Verifying for Compliance to Data Constraints in Collaborative Business Processes

John Paul Kasse, Lai Xu, Paul Devrieze, Yuewei Bai

Verifying for Compliance to Data Constraints in Collaborative Business Processes

John Paul Kasse[1], Lai Xu[1], Paul deVrieze[1] and Yuewei Bai[2]

[1]Department of Computing and Informatics, Faculty of Science and Technology, Bournemouth University, Poole BH12 5BB, United Kingdom
{jkasse, lxu, pdvrieze}@bournemouth.ac.uk
[2]Industry Engineering of Engineering College, Shanghai Polytehnic University
ywbai@sspu.edu.cn

Abstract. Production processes are nowadays fragmented across different companies and organized in global collaborative networks. This is the result of the first wave of globalization that, among the various factors, was enabled by the diffusion of Internet-based Information and Communication Technologies (ICTs) at the beginning of the years 2000. The recent wave of new technologies possibly leading to the fourth industrial revolution – the so-called Industry 4.0 – is further multiplying opportunities. Accessing global customers opens great opportunities for organizations, including small and medium enterprises (SMEs), but it requires the ability to adapt to different requirements and conditions, volatile demand patterns and fast-changing technologies. Regardless of the industrial sector, the processes used in an organization must be compliant to rules, standards, laws and regulations. Non-compliance subjects enterprises to litigation and financial fines. Thus, compliance verification is a major concern, not only to keep pace with changing regulations but also to address the rising concerns of security, product and service quality and data privacy. The software, in particular process automation, used must be designed accordingly. In relation to process management, we propose a new way to pro-actively check the compliance of current running business processes using Descriptive Logic and Linear Temporal Logic to describe the constraints related to data. Related algorithms are presented to detect the potential violations.

Keywords: Compliance, Collaborative Business Processes, Virtual Factory, Business Process Verification, Algorithm.

1 Introduction

Compliance is about adherence to regulations, guidelines or predefined legal requirements like norms, laws and standards. Compliance verification in business process management is addressed at different levels of the life cycle i.e. deign time, runtime, post runtime. A hybrid approach addresses compliance verification for all levels [1]. Moreover, existing research has made significant contribution to addressing verification of models for compliance with a range of requirements such as, activity ordering requirements [2]–[9], resource assignment constraints [10]–[13], data requirements [14], [15], security requirements [16]–[20] and privacy [21], [21]–

[27], compliance between process variants [28]–[31]. These works show the state of the art in business processes compliance management and verification. They have also resulted into various compliance approaches, frameworks, methods, languages and tools. However, more compliance challenges need to be addressed to fully support collaborative processes in the context of a virtual factory.

In this paper, we look at the compliance of running collaborative business processes with data constraints. The paper proposes a new way to describe data constraints using descriptive logic (DL) and Linear Temporal Logic (LTL). The traces of the running processes are used to check whether the current collaborative business processes are compliant with the data constraints described in DL and LTL.
The structure of the paper is as follows: related work is presented in Section 2. Section 3 introduces an exemplary business process and related traces. DL and LTL are used to express the data constraints in Section 4. Section 5 shows how to present compliance properties as well as related verification algorithms. Future work and conclusion are presented in Section 6.

2 Related Work

Pesic et al. propose DECLARE, a declarative constraint based specification language and model compliance checking in relation to ordering requirements [2], [3]. The language is limited to control flow checking. Similar work is presented by Awad et al. and Wynn et al. Awad et al, propose a BPMN-Q language which extends BPMN to search for segments of a process model affected by changes and verify their compliancy in terms of control flow.

Whereas Wynn extends YAWL language with reset nets to determine correctness of business processes with cancellation and OR joins, other work by Elgammal et al. and Taghiabadi present compliance frameworks for managing the compliance of a business process during its life cycle. The constraints are organised according to patterns like control flow, resources, temporal and data [7], [9]. Despite the fact that the frameworks comprehensively cover all perspectivies of the business process, the proposed languages employ complex mathematics and logics that are not intuitive for ordinary end users.

Data specific constraint checking approaches check compliance between the model and data requirements. Knuplesch et al. propose a graph method for modelling compliance rules and address verification through structured compliance checking based on compliance rules and data checking based on abstracted data [14] The resultant graph based approach is data constraint based. Borrego and Barbara enhance earlier work of Declare to include data requirements compliance checking [15]. In this paper we extend our earlier work for supporting compliance verification in collbaorative business processes [32]–[34]. Related work remains limited in various ways; the compliance management framework by Elgammal et al results into a compliance request language in which constraints can be specified by ignores their verification. Taghiabadi' s compliance approach caters for verification for control flow and data constraints. However, its application to collaborative environments is not demonstrated, it is also a domain specific approach and so is Declare language.

Wynn et al's work based on YAWL is geared towards control flow verification to achieve model soundness. Data constraints are not considered by the authors. Moreover, non of these works presents mechanism easily comprehensible for non-expert end users. This limits their application. Our work leverages previous work by supporting specification of data constraints and verifying for their compliancy with collaborative business processes using an approach that empolys syntactic and semantic mechanism close to natural language. Besides, we provide a coarse grained approach in which we cater for data constraints in terms of accessibility, authentication and privacy by means of access control and authorisation. This is a valuable contribution in the wake of revised compliance requirements of the 2008 general data protection regulation.

3 An Exemplary Business Process and Related Traces

We adopt an abstracted industry based use case, the Pick and Pack business process proposed in [33]. In this case, customers submit orders online after registering on the system. Stores' staffs check order details, and proceed to process the order as Fig. 1 illustrates.

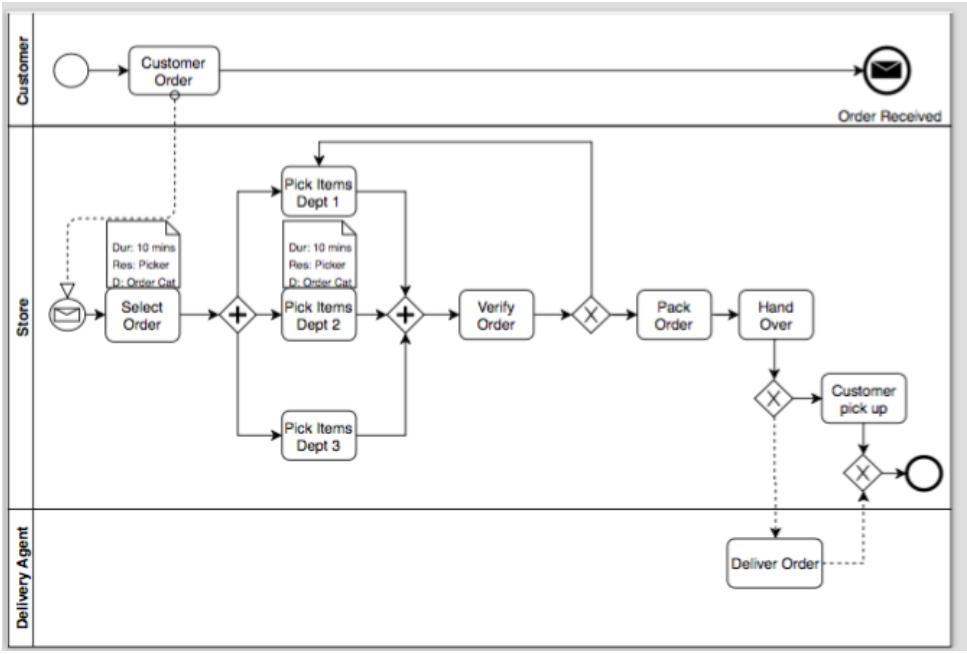

Fig1. Pick and Pack business process

Order processing involves activities summarised in Table 1 and assigned roles: Based on the role assignments, the following resource assignment conditions apply;

1. Pickers cannot participate in the verification of orders.
2. Packers can also do the verify order task.
3. Pickers can participate in hand over task at peak times
4. Supervisors oversee other employees and can execute any task.

Supervisors delegate or share rights of task execution to other staff, e.g. supervisors can delegate pickers to pack items.

Table 1. Process activities and role assignments

Activity	Description	Role
Select Order	Order is chosen from a pool of pending orders.	Picker
Pick items	Order items are picked	Picker
Verify order	Right order in terms of items and quantities	Verifiers
Pack order	Order is packed	Packers
Hand over / Deliver order	Orders ready for picked up or delivery by agent	Customer service

Relatedly, the constraints governing data access are summarized;

1. Supervisors have full data access and can grant access to other staff.
2. Basic data must be accessible and available for staff to execute tasks that do not need much restriction and control. E.g. order list data.
3. Access control and authorization is observed for restricted data. For example, customer personal data, financial data among others.
4. For security of data and system, users must be authenticated by the system.

For illustration purposes, the business process is considered in terms of activities while traces are used as a mechanism to derive process execution to facilitate checking compliance to constraints. Table 2 lists sample traces based on the use case.

Table 2. Exemplified log showing events, activities, constraints and process instances

Event	Activity	Constraints		Process instance
		Accessible Data	Time (units)	
e_1	Select order	OrderList	Duration [6]	P_1
e_2	Hand over order	Customer orderlist	Delay[10]	P_1
e_3	Select order	Orderlist	Between[10]	P_1
e_4	Pick items	ProductList	Duration[15]	P_1
e_5	Pickup or delivery	Contactlist	Duration[20]	P_1
e_6	Select order	Customer address	Duration[10]	P_1

4 Constraint Expression in Description Logic and LTL

To achieve constraint expression, descriptive logic (DL) [35] and LTL are adopted. While using DL, the business process is the domain of discourse with activities and

constraints as concepts. The intention is to support expression of constraint requirements in a way close to natural language for easy intuition by non-experts yet expressive enough to support reasoning. DL is known for knowledge base representation and building in knowledge management systems [35]. Its application in business process design however has not received much attention with fewer applications in [11], [36]. The limitation is due to lack of known syntax to express unique business process requirements.

To enhance its expressiveness, we adopt logical operators and quantifier operators from temporal logic. Temporal logic is a formal method founded on mathematics. Models are specified and checked for correctness against a set of properties expressed as event orderings in time [37]. Role representation establishes a link between the constraints and the activities while the role restrictions impose specific existential and value restrictions of a constraint over the activity. We use unary predicates to represent sets of individual constraints while binary predicates denote relationships between individual constraints. We further use composite predicates to denote relationships between different constraints.

4.1 Expressions of Data Requirements as Constraints in DL and LTL

Data constraints are based on data patterns [38] and represented as unary predicate expressions using DL and logical operators and quantifiers from LTL. Task execution requires access to data. E.g., 'delivery' task needs access to customer address and contact to be accessible. Further still, actions that a user can do must be pre-authorized i.e. action to read, write, modify. Thus, task data assignment TD is composed of a task, data object (o), value (v) and action (∂). TDassignment is achieved by a function $f: a \rightarrow o, v, \partial$ which maps data item attributes like value and action to a task. Fig. 2 exemplifies task data assignment and the required attributes.

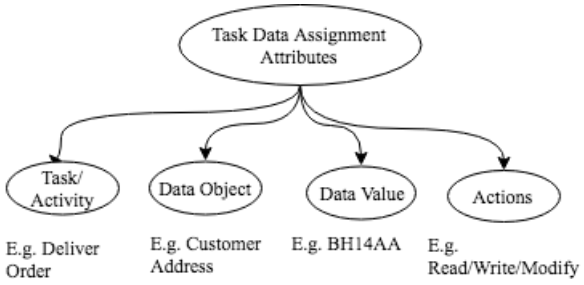

Fig. 2 Task and data assignment attributes

Using DL constructs, the expression for TD is derived as; $TD = a \rightarrow (o \sqcap v \sqcap \partial)$ and formalized as; $a \rightarrow (o \wedge v \wedge \partial)$ in LTL. Based on Fig.2, the use case, it follows that;

$$TD = [Deliver_{Order} \rightarrow (customer\_address \sqcap BH14AA \sqcap Read)]$$

Table 3 presents some examples of expressions as derived for data constraints in;

Table 3. Data Constant Expression in DL

DL data unary expressions	*Description*
∀data → visible /∃data → visible	All/ some data is visible
∀data → valid /∃data → valid	All /some is valid data
∀data → ¬available /∃data → ¬available	All /some data not available
∀data → accessible /∃data → accessible	All /some data is accessible
∀data → privacy /∃data → privacy	All /some data classified as private
∀data → authentic /∃data → authentic	All /some data classified as private
DL data binary expressions	***Description***
∀data → (visible ⊓ accessible) /∃data → (visible ⊓ accessible)	All/ some data has visible and accessible constraints
∀data → (authentic ⊓ privacy) /∃data → (authentic ⊓ privacy)	All /some has authenticity and privacy constraints

5 Verification Algorithms for Compliance to Data Constraints

Data constraints are based on Boolean conditional evaluations where the condition is either true or false. Depending on the outcome of the conditional evaluation assessed against predefined access policies, access is granted or denied. To check for compliance, a constraint is satisfied if a trace is true for the given conditions, and the constraint is violated if trace is otherwise. To that effect, the following specifications and definitions are useful for the data constraints compliance checking algorithm.

Given a set of activities a_1, a_2 and a_3 whose execution by role actor (r_1) requires product catalogue data (Pcd). Access to this data is constrained by access and availability. If the assignment is true per the executed behavior, then the trace (σ) satisfies ($\vDash$) the constraint.

Definition 1: Accessibility and Availability (AA)

$$\sigma \in (((a_1, a_2, a_3), r_1) : (Pcd.[Read]) : AA \quad (1)$$

If $\sigma = True$ **then** $\sigma \vDash AA$

The definition specifies accessibility and availability constraints for Pcd data object with action read granted to r_1 for execution of activities a_1, a_2 and a_3. During verification, the data compliance verification algorithm checks for compliance to the constraint for the data object, action by the user and tasks. If the outcome shows that the trace is true, then the availability and accessibility constraint is satisfied. Otherwise, it is a violation detected for the AA constraint.

Definition 2: Authentication

$$\sigma \in (((a_1, a_2, a_3), r_1) : (Pcd.[True/False]) : Authentication \quad (2)$$

If $\sigma = True$) **then** $\sigma \vDash Authentication$

The definition specifies access control by authentication granted for Pcd data with actions to read and write for role actor (r_1) who executes activities a_1, a_2 and a_3.

Satisfaction of the authentication constraint is achieved if the trace of the executed events exhibits the specified behavior. Otherwise, a violation is detected for the constraint.

Definition 3: Privacy (Prv)

$$\sigma \in (((a_1, a_2, a_3), r_1): (Pcd. [Read]): Prv \tag{3}$$

If $\sigma = True$) **then** $\sigma \vDash Prv$

The definition specifies Privacy constraint for accessing Pcd data where action to read private data is to be granted for the resource actor r_1 who executes activities a_1, a_2 and a_3. During verification using the privacy compliance verification algorithm, the constraint is checked if it is satisfied before access is granted to private data. If trace is true to the specification, then the constraint is satisfied and thus compliance achieved. Otherwise, it is a violation detected for the privacy constraint.

5.1 Data Constraints Verification Algorithms

In this section, four algorithms are introduced, namely access and availability, authentication and privacy data constraints compliancy verification algorithms. At the end, an overall data constraint compliancy verification algorithm is presented.

Access and Availability Constraint Verification

Verifying Access and Availability data constraint ensures that basic non-exclusive data is accessible and available with less restriction to enable accomplishment of basic tasks. Algorithm 1 is composed to the effect. Violation occurs if role actors or tasks are denied access to data or where the permitted action type differs from the initial assignment, e.g. modify action type instead of read action type.

Algorithm 1 Access and Availability Compliance Verification

1: $InPut$:

a. All Process Instances, Traces and Activities events

b. Constraints (AA)

2: **for** all data with constraint $C = (AA : [Read/Write/Modify])$ for actors (r) **do**

Assign = r,e.ac: $\rightarrow$ AA= Data Item.[Read/Write/Modify]

3: **if** $(Assign \neq \in seen, finished \neq AA)$ **then**

Violation: "Deadlock due to denied access to data. AA constraint violated"

Return No violation of AA constraint for the provided processes if $AA \in$ seen and $AA \in$ finished

Violation of AA constraint as per Algorithm 1 exists when tasks or their actors (*r, e.ac*) are denied access. The violation results into to a deadlock or livelock. Deadlock occurs if running activities are denied access to data necessary for the process to progress, while livelock occurs when a task denied data access stays in waiting mode

stagnating process execution. Another form of violation occurs when a task executes without necessary data resulting into wrong outcomes which compromise data integrity.

Verifying Compliancy with Authentication Data Constraint

Authentication Algorithm 2 verifies for compliance by checking that role actor credentials match the credentials stored in a database of authorized actors and their access privileges over tasks. Two forms of authentication errors lead to violations, i.e.;

- Access leakage which occurs when non-authenticated users gain access to data. This is traced from running or finished events
- Deadlocks which occur when users are authorised to execute activities but access to data is denied for technical or logical reasons e.g. improper configurations.

Algorithm 2 Authenticity Data Constraint Checking

1: $InPut$:

a. All Process Instances in the business process

b. Constraints (Authenticity)

2: **for** all data with constraint C.Auth = Data item.$[True/False]$) **do**

$assign \equiv r, e.ac :\rightarrow Auth = DataItem.[True/False]$

3: **if** $(Assign \neq\in seen, finished)$ **then**

Violation: "authenticated are denied access to restricted data."

4: **if** ∃ actor $r_n \in Assign$ **then**

Violation: "Access leakage, non-authenticated actor accessed data. "

Return No violation of Authenticity constraint for the provided business process.

When data constrained by authenticity constraint exists outside the constraint it leads to access leakage since it will be accessible by users without authentication or if it is accessed by non-authenticated role actors. Similarly, where data is not accessible to authenticated actors leads to a deadlock since they cannot progress with the current work being executed.

Verifying Compliancy with Privacy Data Constraints

Privacy constraint is enforced by means of access control and authorization. Authorization involves the process of validating that the authenticated user is granted permission to access the requested resources. Privacy as a data constraint restricts access to data regarded private as defined by GDPR. Data restricted from public access is enforced by authorization. Algorithm 3 checks whether the process is complying with the privacy data constraint. Violation to privacy constraint is checked targeting two forms of errors; deadlocks and privacy breach.

- Deadlocks occur when the executing events authorised to access data are denied access for technical or logical reasons e.g. improper configurations,
- Breach to privacy i.e. non-authorised activities eventually access private data and execute.

Algorithm 3 Privacy Data Constraint Checking

1: $InPut$:

a. All Pi in the Business process

b. Constraints (Privacy)

2: **for** all data with constraint (C=Privacy[R/W/M]) for actors (r) **do**

Assign $\equiv$ r,e.ac: Data Item $\rightarrow$ privacy $\equiv$ Authorise :[Read/Write/Modify]

3: **if** $(Assign \nvDash privacy \in seen, finished)$ **then**

Violation: "Assigned actors denied access to private data"

4: **if** $(Assign \neq (Dataclerk, Assessor) \in seen, finished)$ **then**

Violation: "Access leakage, non authorised actors gain access to private data"

Return No violation of Privacy constraint for the provided processes if $r, e.ac \in$ seen and finished

When data constrained by privacy constraint exists outside the constraint, it leads to a leakage since it is accessible by non-authorized actors. Similarly, where authorized data is not visible in 'seen' and 'finished', it implies denied access as a form of violation.

The overall compliance verification algorithm 4, is a general algorithm that invokes algorithms 1, 2 and 3 to check whether the entire business process complies with above mentioned data constraints.

Algorithm 4 Overall Compliance Data Constraint Verification Algorithm

1: $InPut$:

a. All Process Instances in the business process

b. All Constraints

2: **for** all (Pi) with given constraints C = Control flow, Resource, Data, and Temporal constraints) **do**

Return violation or compliance of Resource flow constraints /* Check for compliance with Data constraints*/

3: **if** e.ac.Exist = True **then**

Check AA $\rightarrow$ call algorithm 1

Check Auth $\rightarrow$ call algorithm 2

Check Privacy $\rightarrow$ call algorithm 3

Return violation or compliance of Data constraints

Return overall compliancy or violation of business process with verified constraints.

6 Conclusion and Future Work

Regardless of the industrial sector, compliance is a major concern not only to keep pace with changing regulations but to address the rising concerns of security, product and service quality and data privacy which are fundamental for implementing industry 4.0. With the EU GDPR in force, concerned organizations (European or otherwise) must meet its requirements by reviewing and realigning their business processes. It is necessary for software to be designed accordingly to reduce overheads from organizational measures used in the interim. In this spirit, we propose a new way to check the compliance of current running business processes. DL and LTL are used to describe the constraints related to data. Related algorithms are presented to detect the potential violations, i.e. data access and availability violation, data authentication violation, and data privacy violation. The research of collaborative process model verification covered also control flow and resource constraint verifications. For page limitation, we only present data constraint verification in this paper. Further research related to data constraint verification will carry out the practical implementation and evaluations as the next step.

Acknowledgements: This research is partially funded by the State Key Research and Development Program of China (2017YFE0118700) and it is part of the FIRST project which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 734599.

References

[1] M. Hashmi, G. Governatori, H. P. Lam, and M. T. Wynn, "Are we done with business process compliance: state of the art and challenges ahead," *Knowl. Inf. Syst.*, vol. 57, no. 1, pp. 79–133, 2018.

[2] M. Pesic, *Constraint-based workflow management systems: shifting control to users*. 2008.

[3] M. Pesic, H. Schonenberg, and W. M. P. Van Der Aalst, "DECLARE: Full support for loosely-structured processes," *Proc. - IEEE Int. Enterp. Distrib. Object Comput. Work. EDOC*, pp. 287–298, 2007.

[4] A. Awad, G. Decker, and M. Weske, "Efficient Compliance Checking Using BPMN-Q and Temporal Logic," 2008.

[5] W. M. P. van der Aalst, H. T. de Beer, and B. F. van Dongen, "Process Mining and Verification of Properties: An Approach Based on Temporal Logic," pp. 130–147, 2005.

[6] A. Speck, S. Feja, A. Lotyzc, and C. Kiel, "Framework for Business Process Verification," no. Bis, pp. 50–61, 2011.

[7] M. T. Wynn, H. M. W. Verbeek, W. M. P. van der Aalst, A. H. M. ter Hofstede, and D. Edmond, "Business process verification – finally a reality!," *Bus. Process Manag. J.*, vol. 15, no. 1, pp. 74–92, 2009.

[8] E. R. Taghiabadi, *Understanding Non-compliance*. 2017.

[9] A. Elgammal, O. Turetken, W. J. van den Heuvel, and M. Papazoglou, "Formalizing and appling compliance patterns for business process compliance," *Softw. Syst. Model.*, vol. 15, no. 1, pp. 119–146, 2016.

[10] C. Cabanillas, M. Resinas, A. Del-Río-Ortega, and A. Ruiz-Cortés, "Specification and automated design-time analysis of the business process human resource perspective," *Inf. Syst.*, vol. 52, pp. 55–82, 2015.

[11] A. Del-Río-Ortega, M. Resinas, C. Cabanillas, and A. Ruiz-Cortés, "Defining and analysing resource-aware process performance indicators," *CEUR Workshop Proc.*, vol. 998, pp. 57–64, 2013.

[12] Z. Huang, X. Lu, and H. Duan, "Mining association rules to support resource allocation in business process management," vol. 38, pp. 9483–9490, 2011.

[13] J. Nakatumba, *Resource-aware business process management: analysis and support*. 2013.

[14] D. Knuplesch, L. T. Ly, S. Rinderle-ma, H. Pfeifer, and P. Dadam, "On Enabling Data-Aware Compliance Checking of Business," 2010.

[15] D. Borrego and I. Barba, "Conformance checking and diagnosis for declarative business process models in data-aware scenarios," *Expert Syst. Appl.*, vol. 41, no. 11, pp. 5340–5352, 2014.

[16] M. Salnitri, F. Dalpiaz, and P. Giorgini, "Modeling and verifying security policies in business processes," in *Lecture Notes in Business Information Processing*, 2014, vol. 175 LNBIP, pp. 200–214.

[17] L. Compagna, D. R. dos Santos, S. E. Ponta, and S. Ranise, "Cerberus: Automated synthesis of enforcement mechanisms for security-sensitive business processes," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9636, pp. 567–572.

[18] G. Karjoth, "Aligning Security and Business Objectives for Process-Aware Information Systems," *Proc. 5th ACM Conf. Data Appl. Secur. Priv. - CODASPY '15*, pp. 243–243, 2015.

[19] C. Combi, L. Viganò, and M. Zavatteri, "Security Constraints in Temporal Role-Based," *Codaspy*, pp. 207–218, 2016.

[20] A. Vijay, "Security for workflow systems," *Inf. Secur. Tech. Rep.*, vol. 6, no. 2, pp. 59–68, 2001.

[21] M. C. Mont and R. Thyne, "Privacy policy enforcement in enterprises with identity management solutions.," *J. Comput. Secur.*, vol. 16, no. 2, pp. 133–163, 2008.

[22] M. C. Mont and R. Thyne, "A Systemic Approach to Automate Privacy Policy Enforcement in Enterprises information lifecycle management A Systemic Approach to Automate Privacy Policy Enforcement in Enterprises," *Policy*, pp. 118–134, 2006.

[23] A. R. Khan, "Access control in cloud computing environment," *ARPN J. Eng. Appl. Sci.*, vol. 7, no. 5, pp. 613–615, 2012.

[24] A. Alshehri and R. Sandhu, "Access Control Models for Virtual Object Communication in Cloud-Enabled IoT," in *2017 IEEE International Conference on Information Reuse and Integration (IRI)*, 2017.

[25] J. Warner and V. Atluri, "Inter-instance authorization constraints for secure workflow management," p. 190, 2006.

[26] D. Basin and E. T. H. Zurich, "Optimal Workflow-Aware Authorizations," *Proc. 17th ACM Symp. Access Control Model. Technol. ACM.*, pp. 93–102, 2012.

[27] K. Tan, J. Crampton, and C. A. Gunter, "The Consistency of Task-Based Authorization Constraints in Workflow Systems," *Proc. 17th IEEE Comput. Secur. Found. Work.*, pp. 155–169, 2004.

[28] A. Tealeb, A. Awad, and G. Galal-Edeen, "Context-based variant generation of business process models," in *Lecture Notes in Business Information Processing*, 2014, vol. 175 LNBIP.

[29] R. Lu, S. Sadiq, and G. Governatori, "On Managing Business Processes Variants ⋆," no. February 2009.

[30] H. Groefsema, *Business Process Variability A:A Study into Process Management and Verification*. 2016.

[31] H. Groefsema and D. Bucur, "A Survey of Formal Business Process Verification: From Soundness to Variability," *Proc. Third Int. Symp. Bus. Model. Softw. Des.*, pp. 198–203, 2013.

[32] J. P. Kasse, L. Xu, P. T. de Vrieze, and B. Yuwei, "Process Driven Access Control and Authorisation Approach," 2019.

[33] J. P. Kasse, L. Xu, P. T. de Vrieze, and B. Yuwei, "The Need for Compliance Verification in Collaborative Business Processes," *Work. Conf. Virtual Enterp. Springer, Cham.*, pp. 217–229, 2018.

[34] J. P. Kasse, L. Xu, and P. de Vrieze, "A comparative survey of Business Process Verification Methods and Tools," *Work. Conf. Virtual Enterp.*, pp. 355--367, 2017.

[35] F. Baader, "Basic Description Logics," *Theory Implementations Appl. Cambridge*, 2003.

[36] C. Cabanillas, M. Resinas, A. Del-Río-Ortega, and A. Ruiz-Cortés, "Specification and automated design-time analysis of the business process human resource perspective," *Inf. Syst.*, vol. 52, pp. 55–82, 2015.

[37] G. Lowe, "Specification of communicating processes: Temporal logic versus refusals-based refinement," *Form. Asp. Comput.*, vol. 20, no. 3, pp. 277–294, 2008.

[38] N. Russell, A. H. M. Hofstede, D. Edmond, and W. M. P. Van Der Aalst, "Workflow data patterns," *Business*, vol. 66, no. FIT--TR--2004--01, p. 2004-01, 2004.