to meet up with the identified challenges of collaborative network compliance.

The rest of the paper is structured as follows: Sect. 2 describes the case description used in identifying some of the challenges of the collaborative process adapted from [2]. Section 3 explores and discusses the eCRG approach and its applicability to our use case. In Sect. 4, we examine which compliance properties can be checked by the decomposition approach and which compliance properties cannot be checked yet using the use case. Lastly, Sect. 5 gives the summary of the paper, and we highlight the challenges of collaborative networks as well as our future research.

2 Case Description

A motivating use case of a collaborative process involving five partners is adapted from [2] depicted in Fig. 1. The process starts with the insurance policyholder who reports a claim in case of any damage to the issued car. Euro Assist is the company that receives the report from the policyholder via the telephone, registers the claim received, and

encourages approved garages. Euro Assist sends the claim to AGFIL which 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 conducted quickly and cost-effectively.

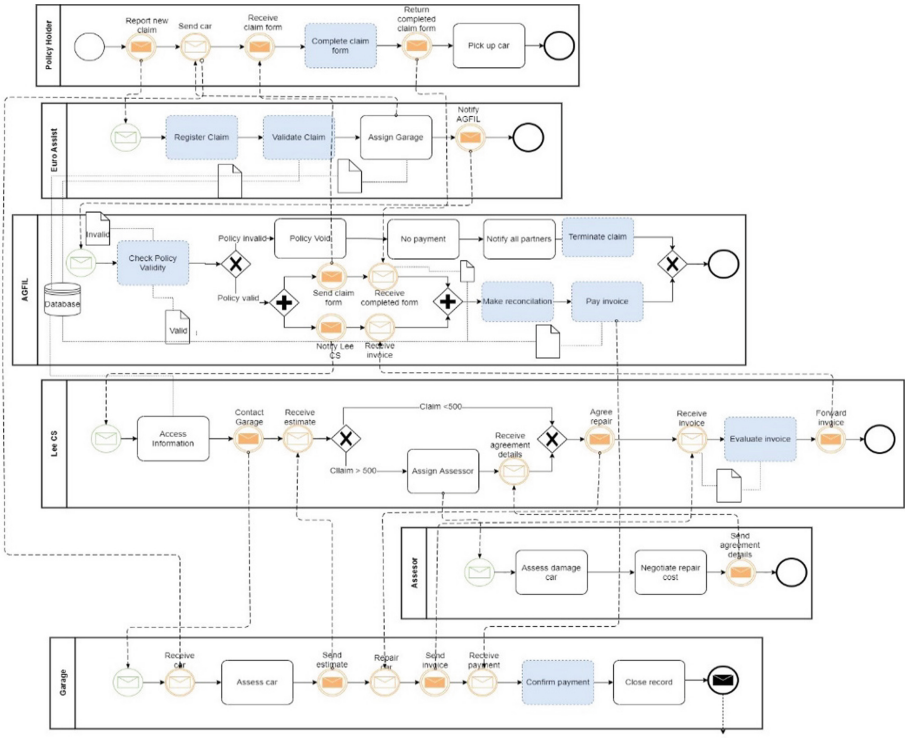


Fig. 1. Collaborative model for insurance case [1, 2]. (Color figure online)

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 the 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 [8–11] 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 to compliance rules is flexible and aids comprehensibility for domain experts [7]. However, it is worth noting that most of these existing visual languages do lack the full support of all the various process perspectives as it solely focuses on the specification of the control flow perspective and an aspect of the data flow. For instance, CRG supports solely the control flow perspectives while BPMN-Q supports control flow and data conditions.

To support the specification of the different process perspectives, [7, 15] presents an extended Compliance Rule Graph (eCRG) i.e., an extension of CRG to visually model compliance rules 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 precondition 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) [7]. The eCRG semantics is formally specified through a translation of eCRGs into FOL (First Order Logic) expressions based on completed process logs. The feasibility of eCRG was scientifically evaluated in [15] using a different approach such as a proof-of-concept prototype, empirical evaluations, its applicability to real-world cases, as well a 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.

3.1 Collaborative Business Process Models in eCRG

Since the focus of this study is on Collaborative Processes (CP), then this section reviews a few works of CP that based their language specification on eCRG. Supporting cross-organizational processes involving multiple partners concerning GCR (Global Compliance rule) is addressed in [14] 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 CPs. The algorithm developed was used to detect the impact of CP changes on GCR. In [4], the authors describe how the global compliance rule of process choreographies could be verified in a decentralized manner by each partner in the process collaboration and deal with the restricted visibility of process activity. The approach uses a decomposition-based approach i.e., the decomposition of GCR into a set of an 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 the 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. Due to the feasibility of their approach (i.e., support for privacy requirements among partner processes and the language specification), this paper explores how the decomposition approach could be applied to more complex CPs.

3.2 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 a situation, each partner's private activities remain invisible to the 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 is split into a set of assertions that are checked locally by each partner and 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 the collaborative process. In general, the 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 blue boxes depicted in Fig. 1). The process model is subject to various global compliance rules that stem 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	The garage must receive and confirm payment from AGFIL within a specific period	AGFIL makes reconciliation and 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 the 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 the state "Valid" or otherwise the process end

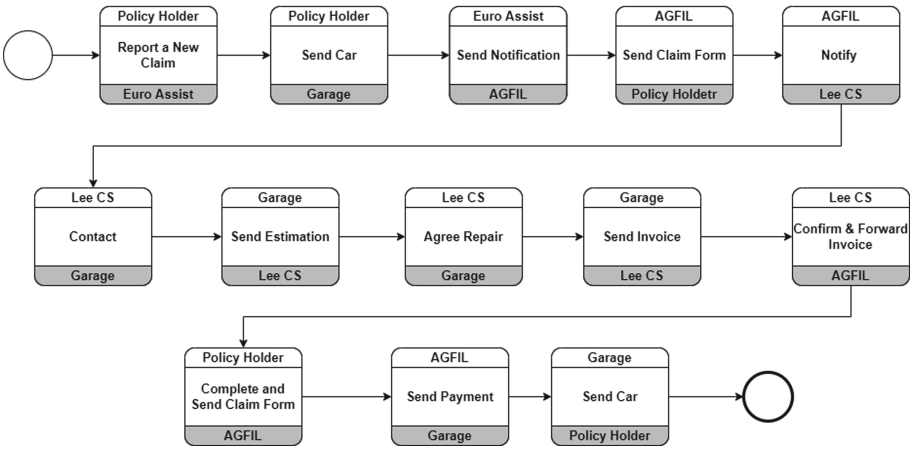


Fig. 2. Choreography model for insurance case

4.1 GCR 1

GCR 1 as shown in Table 1, involved the choreography, public as well as private tasks/activity of the partners involved in the GCR as Table 2.

Table 2. Tasks/activities of partners involved in GCR 1

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

For instance, in Fig. 1, the private task "pay Invoice" is invisible to the other partner and cannot have an idea of when the task is completed or not. However, there are a few complexities as regards this GCR as it involves some conditions that also need 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 with 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.

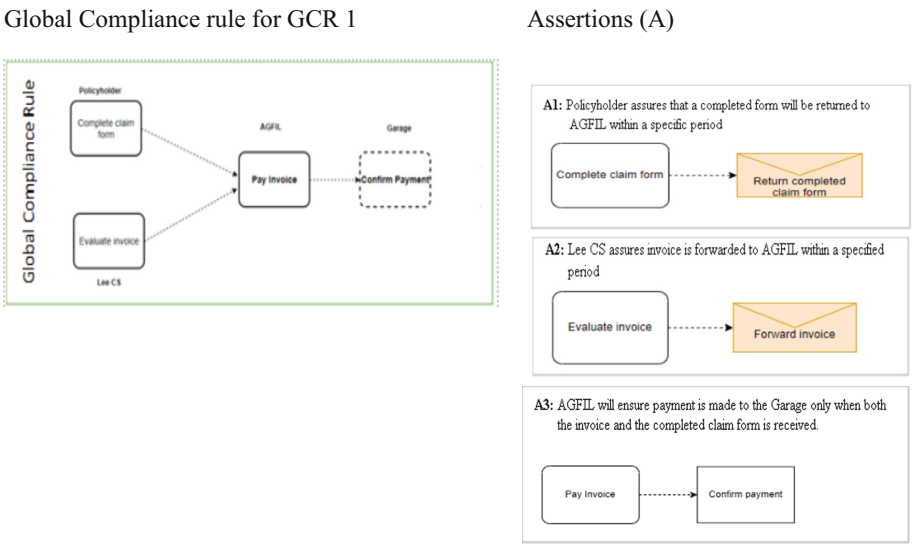
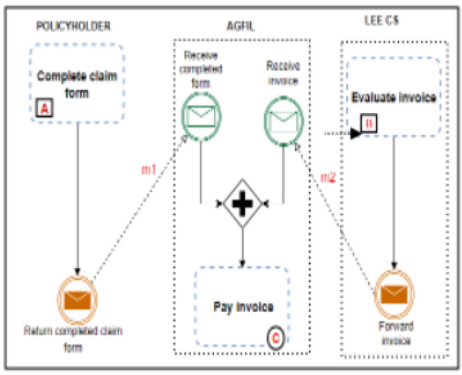


Fig. 3. GCR 1 and its Assertion

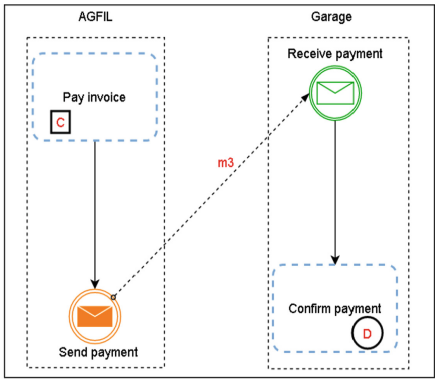
To decompose GCR 1 into assertion as shown in Fig. 3, Policyholder assures that a completed form will be returned to AGFIL (A1), Lee CS assures that the invoice will

be forwarded to AGFIL (A2) and, AGFIL assures payment is made to the Garage upon receiving both the invoice and completed claim form (A3).

We illustrate the decomposition process of GCR 1 using two scenarios to ensure simplicity and readability depicted in Fig. 4. The decomposition of GCR1 includes (a) the sub-conditions that needed to be satisfied first and (b) the rule that needed to be satisfied afterward.



(a): Decomposition scenario of GCR condition for GCR1



(b): Decomposition scenario of GCR 1

Fig. 4. 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 are successfully received by AGFIL.

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

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) explains 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.

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

This 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. 4 is verified, then we can ensure the correctness of GCR 1, that is:

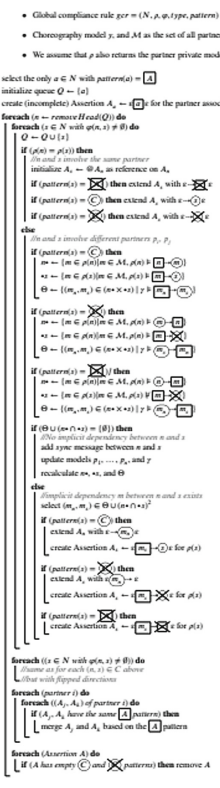
$$\mathbf{A \wedge B \rightarrow C \text{ and } C \rightarrow D \Rightarrow A \wedge B \rightarrow D}$$

4.1.1 The Applicability of the Decomposition Algorithm to GCR 1

This section analyses the capabilities of the algorithm presented as Algorithm 1 in [5], in the Car Insurance Case. We aim to check the applicability and complexity of the algorithm by analyzing whether the algorithm can handle a complex collaborative process with a high level of dependency on the partner's process and data conditions.

Table 3 depicts the application of the algorithm in [5] to GCR 1 to derive the assertions using transitivity is shown below:

Table 3. Application of the algorithm to derive assertion for GCR 1

Algorithm [5]	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 starts 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 cut 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 the policyholder with the 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 connection 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. 4 using the transitivity relationship in [5]**,** 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

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. 4, we realized 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 GCR 1. Hence, we propose the need to extend the algorithm and provide formal proof for the transitivity relationship. Such that if the decomposition in Fig. 4 is verified to be true, then we can ensure the correctness of GCR 1. That is:

$$\mathbf{A} \wedge \mathbf{B} \rightarrow C \text{ eventually lead to the execution of } \mathbf{D}$$

Does the successful execution of $\mathbf{A} \wedge \mathbf{B} \rightarrow \boldsymbol{C}$ will eventually lead to the execution of **D** always? The answer here is No. For example, if the claim form by the Policyholder and the Invoice by Lee CS is successfully received by AGFIL but AGFIL forgot to send payment to Garage within the specified period**.** Or when neither of the partners fulfills their obligations. In such an instance, there is a need to provide a solution that can detect: (1) what is expected from a partner in the future (2) which partner must be reminded of their obligation (3) which partner is the potential violator.

4.2 GCR 2

Expressing GCR 2 (see Table 2) is also complicated because of the data condition associated with the rule. The rule refers to the choreography, and public and private tasks of the partner as shown in Table 4:

Table 4. Tasks/activities of partners involved in GCR 2

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

Though the GCR could be verified on the choreography model, the condition also depends on the private activity of AGFIL, hence the reason to decompose the GCR. To decompose GCR 2 into assertion as shown in Fig. 5, Euro Assist assures that a claim will be forwarded to AGFIL (A1), and AGFIL assures that a claim form will be sent to the policyholder only if the data object is in state valid (A2) and, the policyholder assures that the claim form will be completed (A3).

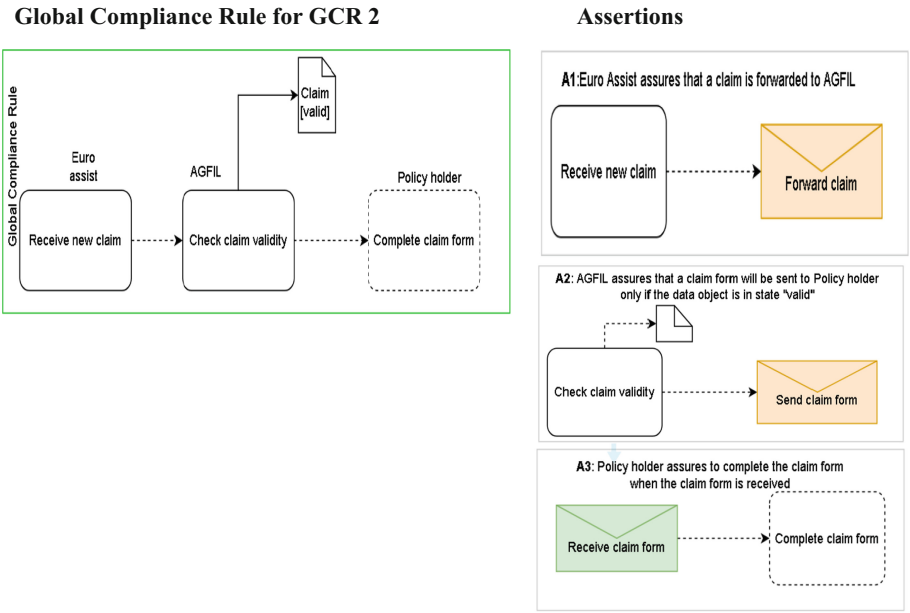


Fig. 5. GCR 2 and its Assertion

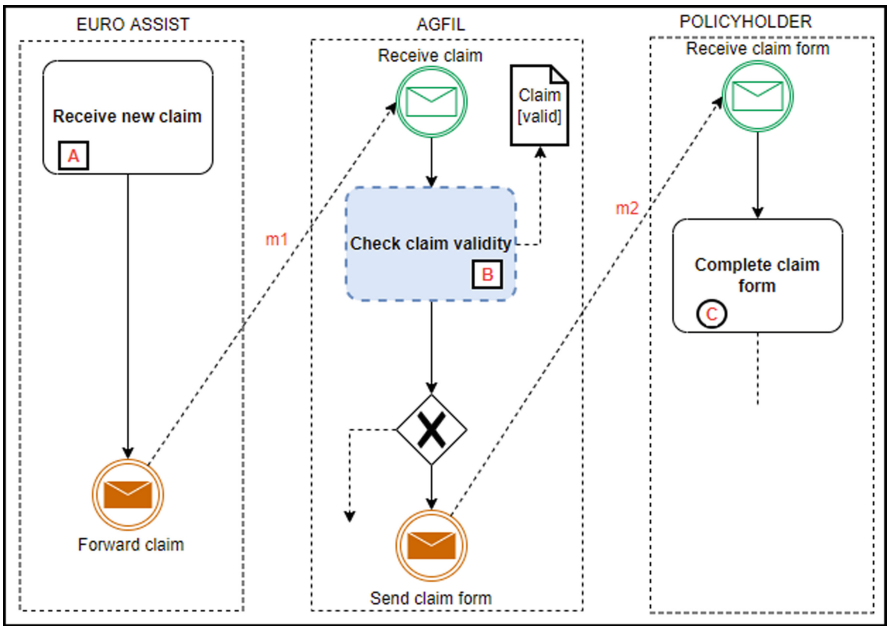


Fig. 6. Decomposition scenario of GCR 2

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. 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. 6). Once the decomposition in Fig. 6 is verified, then we can ensure the correctness of GCR 2.

4.2.1 The Applicability of the Decomposition Algorithm to GCR 2

Applying the algorithm in [5], 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 creates a new assertion for the node "check claim validity" for AGFIL. Expectedly, the algorithm will spot a connector to create a new assertion for the Policyholder. However, this cannot happen as there is a data condition that needs to be satisfied before an assertion is created for the Policyholder. As the data condition will determine the decision of the OR gateway. In case the data condition remains to be invalid because 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 the policyholder. Hence, the algorithm is not applicable in such a scenario. This scenario is common for a typical collaborative process. To fully support the compliance check of a collaborative process, there is also a need to extend the approach in [5] to support the compliance patterns that deal with data flow and data conditions.

To prove a set of theorems that is required to ensure the 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 the state "valid".

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

Hence, there is a need to check whether data validation could be embedded into the different proofed theorems in [5] to support data conditions and data flow in CP.

Overall, it is worthy to note that message interactions among partners in the collaboration could also represent a set of data objects. 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 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 valid data.

5 Conclusion and Future Works

This paper intends to review existing compliance i.e., the decomposition approach as described in [4, 5] to examine its applicability and complexity demonstrated using a

real-world collaborative case i.e., a car insurance case. We examine which compliance properties could be checked and which cannot be checked yet. We further explore how the decomposition algorithm set out in [4], and [5] could be applied to our case. Based on our analysis, compliance rule patterns that involve a high level of dependency between more than two partners as well as the data flow pattern between partner processes, which could involve the private model remains a challenge with this approach. And as a result, our future work plan to optimize their approach to support this limitation. And in this instance, the research on commitment [2, 13, 16] and the use of BPMN-Q [6] could be embedded in this approach to help solve the identified gaps. Lastly, we intend to use the application of the optimized algorithm to further explore how to detect future violations as well as to detect any potential violator and provide the main cause of the violation. This is important because if something goes wrong, the whole business process can become more complex. This will help to add more complexity to the approach and meet up with the challenges of a collaborative process.

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. Oyekola, O., Xu, L., de Vrieze, P.: Compliance checking of collaborative processes for sustainable collaborative network. In: Camarinha-Matos, L.M., Boucher, X., Afsarmanesh, H. (eds.) PRO-VE 2021. IAICT, vol. 629, pp. 301–310. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85969-5_27
2. Xu, L.: A multi-party contract model. SIGecom Exchange 13–23 (2004). https://doi.org/10.1145/1120694.1120697
3. Oyekola, O., Xu, L.: Verification and compliance in collaborative processes. In: Camarinha-Matos, L.M., Afsarmanesh, H., Ortiz, A. (eds.) PRO-VE 2020. IAICT, vol. 598, pp. 213–223. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62412-5_18
4. Fdhila, W., Rinderle-Ma, S., Knuplesch, D., Reichert, M.: Decomposition-based verification of global compliance in process choreographies. In: IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC), pp. 77–86 (2020)
5. Fdhila, W., Rinderle-Ma, S., Knuplesch, D., Reichert, M.: Verifying compliance in process choreographies: foundations, algorithms, and implementation. 101983 (2022)
6. Awad, A., Weidlich, M., Weske, M.: Specification, verification and explanation of violation for data aware compliance rules. In: Baresi, L., Chi, C.-H., Suzuki, J. (eds.) ICSOC/ServiceWave -2009. LNCS, vol. 5900, pp. 500–515. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10383-4_37
7. Knuplesch, D., Reichert, M.: A visual language for modeling multiple perspectives of business process compliance rules. Softw. Syst. Model. **16**(3), 715–736 (2017)
8. Dwyer, M.B., Avrunin, G.S., Corbett, J.C.: Property specification patterns for finite-state verification.In: Proceedings of the Second Workshop on Formal Methods in Software Practice, pp. 7–15 (1998)
9. Ramezani, E., Fahland, D., van der Aalst, W.M.P.: Where did I misbehave? Diagnostic information in compliance checking. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 262–278. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32885-5_21

10. Ramezani Taghiabadi, E., Fahland, D., van Dongen, B.F., van der Aalst, W.M.P.: Diagnostic information for compliance checking of temporal compliance requirements. In: Salinesi, C., Norrie, M.C., Pastor, Ó. (eds.) CAiSE 2013. LNCS, vol. 7908, pp. 304–320. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38709-8_20
11. Turetken, O., Elgammal, A., van den Heuvel, W.-J., Papazoglou, M.P.: Capturing compliance requirements: a pattern-based approach. IEEE Softw. **29**(3), 28–36 (2012)
12. Ly, L.T., Rinderle-Ma, S., Dadam, P.: Design and verification of instantiable compliance rule graphs in process-aware information systems. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 9–23. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13094-6_3
13. Montali, M., Plebani, P.: IoT-based compliance checking of multi-party business processes modeled with commitments. In: De Paoli, F., Schulte, S., Broch Johnsen, E. (eds.) ESOCC 2017. LNCS, vol. 10465, pp. 179–195. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67262-5_14
14. Knuplesch, D., Fdhila, W., Reichert, M., Rinderle-Ma, S.: Detecting the effects of changes on the compliance of cross-organizational business processes. In: Johannesson, P., Lee, M.L., Liddle, S.W., Opdahl, A.L., López, Ó.P. (eds.) ER 2015. LNCS, vol. 9381, pp. 94–107. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25264-3_7
15. Knuplesch, D., Reichert, M., Ly, L.T., Kumar, A., Rinderle-Ma, S.: Visual modeling of business process compliance rules with the support of multiple perspectives. In: Ng, W., Storey, V.C., Trujillo, J.C. (eds.) ER 2013. LNCS, vol. 8217, pp. 106–120. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41924-9_10
16. Telang, P.R., Singh, M.P.: Specifying and verifying cross-organizational business models: an agent-oriented approach. IEEE Trans. Serv. Comput. **5**(3), 305–318 (2012). https://doi.org/10.1109/TSC.2011.4
17. Ly, L.T., Rinderle-Ma, S., Knuplesch, D., Dadam, P.: Monitoring business process compliance using compliance rule graphs. In: Meersman, R., et al. (eds.) OTM 2011. LNCS, vol. 7044, pp. 82–99. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25109-2_7