www.h2020first.eu

FIRST – Project Number: 734599

H2020-MSC-RISE-2016 Ref. 6742023

Overview of Service-oriented Business Process Verification

Project Coordinator: Lai Xu, Bournemouth University, UK

With contributions from:
University of Bournemouth (BU), UK
Shanghai Polytechnic University (SSPU), China
KM Software, China

Revision History	10/08/2018 Layout BU 11/08/2018 First Draft of D1.2 by BU 14/08/2018 update of D1.2 by SSPU

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 734599
H2020-MSC-RISE-2016 Ref. 6742023

Table of Contents

3.Introduction

Virtual factories are created when distributed manufacturing, virtual enterprises, and business management amalgamate. From a technical perspective (Jain, Sanjay, et al. 2001) (Zhai, Wenbin, et al., 2002) a vF refers distributed, integrated, computer-based composite model of a total manufacturing environment, incorporating all the tasks and resources necessary to accomplish the operation of designing, producing, and delivering a product (Jain, Sanjay, et al., 2002). From the manufacturing practice, the machines, processes, related products and services are directly made compatible to support automated design and verification of collaborative business processes. All manufacturing processes from individual enterprises integrate into a joint collaborative business process (cBP) that is specified, designed and verified to take advantage of the pool of skills, resources and technology. In that way, support for analysis of different design alternatives, performance evaluation and reduced time-to-production is achieved through a consolidated effort.

By nature cBPs are complex, dynamic, and cross organizational borders to involve different partners. They heavily rely on data emanating from partners for their design and execution. Therefore a need for verification of cBPs remains resonant and we posit that it should be supported with canonical methods or tools to avoid errors at execution. Literature remains scanty concerning methods, techniques and tools applicable to verify cBPs especially in a vF environment. Verification of single organization business processes has been well addressed with various approaches (Larsen et al., 1997), (van der Aalst, W., 1997), (Alur, 1996) (van der Aalst, W., 2000), (van der Aalst W. and Ter Hofstede, A. 2000), (van der Aalst, W. and van Hee, K., 2004), (Anderson, B. B, et al., 2005) (Pesic and Van der Aalst, 2006). cBPs differ from single organization business processes in nature and structural design (van der Aalst, W. and van Hee, K., 2004) (van der Aalst, W., 2000), thus their design, verification and execution differs more so in virtual environments where execution is automated.

Existing verification approaches present realizable knowledge gaps; they concentrate on verifying control flow aspects (van der Aalst, W. and van Hee, K., 2004) (Anderson, B. B, et al., 2005) (Pesic and Van der Aalst, 2006), (van der Aalst W. and Ter Hofstede, A. 2000), (Aalst, Wil M. P. van der. 2000), (Varea, M. 2002), (Adamides, E. D., & Karacapilidis, N. 2006), (Aalst, van der W. W., & Pesic, M. M., 2006), (Norta, A. A., Grefen, P. P., & Narendra, N., 2014) while abstracting from other perspectives like data. Data is a major input for smart devices and machines in a vF supporting automated execution of processes. Besides, best practice linking verification approaches to vF cBPs is missing. The EU H2020 FIRST (vF Interoperation suppoRting buSiness innovaTion) project aims to develop a method to support non expert end users to model and verify vF cBPs.

This report presents the state of art in business process verification approaches and makes a comparative assessment of their fitness to verify vF cBPs. The vF being a highly data intensive environment, we describe the data requirements of cBPs and their verification for error free execution of cBP.

Compliance constrains business processes to adhere to certain guidelines, standards, laws and regulation. Non-compliance subjects enterprises to litigation and financial penalties like fines or law suits. Collaborative business processes cross organizational boundaries and regional borders implying that internal and cross regional laws and regulations must be complied with or satisfied. To protect customs' data, European enterprises need to comply with the data privacy regulation from EU (general data protection regulation - GDPR) and each member state's data protection regulations or laws. One example of non-compliance with GDPR is Facebook, it is accused for breaching its subscribers' trust. Collaborative business processes often cross different organizations from different countries. Compliance verification is thus essential for deploying and implementing collaborative business process systems. Compliance verification involves ensuring that collaborative business processes are checked for conformance to compliance requirements

throughout their life cycle. In this paper we take a proactive approach aiming to discuss the need for design time preventative compliance verification as opposed to after effect runtime detective approach. We use a real-world case to show how compliance needs to be analyzed and show the benefits of applying compliance check at the process design stage.

Compliance is about adherence to regulations, guidelines or predefined legal requirements like norms, laws and standards. In terms of business processes, compliance relates to conformance to different process perspectives (Borrego and Barba 2014, Ramezani et al. 2014), namely control flow, resources, data, and time, where control flow strict adherence to the sequential of flow of activities and their relationships; resources adherence to policies constraining allocation and assignment of resources to perform tasks; data adherence to data access control; and time relates to temporal aspects like delays and time lags. These perspectives constrain the business process according to the internal organizational policies. Besides, external policies and regulations present compliance demands that must be satisfied especially for by business processes that cross organizational and regional borders. Such are known as collaborative business processes (Schulz and Oklowska 2004, Ziemann and Matheis 2007, Telang and Singh 2012), a new trend of borderless business processes subject to international regulations. Moreover, partner organizations vary the core process to suit specific needs of their market or business environment resulting into process variants. Notably, the variants must stay in compliance with the core business process. Such scenarios signify compliance as a big and relevant topic with numerous forms of application.

The rising challenges and dynamics surrounding compliance today have compelled new laws and regulations to come into existence or revision of existing ones by the regulatory agencies, e.g. the GDPR, Sabanese-Oxley Act 2002 (SOX), Base III, ITIL, ISO 2700, and Consumer Protection Act 2015 (CPA) among others. The non-regulatory organizations are on the other hand required to exhibit compliance to the regulations and laws through their business processes. Non-compliance results into fines, litigations or loss of business and costs the image of the organization. Facebook is currently striving to rebuild public trust due breach of subscriber trust and non-compliance to data privacy (Guynn 2018).

Compliance provides means to monitor adherence to quality standards for products and services, consumer protection and operational transparence. Also, strict adherence to financial and accounting standards enables firms to keep in operation without which they plunder into oblivion or bankruptcy as was the case of Tyco, Global Crossing and Adelphia, Enron, HIH, Société Générale and Worldcom corporate scandals. Furthermore, where process variants exist and entry into a new market is required, compliant variants can easily be selected for similar environments. For example, a collaborative business process is varied to suit laws and regulations of different countries, the most similar or closely compliant process variant is chosen thereby saving time.

Compliance in business process management is complex and not an automatic task to achieve especially where end users are non-experts in modeling. As observed (section 2), support for compliance has been built upon well-structured but non-collaborative business processes whose interaction is limited to single organizations, based on control flow and resource perspectives. Employed techniques like process mining are curved upon the detective after-the-effect approach seeking to monitor conformance of observed behavior with modeled behavior. This mainly addresses control flow conformance internal policies. A knowledge gap exists to support compliance of collaborative business processes with policies beyond control flow to external regulations, laws and standards. With the expanded scope of constraints, it is also necessary at design time to verify between internal and external regulations to ensure that they map and synchronize to avoid any unpredictable conflicts that can cause deadlocks in the process.

To that effect, we adopt concept of compliance-by-design (Sadiq and Governatori 2010) as a paradigm to achieve design time preventive compliance of the business process models with regulatory requirements. Compliance-by-design is a process of developing a software system that implements a business process in such a way that its ability to meet specific compliance requirements is ascertained (Kochanowski et al. 2014). To achieve compliant business processes at

runtime, compliance strategies must be built and checked at design time. In this paper we emphasize the need for design time compliance checking. This is tenable through application of formal methods to reason about business processes as system models and compliance requirements as properties to automate compliance rule verification.

This report is based on two published papers (Kasse JP et al., 2018) and (Kasse JP, Xu L, and de Vrieze P, 2017). The report is mainly contributed by John Kasse, Dr. Lai Xu, Dr. de Vrieze Paul from Bournemouth University, Prof. Yuewei Bai from Shanghai Polytechnic University, and Mu Hua from KMSoftware.

4. State of the art in Business Process Verification

2.1 Business Process Verification Approaches

Business process verification also known as model checking (MC) is an area of different application expressed in terms of variability, compliance, compatibility and verification; *Variability* involves checking to ascertain how business processes vary in behavior given a set of conditions at design time or run time (Varea, M. 2002), (Aiello, M., Bulanov, P., & Groefsema, H., 2010). *Compliance* checks model conformance to a set of specifications like business requirements or laws/ standards (Adamides, E. D., & Karacapilidis, N. 2006), (Aalst, van der W. W., & Pesic, M. M., 2006), (Norta, A. A., Grefen, P. P., & Narendra, N., 2014), (Knuplesch, D. et al., 2013), (Kochanowski, M, et al.2014), (Fdhila, Walid, et al., 2015). *Compatibility* involves making sure that partner business processes are aligned to fit the interaction model. The interaction model represents the interaction architecture through which the cBP is executed (Aiello, M., Bulanov, P., & Groefsema, H., 2010), (Backer, Manu De, et al., 2009). *Verification* aims at checking and correcting errors in process models. The above represents the state of art in business process verification. Traditionally modelling languages only supported simulation as a way to validate designed models. Simulation is however limited since it is based on partial data and assumptions that may not be a true representation of the actual business requirements.

During business process design, more time is spent on verification than actual design. Formal verification leads to seminal advantages like connectivity to achieve timeliness, security and dependability, model correctness and, model scalability in terms of number of services that can be supported (Knuplesch, D. et al., 2013), (Morimoto, S., 2008). There are many verification approaches, presented herein is a description of selected approaches based on their application and popularity in the cited literature. The discussion is biased towards how each approach aligns itself towards supporting cBP model verification.

E-C-A Based Business Rules: A declarative approach that supports validation and simulation of process models to detect design errors. Events (triggered by human, machine or application) are checked against a Condition and Action is executed if the condition is satisfied. The approach integrates with CPN tools to convert specified models into a set of E-C-A rules which are automatically checked for compliance and semantic correctness at run time (Pham and Thanh 2015) as well as termination and confluence (Jin et al. 2013).

Temporal Logic: Supports ways to specify systems and check models for correctness against a set of properties expressed in form of event orderings in time (Kochanowski, M. et al., 2014), (Morimoto, S. (2008). (Lowe, G., 2008). The specifications are expressed in Linear Time (LTL) or Branching Time logic. Temporal Logic employs a set of temporal operators like Eventually, Next-time, Always, and Until to specify model behavior and, express constraints and rules that the models must conform to through model checking. It is widely applied to verify concurrent systems, distributed systems, context aware and collaborative systems due to its richness and mathematical foundation supporting theorem proof. Thus the extensions into Computation Tree Logic (CTL), Proposition Tree Logic and Timed LTL (Giannakopoulou, D., & Havelund, K., 2001), (Havelund, K., & Rosu, G., 2001), (Roşu, G., Chen, F., & Ball, T. 2008) as summarized in table 1 summarizes in relation to their application (Baier and Katoen 2008).

Logic	**Linear time (path-based)**	**Branching time (state-based)**	**Real time Requirements (continuous-time domain)**
LTL	✓		
CTL		✓	
Timed LTL		✓	✓

The limitation of temporal logic is the lack for a graphical interface restricting its application to expert users. However tools with graphical user interface have been developed based on LTL like DecSerFlow (Pesic, M., & Aalst, van der W., 2006), (Aalst, van der W., & Pesic, M., 2006).

Declarative Service Flow Language (DecSerFlow): DecSerFlow supports specification, enactment, and monitoring of service flows in a declarative nature. It supports verification of service workflow conformance (Aalst, van der W., & Pesic, M., 2006) by subjecting developed graphical models too hard or soft temporal constraints which are enforced by the engine. Constraints are used to guard against violations and monitor observed violations.

Petri nets (Petri, C. A., 1977)*:* as a modelling and verification technique Petri nets are simple, easy to access and well-grounded mathematically (van der Aalst, W. M. P., 2004). When are applied in workflow technology Workflow Nets are created. A workflow net must meet a syntactical requirement of having each place or transition on a direct path from start to end. Such requirement satisfies the workflow property of soundness (van der Aalst and van Hee 2004) which is the major property verified in workflow nets. A workflow net is sound when; it is live, there are no deadlocks and all states are reachable. A live workflow net implies that tokens cannot be held in endless loops (live-locks), lack of deadlocks implies nonexistence of dead transitions or cases where the transition of a token cannot reach a final state, and Reachable refers to having all states in a net reachable from the initial state to the final state.

Application of classical petri nets in large cases exposes their limitations i.e. they become so large, inaccessible and difficult to express or interpret (van der Aalst and van Hee 2004). To overcome the limitations to high level petri nets were proposed.

High Level Petri nets/ Colored Petri Nets (CPN): CPN tools enable modelling of data, objects and structures using color (Fahland, D., et al., 2009) and support verification (Jensen, K., Kristensen, L. M., & Wells, L., 2007), (Gottschalk, F., et al., 2008). *Color* ex*tension* expresses each instance as unique specifying its characteristics in a case, *time extension* captures time related information to track expected completion time or expected capacity of a process, and *hierarchy extension* supports hierarchical design of process models and sub process models that are designed with subnetworks that comprise places, transitions and arcs hierarchically linking to or from the main process model. CPN tools integrate with other tools to support verification of models, for instance Protos and E-C-A.

Application Development based on Encapsulated pre-modelled Process Templates (ADEPT)/AristaFlow: ADEPT/AristaFlow is a family of tools used to support modelling and verification of flexible and dynamic business processes (Reichert, M. & Dadam, P., 1998) (Weber, B., Reichert, M., & Rinderle-Ma, S., 2008) (Weber, B., et al, 2008), (Dadam, P. & Reichert, M., 2009). Based on clinical business scenarios, ADEPT enables process implementers, application developers and end users to model and verify models through its feature like; extended graphical interfaces, plug and play style supporting the on-the-fly correctness checks (Dadam, P. & Reichert, M., 2009), process templates and structural transformation of processes, support for ad-hoc changes and their propagation.

Yet Another Workflow Language (YAWL): YAWL supports both a modelling and verification based on the Petri nets technology (Petri, C.A., 1997) and workflow patterns (van der Aalst, W. M. P., Kiepuszewski, B., & Hofstede, A. ., 2003). It caters for early time detection of model errors. The WofYAWL editor plug-in functionality provides support to verify models for soundness and proper termination properties (YAWL Foundation, 2016).

*Protos***:** Protos supports process model definition and analysis based on different perspectives of data, user, or control flow. It supports simulation for quantitative analysis of models before their enactment and execution. Protos2CPN tool is an integration of Protos with CPN tools to support process model verification (Gottschalk, F. et al., 2008).

FlowMake: The tool supports design time identification of errors in business process models before implementation (Wasim, S. & Maria, O., 2000). Graph reduction algorithm (Lu, R. & Sadiq, S., 2007) is employed to verify workflows for syntactic correctness based on a set of constraints.

The algorithm specifies rules to reduce the WF graph by identifying and eliminating structural conflicts like deadlocks and lack of synchronization. Correct structures are removed until the WF graph remains empty through a conflict reserving reduction process. A WF graph with structural conflicts is not completely reduced.

HYbrid TECHnology (HyTech): HyTECH supports automatic verification of embedded systems with specification of properties expressed in real time temporal logic and verified through symbolic computation (Henzinger, T. A. & Wong-toi, H., 1997). Systems are modelled as hybrid automata i.e. a finite state machine with both discrete and continuous variables. Models are verified for reachability, liveness, time boundedness and duration properties (Henzinger, T. A. &Wong-toi, H., 1997). HyTECH is recommended verification of mission critical systems that require no margin of error. However, the tool lacks support for simulation, limited to verification of small systems (Bérard, B., et al, 2001) and linear hybrid systems (Henzinger, T. A., Horowitz, B., & Majumdar, R., 1999). Some of the limitations have been overcome by HyTECH+ tool (Bérard, B. et al, 2001) which is an extension to the classical HyTECH.

Symbolic Model Verifier (SMV): SMV is a model checker based on binary decision diagrams where a set of states and transitions are considered in a single block than a single state at a time (Clarke, E., Emerson, E., & Sistla, P., 1986). Due to state explosion issues, NuSMV is a modified version based on LTL and CTL to verify synchronous finite-state and infinite-state systems. Results must satisfy a set of temporal specifications and if not, a counter example yields (Cimatti, A., Clarke, E., & Giunchiglia, E., 2002), (Kadono, M., Tsuchiya, T., & Kikuno, T., 2009).

SPIN: It supports verification of asynchronous systems by verifying correctness properties expressed as standard LTL against model specifications expressed as a Buchi automaton. The Buchi automaton is a product from computation of the claims and the automaton representing the global state space. The product is then checked, if empty then the claims are not satisfied for a given system, otherwise it contains the behavior that satisfies the original temporal formula. To limit state explosion during verification, partial order reduction method is employed (Petri, C.A., 1977) , (Holzmann, G. J., 1997), (Holzmann, G. J., 2017), (Holzmann, G. J., Godefroid, P., & Pirottin, D., 2013). The tool still faces the state explosion issues limiting its applicability to cBP model verification that come with voluminous states.

KRONOS: KRONOS is based on timed automata and timed temporal logic supported with an engine which integrates with other design environments. Models are verified for reachability properties (Holzmann, G. J., Godefroid, P., & Pirottin, D., 2013), (Yovine, S., 1997) like ; safety (system never enter unsafe states), non zenoness (the state of the system does not prevent time to diverge) and bounded response (ability to respond to requests issued in specified time).

UPPAAL: UPPAAL supports on-the-fly verification of real time systems modelled as timed automata with extended data. It checks models for reachability and invariability properties with support for diagnostic trace showing why a particular property or is not satisfied (Larsen, K. G., Pettersson, P., & Yi, W., 1997), (Larsen, K. G., Pettersson, P., and Yi, W., 1995). State explosion remains a challenge limiting its application to cBP model verification.

Table 2 Summary of Tools and Properties

Language	Tool	Properties	Environment
Petri nets	YAWL	Soundness and Liveness	Non collaborative
	CPN Tools	Coverability & occurrence	Non collaborative
	Woflan	Soundness, Reachability and Liveness	Non collaborative
	XRL\Woflan	Soundness, Reachability and Liveness	Collaborative
	Protos2CPN	Soundness, Reachability and Liveness	Non collaborative
BPMN	BPMN – Q	Boundedness and Reachability	Non collaborative
	FlowMake	Consistency, deadlocks, synchronization	Non collaborative
Temporal logic	SPIN	Correctness and logical consistency	Non-collaborative
	UPPAAL	Bounded Liveness, deadlock freeness and ability to meet deadlines	Non-collaborative
	KRONOS	Reachability (Safety, Non zenoness, Bounded response)	Non-collaborative
	SMV/ NuSMV	Correctness, safety, and liveliness	Non-collaborative
	HyTECH	Reachability, Safety, Liveness, time-bounded, and duration	Non-collaborative

2.2 Comparison Framework for Business Process Verification Approaches

Language comparisons are based on different factors that may be objective or subjective (Falkenberg, E., Hesse, W., & Lindgreen, P., 1998). We choose a set of parameters to compose our criteria to assess the inherent traction and precision of the verification approaches and their appropriateness to verify vF cBP models. The following section briefly describes the parameters that compose the assessment criteria;

Expressibility describes the degree to which an approach can represent any number of models in different application domains (Falkenberg, E., Hesse, W., & Lindgreen, P., 1998), (Hommes, B. J., 2004). In (Lu, R. & Sadiq, S., 2007), the expressive power of a modelling technique was gauged in terms of its capability to represent specific process requirements. In our case, we consider expressiveness of a model verification approach in terms the degree to which it enables one to verify different properties of cBP models given their specifications.

Flexibility describes the ability to support exception handling, possibility to make changes at design time verification or runtime, and support for scalability especially as the cBPs evolve and grow.

Suitability describes the appropriateness of an approach to a particular application domain (Falkenberg, E., Hesse, W., & Lindgreen, P., 1998), (Hommes, B. J., 2004). In our case we assess suitability in terms of the degree to which an approach is applicable to verify vF cBP models given their structure and architecture for instance; verify semantical correctness of main models and sub models simultaneously.

Complexity assesses the level of difficulty an approach presents to work with while being used to verify a process model (Lu, R. & Sadiq, S., 2007).

Limitations are the different forms of inadequacies of an approach that render it inappropriate and inapplicable to verify vF collaborative business process models.

2.3 Limitations of the Verification Approaches to Verify cBP Models

Based on the assessment in table 1, we find verification approaches lacking in terms of support to verify cBPs. We expound on these limitations;

Not built for verification purposes: existing approaches were developed to support modelling and simulation of single organization business processes, not cBPs. Models would be analyzed through simulation but it remains limited as noted in section 2.1. Upon verification, some techniques were modified or integrated with other tools to support verification (e.g Protos and E-C-A integrate with CPN tools) (Gottschalk, F., 2008). More so, some approaches like YAWL can only verify models designed in the same language. For Woflan which was created as an independent verification approach, it can only support a few models developed in Staffware, COSA and MQ (Verbeek, H. M. W., Basten, T., & Van Der Aalst, W. M. P., 2001). Therefore the existing approaches were not built for cBP verification.

The semantical and architectural structure: The approaches do not support the semantical structure and architecture required in the cBP verification. For instance the lack of interfaces or open structures to permit integration with other collaborating systems. YAWL avails web based plugins for integration to other system but the limitation of inability to simultaneously verify models and sub models remains a challenge. Additionally the semantical structure of some of the tools is ambiguous and a source of semantical errors and conflicts during the merging of models for verification (Koliadis, G., 2007).

Lack of consideration for data and data analytics: Most approaches target verification based on control flow perspective while abstracting from other perspectives like data, resources, tasks and applications (Aalst, W. M. P. Van Der, 1997), (W.M.P. van der Aalst, 2000), (Verbeek, H. M. W., Basten, T., & Van Der Aalst, W. M. P., 2001), (Roa, J., Villarreal, P., & Chiotti, O., 2011). The justification advanced for abstraction never anticipated future data requirements that vF processes present now. vF heavily relies on data routed among interconnected smart devices to drive the automated machines at the factory floor. Moreover, analyzing existing data will useful for analytics to support process verification, decision making, projections and future planning. Therefore during verification data and data analytics should be supported at both design time and runtime.

Table 1 Summary of the Assessment of the Approaches

Approach	Properties	Flexibility	Suitability	Complexity	Limitations
Woflan	Soundness and Liveness	Verifies complete models	Verifies models from other languages.	Ease of use with user interface. Hard to trace errors or understand outcome.	Non-collaborative. Single model verified at a time.
YAWL	Soundness and Liveness	Design time exception handling model verification	Control flow specific Main model & sub model verified independently	Supports extension through plugins. Graphical interface	Non-collaborative
FlowMake	Synchronisation, Deadlocks, consistency, Boundedness, Liveness	Design time exception handling. Non scalable as models grow	Supports data perspective. Non domain specific. Models and sub models verified independently.	Graphical interface makes it usable for non-expert users	Non-collaborative Control flow based. It is difficult to trace errors
Colored Petri Nets	Performance analysis Coverability and occurrence	Supports exception handling on time outs	Verifies concurrent systems Not domain specific Models and sub models verified independently	Graphical tool with less complexity	Non-collaborative support
SPIN	Correctness and logical consistency	Support for exception handling	Based on temporal logic viable for vF cBP Wide application Not domain specific	Complex syntactical structure and semantics. XSPIN provides a graphical interface.	Non-collaborative. State explosion. Restricted to smaller systems
UPPAAL	Bounded Liveness, deadlocks & meet deadlines	Supports on-the-fly verification.	No support for data analytics.	Supports diagnostic trace to source of errors.	Non-collaborative support. Non scalable
KRONOS	Reachability - Safety, Bounded response	Design time verification. Support for exception handling	No known application to vF domain Models and sub models verified independently	Graphical interface eases use Counter examples to aid verification	Non-collaborative Limited to smaller models No support for data
SMV/ NuSMV	Correctness, safety, and liveliness	Support for exception handling at design time	Non domain specific, Models and sub models verified independently	Graphical interface eases usability Counter examples to aid verification	Non-collaborative State explosion
HyTECH	Reachability, Safety, Liveness, time-bounded, duration	Less regard to exception handling. Non scalable	Lacks elements like data which a key to vF cBP	Complex tool due to syntactical and semantic requirements	Non-collaborative State explosion Restricted to smaller systems
Woflan	Soundness, Liveness and Reachability	Verifies complete models, Non flexible.	Verifies models from other languages. Single model verified at a time	Graphical interface for usability	Non collaborative models. Output not easy to understand
ADEPT	Semantic correctness, deadlock and Safety	Supports for exception handling	Applicable to other domains besides Clinical.	Use of process templates to easily create processes.	No proven application. Models and sub models verified independently

5. State of the Art in Compliance

Compliance, its checking and verification in business process management and workflow management has been widely addressed from different angles; compliancy to control flow aspects of the business process i.e. checking whether observed behavior in execution logs matches the modeled behavior (Goedertier and Vanthienen 2006, Borrego and Barba 2014, Ramezani et al. 2014), resource allocation i.e. role, task and attribute based approaches (Thomas and Sandhu 1997, Sandhu 2003, Yuan and Tong 2005, Gautam 2017), as a security mechanism for workflow systems (Salnitri et al. 2014, Müller 2015, Combi et al. 2016, Robol et al. 2017) and compliance verification approaches (Elgammal et al. 2016, 2016). Similarly, compliance is addressed from 2 fronts i.e. at design time or runtime. Some approaches however target both design time and runtime compliance.

Design time compliance checking is a preventative approach that addresses compliance of business process models to constraints before execution i.e. compliance constraints are enforced on models and checked before execution. On contrary, *runtime compliance* checking is a detective after-the-effect approach for monitoring compliance of business processes while they are in execution (Sadiq et al. 2007, Sadiq and Governatori 2010). Each approach presents pros and cons, while the runtime approach is considered flexible and declarative being able to capture compliance issues beyond design; the design approach is preferred for being proactive to deal with compliance violations before they arise and permitting early time correction during process design. Following is a discussion of some relevant related work.

The PENELOPE tool is based on deontic temporal logic to support declarative modeling and expression of control flow constraints of process events. Compliance to constraints in form of permissions and obligations to perform events are explicitly expressed as temporal deontic assignments enforced on business processes at design time. A compliant control flow non-executable business process model is generated to support process designers to verify and validate other models by showing decision points and possible violations (Goedertier and Vanthienen 2006, 2007, Goedertier 2008). The approach's application is limited to control flow and resource related compliance checking.

Relatedly, a process fragment lifecycle technique is proposed to support consistent specification, integration and monitoring of compliance controls in business processes. A process fragment is a connected graph representing part of a business process modified to incorporate compliance requirements, which are later integrated into the original business process by means of the so called process 'gluing' and 'weaving' methods to create a compliant business process (Schumm et al. 2010). In this approach, compliance related to control flow and data perspectives is supported. Even then, there is no way to prove lack of deadlocks or livelocks in a compliancy constrained process model i.e. no verification is supported which renders it difficult to determine correctness of integrated compliance changes.

In the paper (Sadiq et al. 2007) the concept of compliance-by-design is coined to overcome limitations of the after-the-effect approaches like process mining. It provides means to reason about compliance rules by modeling control objectives and applying formal methods to enrich business process models with annotations and visualizations (Sadiq and Governatori 2010). The concept is supported by with a formalism for expressive modeling of compliance specifications i.e. the Formal Contract Language (FCL). FCL is a deontic logic and non-monotonic based language supporting design time compliance constraints specification and enforcement on BPMN business process models.

A Contract Language (CL) based on deontic logic is proposed as an approach targeting specification compliance requirements sourced from business contracts written in natural language. Compliance between contract language rules and models is checked via an evaluation algorithm (Prisacariu, Fenech, 2009).

A compliance request language (CRL) is proposed through a compliance management framework as a design time approach to support automated application and checking for

compliance of business process models. CRL is based on temporal logic utilizing formal reasoning over formalized compliance patterns to support compliance constraints enforcement and checking (Elgammal et al. 2016).

Compliancy has as well been addressed from a privacy and security perspective. Policies are specified and enforced on process models to comply to security and privacy requirements. Role based models are proposed in (Sandhu 1996, 2003, Khan 2012, Ertugrul and Demirors 2015, Combi et al. 2016, Alshehri and Sandhu 2017) to support allocation and access to tasks and resources based on roles. Users are grouped into roles and permissions are assigned to groups e.g. Auditors assigned access to some resources in the process. Task based models as proposed in (Thomas and Sandhu 1997, Tan et al. 2004, Wu 2007) provide a dynamic approach to compliance of business process models to access and authorization policies based on the tasks executed in the process. Compared to RBAC (Role-based Access Control ???), TBAC offers simplified, automated and self-admissible models where access to tasks is authorized following the context and progress of the process. On another hand, Attribute based models regulate access and authorization through a combination of attributes of both the subject (requester) and the object (e.g. file), and the environment (Yuan and Tong 2005, Khan 2012, Gautam 2017, Axiomatics 2018). The proposed models in this case guide the specification, enforcement and monitoring of to ensure compliance to policies related to resource allocation, authorization and access control to tasks, resources and data in workflow systems. Such policies target constraining business processes and the user to comply to requirements like segregation of duty, binding of duty, need to know among others which prevent or detect fraud, errors of commission or omission. However, these proposals do not provide mechanisms for design time verification. Besides, there is no application to collaborative environments can be noticed so far.

Moreover, in (Salnitri et al. 2014) a framework for supporting compliance to security policies in large autonomous information systems is proposed and implemented. SecBPMN is used to design process models while security policies are expressed using SecBPMN-Q after which the SecBPMN-Q are verified against SecBPMN specifications via an implemented query engine. The approach remains limited to security policies disregarding other relevant policies.

A socio-technical security modeling language (STS-ml) is extended to support privacy by design i.e. to model privacy as a requirement and support verification of privacy properties of models through formal reasoning (Robol et al. 2017). The approach is bound to privacy policy compliancy and no attention is paid to other compliancy requirements. Moreover, little support is provided to address verification among the compliancy constraints.

A compliance approach based on Petri-net semantics and syntax is proposed to check compliance on two fronts, i.e. checking rules restricting data attributes and rules restricting activities when a certain data condition holds. Process mining technique is employed to extract logs from the process execution and observe behavior. The approach is an after-the-effect theory tracing already executed processes, this way it differs from our proactive compliance approach.

Lastly, a conformance approach for checking compliance of declarative business process models is proposed. It emphasizes inclusion of business data rules on top of control flow rules in the conformance checks and providing related diagnostic information to increase the effectiveness of outcomes. The approach may seemingly be like what we propose, however, the difference lies in our consideration of cross organization processes and cross border regulations. Further still, we also suggest checking for consistence and lack of ambiguity between internal and external regulations.

Table 1 summarizes above mentioned compliance methods. For each compliance method, we look at is the approach related to run time or design time, which formal method is used, and which process aspects of compliances are considered.

Table 1: Summary of Compliance Methods

	Formalism	Application	Methods	Control flow	Resource	Data	Time
Process Mining		Run time	Log data	√	√		
PENELOPE	Deontic logic	Design time	Declarative	√			
Security	-					√	
Process fragment lifecycle	Non	Run time	Imperative	√		√	
Formal Contract Language	Deontic logic	Design time	Imperative	√	√	√	√
Contract Language	Deontic logic, temporal logic	Design time	Imperative	√		√	
Compliance Request language	Temporal logic	Design time	Imperative	√	√		√
AC agent enforcement architecture	-	Design time Runtime	Imperative		√	√	
Formal constrained workflow	Temporal logic	Design time	Imperative		√	√	
PrVBPMN		Design time	Imperative		√	√	
RBAC	Temporal logic	Design time					
TBAC	Temporal logic	Design time			√	√	√
ABAC	Temporal logic	Design time			√	√	√
SecBPMN	Temporal logic	Design time Runtime	Imperative	√	√		
STS-ml	-	Design time Runtime	Imperative	√	√		

6. Requirements for cBP Verification

For a process to be regarded as a cBP, it must conform to a set of requirements that characterise their nature as described;

Span different organisation: Collaboration involves different partners working together and to achieve a common business goal. In terms of business process management the different partners converge to jointly define business and technical solutions. The business solution describes partner behavior in the cBP while the technical solution defines the specifications and implementation of the supporting system (Roa, J., Villarreal, P., & Chiotti, O., 2011). The approach to verify such processes should take into consideration diversity of users will work together at different time and location.

Communication/ Interaction Protocol: Typical of the cBP are the forms of communications and interactions in form of message exchanges among partners who engage in discussions and iterations before reaching a decision. cBPs require dedicated interaction protocols through which partners can communicate as they model and execute processes. Many researches propose interaction protocols (Aalst, W. van der, 2000), (Chiotti, P. D. V. L. R., 2010) but they do not pass the criteria to support cBP verification.

Dynamism, Flexibility and Complexity: Several activities compose a cBP and continuous changes keep coming in timelessly which affects process outcome. The volatility of such processes should be verified and the approach should support integration, propagation and continuous verification.

Data requirements relate to several issues: The operations and decision making in cBP largely rely on the data that surrounds the business processes and the operations. Ordinary workflow systems embraced 2 kinds of data i.e. control data as well as production data. Control data refers variables used for routing purposes while production data pertains to the information objects (e.g. documents, forms, and tables) and its existence does not affect the operations of workflow system (W.M.P. van der Aalst, 2000). Our concern is on production data whose gathering and management is crucial to support data analytics for decision making. The appropriate technique should be able to verify the data patterns to support analytics and decision making. Additionally, big data exists and it is upon us to exploit it for competitive use like faster decision making. Virtual factories work by means of smart devices that drive machines and operations at the factory floor. It involves factory automation relying on intelligent data gathering and data exchange between the cyber physical systems in order to drive operations. Consideration to verify data requirement in cBP is limitation that requires attention. Therefore an approach to verify cBP based on the described requirements is necessary.

Service Oriented Design: cBPs are composed from autonomous business processes and services of the partners forming a choreography implemented across boundaries. Choreography describes interaction between service providers and their users to achieve a particular goal. The functionality of cBPs should be described in such a way that permits flexible integration following a Service oriented design (SOD) approach. SOD supports communication between business process architects enabling the verification of designed cBPs for conformance with requirements. It also facilitates model driven approach to service development and composition (Dijkman, R. & Dumas, M., 2004).

7. Framework for Collaborative Business Process Verification

The assessment based on our criteria revealed various properties being checked. However, these properties were expressed in relation to single organization business processes. The interpretation and connotation of these properties may not the same for inter-organization business processes: for instance having sound models for a single organization process does not guarantee their soundness in a collaborative environment. Furthermore verifying for reachability, safeness, liveness and boundedness in a single organization process is not as complex as verifying the same properties for collaborative business processes. Moreover, there is no silver bullet solution; no single approach verifies all necessary properties for all situations. For example Petri net based approaches and tools like YAWL, Woflan, and CPN are lacking in terms of time based requirements for models. Temporal logic based approaches like SPIN, KRONO and HyTECH suffer from state explosion problem that limits the number and size of models that can be checked. Besides, the counter examples they provide on discovery of errors remain un-understandable to the ordinary users. Above of all, the inability and inconsideration for data perspective leaves them inappropriate to verify collaborative business processes that are highly data intensive. In summary, using the parameters in our criteria we note the following in view of collaborative business processes;

Expressiveness: most approaches are not specific to a particular application domain but incapable of representing as many models for interacting enterprises as may be required. To that effect such approaches would not verify the structure, data and execution requirements of cBP.

Flexibility; besides YAWL, DecSerFlow and AristaFlow tools, other techniques do not show capability for exception handling, support for ad hoc changes and scalability. cBPs are highly variable and dynamic given the diversity of process owners and environment in which they apply. Moreover, the techniques verify completely designed models which renders them rigid and inflexible (Chiotti, P. D. V. L. R., 2010).

Complexity most tools present a graphical user interface making them easy for the non-expert users to use. However, temporal logic expressions are complex for non-expert users from the collaborative environments whose backgrounds vary (Lu, R. & Sadiq, S., 2007).

Suitability and limitation; the techniques are found to inappropriate and not suitable for verification of vF cBP models given the cited limitations in their structural nature and architecture. Lack of standardized semantics introduces semantical errors where models verified are developed from different tools.

8. Motivating Case Study

This section presents a description of an industry collaborative business process that serves as a motivating case study. The case is a 'Pick and Pack' process from a big supermarket with a chain of stores across Europe and some parts of Asia.

To create orders, customers must register on a store's system online. Once a customer order is received, a notification is received at the store while the customer receives a confirmation. Store staffs check order details, pick and pack items. Before packing items are verified by picking staff for conformity with order, and after by handover staff. One or more staff may be assigned to an order depending on its size. For items that may be out of stock, the order is put on suspense for a period until stock is availed or staff is permitted to contact customer to seek opinion either to wait, change or cancel order. Item substitution is permissible, for instance changing fresh a fresh vegetable item to tinned one. A customer can cancel an order delayed beyond a specific time. Ready orders are either picked up by the customers, delivered by store or by a preferred courier. Figure 1 is the model of the pick and pack business process.

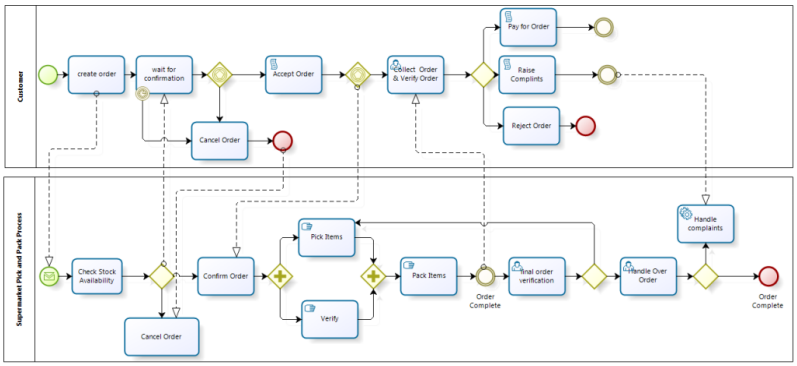

Figure 1: Pick and Pack Business Process

The case study serves as an example of different perspectives and compliance rules specific to a collaborative process e.g. control flow; order confirmation is subject to stock availability, process data; orders can only be delivered if they pass the verification check, process resource; final order verification must be done by different staff, process time; orders can be rejected or cancelled if delay is beyond a specific time. Moreover, there are different stakeholders who present various interests that must be matched and satisfied. Customers buy items and they expect them to be of acceptable quality, non-defective, in right quantities and delivered on time. The store staff and managers work on customer orders; they are expected to meet customer expectations, item availability and timelines. Also, there are different companies in the supply chain like suppliers and couriers. In the background are also shareholders whose aim is profit oriented. They expect financial fluency non-solvency of the company. Unverified compliance issues could lead to potential flaws in the process. For instance; verification concerns like packing unordered items, wrong items quantities, running out of stock, defect items etc. The business process accesses customer data during execution which raises data privacy concerns in terms of legality and legitimization i.e. who has access to data when and for what purpose.

9. Need for Compliance Checking in Collaborative Business Processes

Compliance constrains business processes to adhere to the required behavior as required by the organization, laws or regulations. Therefore, at design time it is cognitively cheaper and useful to check the veracity of the process model to the expected behavior through verification to avoid violations or non-compliance at runtime. Moreover, for collaborative business processes it is important to ensure that all concerns of the stakeholders are under consideration i.e. the customers, employees, partners and regulatory agencies. In this case therefore, verifying for compliance would lead to early time detection a range of errors that would otherwise make the model deviate from expected behavior. Verification should target errors for all perspectives of business process i.e. control flow, resources, applications, data and time. For instance, errors of omission or commissions can lead to omitting a step required for the process to comply to a given law or adding a step that drives the model off compliance track. Logical errors also lead to undesired behavior, data and time related constraints are of major concern in this aspect. Also, ensuring that the process meets compliance to data privacy by verifying for access control and authorization is necessary. The concept of process driven authorization (cf. 1) becomes of relevancy where the need for a resource to access data is derived from its relevance in executing specific task for that specific time. This concept diverges from the traditional access control and authorization concepts discussed in section 2.

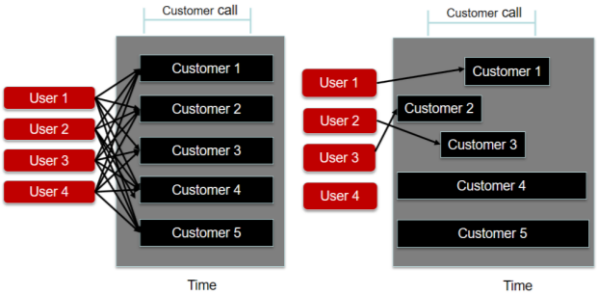

Figure 2: Process driven authorization

In relation to the case in section 3, The aim of analysis is twofold i.e. to improve the business process targeting increase in customer satisfaction and experience, and to ensure compliance of the process to the internal and external regulatory requirements.

9.1 Analysis for Process Improvement

Analysis and application of simulation verification would create room for an improved process or detect design errors that may affect successful execution. Essentially, the output from simulation would create different pointers on how best the process can be improved but also be applicable as input parameters for verification checks. For instance, what if scenarios can be simulated (using historical data) to determine optimum order levels, staff levels and allocation, process cycle times etc. sample scenarios are given in table 1.

Table 2: Examples of Scenarios

	What if scenarios	Impact
1	Dedicated staff work on the pick and pack process	- Specific staffing levels and capacity planning for the process - Specific time required to serve an order - Staff performance indicators
2	Less busy staff are assigned to the process at peak hours	- Identify process peak times and know when to exchange staff - Optimum staffing levels and utilisation or balancing - Work/process optimization - Effect on other departments to which staff belong - Staffing cost indicators - Role conflict indicators
3	Dedicated pick and pack department or store	- Optimum staffing levels and capacity planning for the department - Order management and efficiency - Process cycle times - Optimum staffing levels - Single store location or multi locations - Store navigation layout
4	Staff can cross departments to pick orders for items	- Store navigation layout planning and analysis - Staff competence and performance indicators - Bottleneck or collision analysis
5	Item stock management	- Optimum item stock management (economic order quantity) - Highly/least items on demand

The simulation scenarios would yield different results as indicative impact on the business process from which the best scenario that optimizing case can be selected given dependent factors to inform implementation decisions. Additionally, the outcomes are useful pointers for strategic internal policy formulation as well direction for meeting standardization or regulatory requirements. Lastly, outcomes from the simulation scenarios form input parameters for the intended verification requirements.

9.2 Compliance with Data Privacy

The case shows need to comply with both internal policies and external regulations like the GDPR, SOX and BASE III, national fiscal policy, customer protection rights. Specifically, we show data privacy compliance through access control and authorization as propose process driven authorization. This is due to space limitations and need to emphasize compliance with the revised GDPR before it comes into effect. We however describe other regulations.

The GDPR emphasizes compliance to data privacy in which data controllers are responsible for data protection in the organization. It requires keeping data for individuals private, have their consent to collect and process it, notify them if there is any change, avail it to owners if needed in a required format and seek their consent before it can be transferred to third parties. In the case, the business process runs on customer data which is collected at registration time. Often, orders may be delivered by other delivery companies whereby customer data is passed to a third party. Within

Europe, different countries treat different kinds of customers' data differently. Therefore, there are challenges even for specifying a same business function process in different ways in different countries. Financial reporting requirements are based on international financial and accounting standards like the SOX and Base III. They regulate compliancy to financial standards to protect shareholders and the public from financial manipulations, intentional errors and fraudulence. This improves the accuracy of corporate disclosures. The super market is required to maintain a stable financial position to the satisfaction of shareholders. Fiscal policy is a national border law that differs per region. It demands openness and transparency of business processes to enable tax bodies to assess, track and monitor compliance to regional tax policies to prevent tax fraud.

Against the described regulations and in relation to the business process details (section3), table 2 shows extracted compliance requirements along with the relevant sections of the regulations.

Table 3: Compliance Requirements Generated from the Case

Req	**Use case compliance scenario**	**Compliance requirement**	**Policy level/ Regulation**
Rq.1	Customer registers on system with private data	Inform owners what data is collected, processed and intended use	Data privacy GDPR
Rq.2	Customer submits order(s). The system notifies customer of successful submission immediately	Notify customer of the order details submitted	Internal policy
Rq.3	Notify customer when order(s) will be ready. Orders are ready between 30 and 60 minutes	Notify customer of the waiting times	Internal policy
Rq.4	For delays notify customer. Customer can choose to wait or cancel the order.	Notify customer of any delays Right to terminate the purchase and get a full refund If the delivery isn't time-essential but another reasonable delivery time can't be agreed, your right to cancel the order for a full refund	Internal policy, Consumer Rights Act 2015
Rq.5	When item is not in stock the customer must be informed. Customer can cancel item, substitute it or cancel the order.	Customer priority	Internal policy Consumer Rights Act 2015.
Rq.6	Initial order verification is done by same staff who picked the items before packing	Binding of duty	Internal policy
Rq.7	Different staff verify order details before delivery	Critical duties are segregated (duty separation)	Internal policy ISO IEC 27002 6.1.2
Rq.8	Orders or items can be rejected if they don't meet expectation e.g. defect items	Goods should be as described, of satisfactory quality, fit for purpose	Consumer Rights Act2015. Goods return policy
Rq.9	Customer can reject order after delivery	30-day right to reject	Consumer protection rights
Rq.10	Orders can be picked by customer, delivered by store or preferred courier	Privacy when customer data is handed to third parties	Data privacy (GDPR)
Rq.11	Access to private data must be task driven	Sole access to data is to accomplish a task in the process	Data privacy (GDPR)
Rq.12	Authorization to data must be legitimized and legalized	Data access control and legitimization	Data privacy (GDPR)

Considering the discussed knowledge gaps in the previous section and the analysis from the case study, we illustrate the need for compliance verification and propose a design time compliance verification approach.

9.3 Supporting the Verification

9.3.1 Compliance with Data Privacy

The listed compliance requirements cannot be met by existing solutions due to limitations discussed (section 2). Compliance is embedded within the business process and verifying for conformance at design time to achieve proactive preventative compliance. Underpinning the approach is supporting end users to specify and verify collaborative business process for adherence to compliance constraints that are specified and verified. The proposed approach has three components i.e. the rules modeler, rules verifier and the rules enforcer.

i. Compliance rules modeler

Compliance rule modeler supports the extraction of requirements from their sources (policy statements, national and international laws, regulations and standards) and translates them into constraints based on compliance patterns. Some patterns are adopted and adapted from (Hall et al. 1998, Gammal 2014, Elgammal et al. 2016) as presented in the meta model (figure 3).

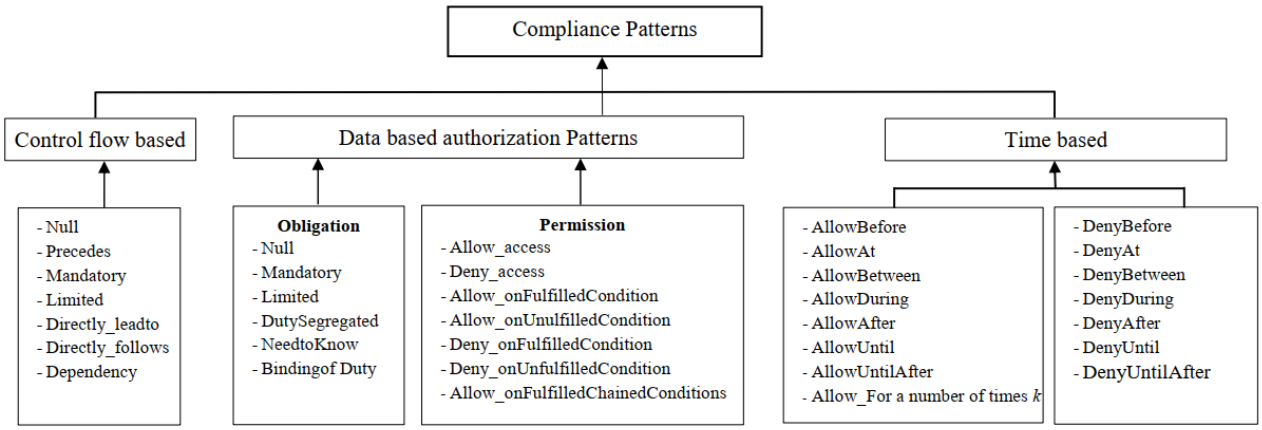

Figure 2: Compliancy Patterns Meta-model

Patterns offer reusable solutions to recurring problems to increase the productivity and quality of the design (Hall et al. 1998). In this respect, the meta-model is composed of patterns intended to support the design of the compliancy requirements verification approach. For automated application, the patterns are formalized to achieve formal semantics and syntax based on temporal logic languages. Their application will involve; 1) choice of the right pattern, 2) instantiation through adaption, 3) generate the implementation (Albin-Amiot and Guéhéneuc 2001)

ii. Compliance rules verification

To our knowledge none of the existing framework supports capability. It is intended to ensure coherent, accurate, complete and consistent compliance constraints. Cases of conflicts between compliance constraints are likely to exist and thus it is necessary to verify them before enforcement. For example, internal policies may conflict with external regulations. If unchecked, conflicts may create deadlocks or live-locks that may prevent process execution. Consistency is required between; internal policies and collaborative policies and, internal policies and national regulatory policies.

The internal policies will be translated into requirements i.e. properties to be satisfied while the external policies into system models using temporal logic, then apply formal reasoning and model checking techniques to support automatic verification amongst them. The intention is to derive a

state where both internally and externally derived constraints can be used to constrain a business process without inbound conflicts, ambiguities and inconsistencies. Some of the targeted error checks relate to resource authorization and access control that would otherwise be a source of flaws and insecurity in the business process; for instance (Tan et al. 2004).

- Privilege leakage – access to resources prohibited by safety requirements
- Privilege locking – fault that blocks privilege from having legitimate access to a resource
- Privilege conflict – direct or indirect assignment of one or more privileges that conflict each other
- Cyclic inheritance – inherited privileges that connect back to other inherited privileges leading cyclic redundancy.

Verification will be achieved by integrating with existing model checkers. Specifically, NuSMv a version of the traditional SMV model checker is preferable for its expressive power in checking models for satisfiability to constraints. A verification report will be generated to the user in human understandable format indicating satisfiability i.e. lack existence of the stated errors above. If non-compliance exists, violations will be traced to source and re-verified until satisfiability is achieved.

iii. The enforcement component

Verified compliance requirements are enforced on the business process activities constraining them to satisfy conformance. For instance, to achieve privacy, access to data is controlled and authorized based on its need to accomplish a time bound activity in the business process i.e. access is legitimized. In such scenario, during runtime the task will invoke the authorization API seeking access to a specific data item. The authorization engine will then check its access policy repository built according to the access control policy. Whatever the request outcome, the task will progress, halt, terminate or be skipped for the business process to progress too completion.

9.3.2 Application to the Case Study

To show a minimal illustration of the application of a section of the proposed compliancy approach, we use the case study highlighting the compliancy requirements and how the proposed design patterns can be mapped to support automated checking.

Table 1: Application of Some Compliance Patterns

Req.	Pattern category	Applicable Pattern(s)	Condition	Otherwise
Rq.1	Auth. Task Time	Mandatory Allow_onFulfilledCondition AllowAfter	Upon successful system registration	Deny access DenyUntil
Rq.2	Auth. Task Time	Precedes Rq.1 Allow_access AllowAfter	- On order submission	Deny access DenyAt
Rq.3	Auth. Task Time Role	Mandatory Allow_access AllowAt NeedtoKnow	On order submission	Deny access DenyAt
Rq.4	Task Time Role	Allow_onUnfulfilledCondition AllowUntil NeedtoKnow	Until communication to customer is made Until order change or cancelling	Deny access DenyAt
Rq.5	Auth. Task Time Role	Mandatory Allow_onFulfilledChainedConditions AllowDuring NeedtoKnow	During communication to customer, order change or cancelling	Deny_onUnfulfilledChainedCondition DenyDuring
Rq.6	Auth. Task Time Role	Mandatory Allow_access AllowBetween BindingofDuty	Item picking and initial verification execution	Deny_onUnfulfilledCondition DenyDuring
Rq.7	Auth. Task Time Role	Dependency R9 Allow_onFulfilledChainedConditions AllowBetween DutySegregated	Duties concerning final verification of orders	Deny_onUnfulfilledCondition DenyBetween
Rq.8	Task Time Role	Allow_UnFulfilledCondition AllowAt NeedtoKnow	Order/item rejection or cancellation	Deny_onUnfulfilledCondition DenyAt
Rq.9	Auth. Task Time Role	Limted Allow_UnFulfilledCondition AllowAt NeedtoKnow	Order/item rejection or cancellation	Deny_onUnfulfilledCondition DenyAt
Rq.10	Auth. Task Time Role	Directly_follows Rq.3/ Limited Allow_onFulfilledChainedConditions AllowDuring Limited	Check third party access rights,	Deny_onUnfulfilledCondition DenyUntilAfter
Rq.11	Auth. Task Time Role	Mandatory, Allow_onFulfilledChainedConditions AllowDuring Limited	Check third party access rights, intent, consent and legality	NeedtoKnow Deny_onUnfulfilledChainedCondition DenyUntilAfter
Rq.12	Task Time Role	Mandatory Allow_onFulfilledChainedConditions AllowDuring Null	Check third party access rights, consent and legality	NeedtoKnow Deny_onUnfulfilledChainedCondition DenyUntilAfter

To the automation the application especially for the non-expert end-users as illustrated in table 4, a declarative approach will be adopted for implementation where all possible combinations of patterns as well as executions or behavior are implicitly permissible except where they are explicitly forbidden i.e. by stating what is non-permissible.

10. Conclusion

Verification is a way to ensure error free business process models at execution time. The existing research reveals a lot of work done in process modelling and verification in form of theories, approaches, tools and methodologies but realizable gaps still exist. Verification of single organization processes is well addressed in literature but work remains at large concerning techniques and tools specific for verification of cBPs more so in a vF environment. The nature of cBPs in vF relies on data that enables real-time actionable intelligence. Supported data analytics present the potential to increase productivity, undertake preventive maintenance through projected breakdowns and generate cost savings. A recommendation for a verification method specific to cBPs in a vF environment is appropriate to meet the expressiveness, flexibility, suitability and Limitations that is required in such environment given its requirements as discussed in the report.

Compliance is a major concern today regardless of the industrial sector given the rising concerns of security, product and service quality and data privacy. With the EU revising its GDPR set to commence by May 2018; concerned organizations are working towards meeting its requirements before deadline by realigning their business processes. To support them in the due course is a welcome and necessary step. For doing so, other than the detective after-the-effect compliance checking, a proactive preventive approach is preferred to identify and combat compliancy violations before they take place to avoid the costs of fines or litigations. The effort of this research is geared towards a comprehensive approach for modeling, verification and enforcement of compliance constraints on collaborative business processes with an end user perspective.

References

Aalst, W. M. P. Van Der, 1997. Verification of Workflow Nets.

Aalst, W.M.P. van der, 2000. Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques. Business Process Management, 1806, 19–128.

Aals,t W.M.P. van der, Ter Hofstede, A. ., 2000. Verification of Workflow Task Structures: A Petri-net-based approach. Information systems, 43–69.

Aalst, van der, W. and van Hee, K., 2004. Workflow Management.

Aalst, W. van der, 2000. Loosely coupled interorganizational workflows: modeling and analyzing workflows crossing organizational boundaries. Information & Management [online], 37 (2), 67–75. Available from: http://www.sciencedirect.com/science/article/B6VD0-3YJ9Y2V-2/2/d9c28a0dfa2816dcd7f419de6a56d7cf\nhttp://www.sciencedirect.com/science/article/pii/S0378720699000385

Aalst, van der, W. M. P. and Pesic, M., 2006. DecSerFlow: Towards a Truly Declarative Service Flow Language. Web Services and Formal Methods [online], 4184, 1–23. Available from: http://link.springer.com/chapter/10.1007/11841197_1

Aalst W.van der, 2004. Business Process Management Demystified: A Tutorial on Models, Systems and Standards for Workflow Management. Business process management demystified: A tutorial on models, systems and standards for workflow management, 3098, 1–65

Aalst W. van der, Kiepuszewski, B., and Hofstede, A., 2003. Workflow Patterns. Distributed and Parallel Databases [online], 14 (1), 5–51. Available from: http://link.springer.com/article/10.1023/A:1022883727209.

Adamides, E. D. and Karacapilidis, N., 2006. A knowledge centred framework for collaborative business process modelling. Business Process Management Journal, 12 (5), 557–575.

Aiello, M., Bulanov, P., and Groefsema, H., 2010. Requirements and tools for variability management. Proceedings - International Computer Software and Applications Conference, (July), 245–250

Albin-Amiot, H. and Guéhéneuc, Y., 2001. Meta-modeling design patterns: Application to pattern detection and code synthesis. *First ECOOP Workshop Automating Object-Oriented Software Development Methods* [online], (January 2001), 1–8. Available from: http://www-etud.iro.umontreal.ca/~ptidej/yann-gael/Work/Publications/Documents/ECOOP01AOOSDM.doc.pdf.

Alshehri, A. and Sandhu, R., 2017. Access Control Models for Virtual Object Communication in Cloud-Enabled IoT.

In: *2017 IEEE International Conference on Information Reuse and Integration (IRI)*.

Alur, R., Henzinger, T. A., and Ho, P., 1996. Automatic Sybolic Verification of Embedded Systems. IEEE Transactions on Software Engineering, 22 (3), 2–11.

Anderson, B. B., Hansen, J. V, Lowry, P. B., and Summers, S. L., 2005. Model Checking for E-Business Control and Assurance, 35 (3), 445–450.

Axiomatics, 2018. *Attribute Based Access Control (ABAC)* [online]. Available from: https://www.axiomatics.com/attribute-based-access-control/ [Accessed 9 Apr 2018].

Baier, C. and Katoen, J.-P., 2008. Principles Of Model Checking [online]. MIT Press. Available from: http://mitpress.mit.edu/books/principles-model-checking

Bérard, B., Bidoit, M., François, A. F., Antoine Petit, L., Petrucci, L., Schnoebelen, P., and Pierre, M., 2001. HYTECH — Linear Hybrid Systems.pdf

Borrego, D. and Barba, I., 2014. Conformance checking and diagnosis for declarative business process models in data-aware scenarios. *Expert Systems with Applications* [online], 41 (11), 5340–5352. Available from: http://dx.doi.org/10.1016/j.eswa.2014.03.010.

Chiotti, P. D. V. L. R., 2010. A Modeling Approach for Collaborative Business Processes Based on the UP-ColBPIP Language. Business Process Management Workshops..

Cimatti, A., Clarke, E., and Giunchiglia, E., 2002. Nusmv 2: An opensource tool for symbolic model checking. Computer Aided Verification [online], 2404, 359–364. Available from: http://link.springer.com/chapter/10.1007/3-540-45657-0_29.

Clarke, E. M., Emerson, E. a., and Sistla, a. P., 1986. Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Transactions on Programming Languages and Systems, 8 (2), 244–263.

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

Dadam, P. and Reichert, M., 2009. The ADEPT project: A decade of research and development for robust and flexible process support : Cllenges and Achievements. Computer Science - Research and Development, 23 (2), 81–97.

Daws, C., Olivero, A., Tripakis, S., and Yovine, S., 1996. The tool kronos. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1066, 208–219.

De Backer, M., Snoeck, M., Monsieur, G., Lemahieu, W., and Dedene, G., 2009. A scenario-based verification technique to assess the compatibility of collaborative business processes. Data and Knowledge Engineering, 68 (6), 531–551

Dijkman, R. and Dumas, M., 2004. Service-oriented design: A multi-viewpoint approach. International journal of cooperative information systems, 13(04), pp.337-368.

Elgammal, A., Turetken, O., van den Heuvel, W. J., and Papazoglou, M., 2016. Formalizing and appling compliance patterns for business process compliance. *Software and Systems Modeling*, 15 (1), 119–146.

Ertugrul, A. M. and Demirors, O., 2015. An exploratory study on role-based collaborative business process modeling approaches. *In*: *Proceedings of the 7th International Conference on Subject-Oriented Business Process Management - S-BPM ONE '15* [online]. 1–5. Available from: http://dl.acm.org/citation.cfm?doid=2723839.2723857.

Falkenberg, E., Hesse, W., and Lindgreen, P., 1998. A framework of information systems concepts [online]. Ifip Wg. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.1492&rep=rep1&type=pdf.

Fahland, D., Luebke, D., Mendling, J., Reijers, H., Weber, B., Weidlich, M., and Zugal, S., 2009. Declarative versus Imperative Process Modeling Languages: The Issue of Understandability. Enterprise, Business-Process and Information Systems Modeling [online], 29, 353–366. Available from: <Go to ISI>://WOS:000268581700029.

Fdhila, W., Rinderle-Ma, S., Knuplesch, D., and Reichert, M., 2015. Change and Compliance in Collaborative Processes. Proceedings - 2015 IEEE International Conference on Services Computing, SCC 2015, 162–169.

Foundation, T. Y., 2016. YAWL - User Manual.

Gammal, E., 2014. Towards a comprehensive framework for business process compliance FRAMEWORK FOR BUSINESS PROCESS.

Gautam, M., 2017. Poster : Constrained Policy Mining in Attribute Based Access Control, 121–123.

Giannakopoulou, D. and Havelund, K., 2001. Automata-based verification of temporal properties on running programs. Proceedings 16th Annual International Conference on Automated Software Engineering (ASE 2001), (August), 412–416

Goedertier, S., 2008. Declarative Techniques for Modeling and Mining Business Processes. [online], (284), 248. Available from: https://lirias.kuleuven.be/handle/1979/1908.

Goedertier, S. and Vanthienen, J., 2006. Designing Compliant Business Processes with Obligations and Permissions. *BPM 2006 International Workshops, BPD, BPI, ENEI, GPWW, DPM, semantics4ws, Vienna, Austria, September 4-7, 2006*. [online], 5–14. Available from: http://link.springer.com/10.1007%2F11837862_2.

Goedertier and Vanthienen, 2007. Compliant and Flexible Business Processes with\nBusiness Rules. *Bpmds* [online], (January), 94–103. Available from: https://lirias.kuleuven.be/handle/123456789/103560%5Cnhttp://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-236/paper3.pdf.

Gottschalk, F., van der Aalst, W. M. P., Jansen-Vullers, M. H., and Verbeek, H. M. W., 2008. Protos2CPN: Using colored Petri nets for configuring and testing business processes. International Journal on Software Tools for Technology Transfer, 10 (1), 95–110.

Groefsema, H. and Bucur, D., 2013. A Survey of Formal Business Process Verification: From Soundness to Variability. Proceedings of the Third International Symposium on Business Modeling and Software Design, 198–203.

Guynn, J., 2018. *Facebook CEO Mark Zuckerberg finally speaks on Cambridge Analytica: We need to fix 'breach of trust'* [online]. Tech. Available from: https://www.usatoday.com/story/tech/2018/03/21/facebook-ceo-mark-zuckerberg-finally-speaks-cambridge-analytica-we-need-fix-breach-trust/445791002/ [Accessed 12 Apr 2018].

Hall, N., Dwyer, M. B., Avrunin, G. S., and Corbett, J. C., 1998. Property Speci cation Patterns for Finite-State Veri cation 1 INTRODUCTION 2 DESIGN AND OTHER PATTERNS. *n Proceedings of the second workshop on Formal methods in software practice*, 2, 7–15.

Havelund, K. and Rosu, G., 2001. Monitoring Programs rising Rewriting. Automated Software Engineering, (16th Annual International Conference), 135–143.

Henzinger, T. A. and Wong-toi, H., 1997. HyTech : A Model Checker for Hybrid Systems 1 Introduction. International Journal on Software Tools for Technology Transfer (STTT), 1 (1997), 110–122.

Henzinger, T. A., Horowitz, B., and Majumdar, R., 1999. Beyond HyTech :, 89–95.

Holzmann, G. J., 1997. The model checker SPIN. IEEE Transactions on Software Engineering, 23 (5), 279–295.

Holzmann, G., 2017. The Design and Validation of the CLASS.pdf, (March).

Holzmann, G. J., Godefroid, P., and Pirottin, D., 2013. Coverage preserving reduction strategies for reachability analysis. [online], 6, 349–363. Available from: https://books.google.com/books?hl=en&lr=&id=Q1EvBQAAQBAJ&oi=fnd&pg=PA349&dq='establish the correctness of systems of interacting concurrent processes by anSections 3 and 4 discuss the foundation for a partial order semantics'.

Hommes, B. J., 2004. The Evaluation of Business Process Modeling Techniques.

Jain, S., Choong, N. F., Aye, K. M., and Luo, M., 2001. Virtual factory: an integrated approach to manufacturing systems modeling. International Journal of Operations & Production Management, 21 (5/6), 594–608.

Jensen, K., Kristensen, L. M., and Wells, L., 2007. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. International Journal on Software Tools for Technology Transfer, 9 (3–4), 213–254.

Kadono, M., Tsuchiya, T., and Kikuno, T., 2009. Using the NuSMV model checker for test generation from statecharts. 2009 15th IEEE Pacific Rim International Symposium on Dependable Computing, PRDC 2009, 37–42.

Kasse JP, Xu L, de Vrieze P (2017) A Comparative Assessment of Collaborative Business Process Verification Approaches. In:18th IFIP Working Conference on Virtual Enterprises (PRO-VE 2017), Vicenza, Italy, 18 Sep 2017 - 20 Sep 2017. Springer.

Kasse JP, Xu L, de Vrieze P, Bai Y, (2018) The Need for Compliance Verification in Collaborative Business Processes. In:19th IFIP Working Conference on Virtual Enterprises (PRO-VE 2018), Cardiff, UK, 17-19 Sep 2018. Springer.

Khan, A. R., 2012. Access control in cloud computing environment. *ARPN Journal of Engineering and Applied Sciences*, 7 (5), 613–615.

Kim, S. and Smari, W. W., 2006. A Petri Net-based Workflow Modeling for a Human-centric Collaborative Commerce System, 5 (Cd), 28–31.

Kochanowski, M., Fehling, C., Koetter, F., Leymann, F., and Weisbecker, A., 2014. Compliance in BPM today - an insight into experts ' views and industry challenges. *Informatik 2014. Big Data - Komplexität meistern.*, 769–780.

Koliadis, G., 2007. Verifying Semantic Business Process Models in Verifying Semantic Business Process Models in

Inter-operation.
Knuplesch, D., Reichert, M., Fdhila, W., and Rinderle-Ma, S., 2013. On enabling compliance of cross-organizational business processes. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8094 LNCS, 146–154.
Larsen, K. G., Pettersson, P., and Yi, W., 1997. U PPAAL in a nutshell, 134–152.
Larsen, K. G., Pettersson, P., and Yi, W. Y. W., 1995. Compositional and symbolic model-checking of real-time systems. Proceedings 16th IEEE Real-Time Systems Symposium, 76–87.
Lowe, G., 2008. Specification of communicating processes: Temporal logic versus refusals-based refinement. Formal Aspects of Computing, 20 (3), 277–294.
Lu, R. and Sadiq, S., 2007. A survey of comparative business process modeling approaches. Proceedings of the 10th International Conference on Business Information Systems (BIS2007) [online], 82–94. Available from: http://www.springerlink.com/index/82M6138P17R5G732.pdf.
Morimoto, S., 2008. A Survey of Formal Verification for Business Process Modeling. Computational Science – ICCS 2008 - Lecture Notes in Computer Science [online], 5102, 514–522. Available from: http://www.springerlink.com/index/96x0051124530845.pdf
Müller, J., 2015. Security Mechanisms for Workflows in Service-Oriented Architectures.
Norta, A., Grefen, P., and Narendra, N. C., 2014. A reference architecture for managing dynamic inter-organizational business processes. Data and Knowledge Engineering [online], 91, 52–89. Available from: http://dx.doi.org/10.1016/j.datak.2014.04.001.
Petri, C. A., 1977. General Net Theory. Computing System Design: Proceedings of the Joint IBM-University of Newcastle upon Tyne Seminar, Sept. 1976 [online]. Available from: papers3://publication/uuid/80FAA443-1FAC-47FD-9828-B7D271194C80.
Pesic, M. and van der Aalst, W. M. P., 2006. A Declarative Approach for Flexible Business Processes Management. Business Process Management Workshops, 169–180.
Ramezani, T. E., Gromov, V., Fahland, D., and van der Aalst, W. M. P., 2014. Compliance Checking of Data-Aware and Resource-Aware Compliance Requirements. *On the Move to Meaningful Internet Systems: OTM 2014 Conferences. OTM 2014. Lecture Notes in Computer Science, vol 8841* [online], (2), 237–257. Available from: http://link.springer.com/10.1007/978-3-662-45563-0_14.
Reichert, M. and Dadam, P., 1998. ADEPTflex - supporting dynamic changes of workflows without losing control. Journal of Intelligent Information Systems, 10 (2), 93–129.
Roa, J., Villarreal, P., and Chiotti, O., 2011. A Methodology for the Design , Verification , and Validation of Business Processes in B2B Collaborations. International Conference on Business Process Management. Springer Berlin Heidelberg., 293–305.
Robol, M., Salnitri, M., and Giorgini, P., 2017. Toward GDPR-compliant socio-technical systems: Modeling language and reasoning framework. *Lecture Notes in Business Information Processing*, 305, 236–250.
Roşu, G., Chen, F., and Ball, T., 2008. Synthesizing monitors for safety properties: This time with calls and returns. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5289 LNCS, 51–68.
Sadiq, S. and Governatori, G., 2010. Managing Regulatory Compliance in Business Processes. *Handbook on Business Process Management 2*, 2008, 159–175.
Sadiq, S., Governatori, G., and Namiri, K., 2007. Modeling Control Objectives for Business Process Compliance. *5th International Conference, BPM 2007, Brisbane, Australia, September 24-28, 2007.*, 149–164.
Salnitri, M., Dalpiaz, F., and Giorgini, P., 2014. Modeling and verifying security policies in business processes. *In*: *Lecture Notes in Business Information Processing*. 200–214.
Sandhu, P. R., 2003. The RBAC96 Model.
Sandhu, R., 1996. Rationale for the RBAC96 family of access control models. *Proceedings of the first ACM Workshop on Role-based access control - RBAC '95* [online], (1), 9–es. Available from: http://portal.acm.org/citation.cfm?doid=270152.270167.
Schulz, K. A. and Oklowska, M. E., 2004. Facilitating cross-organisational workflows with a workflow view approach. *Data and Knowledge Engineering*, 51 (1), 109–147.
Schumm, D., Leymann, F., Ma, Z., Scheibler, T., and Strauch, S., 2010. Integrating Compliance into Business Processes Process Fragments as Reusable Compliance Controls, 2125–2137.

Tan, K., Crampton, J., and Gunter, C. A., 2004. The Consistency of Task-Based Authorization Constraints in Workflow Systems. *Proceedings of the 17th IEEE Computer Security Foundations Workshop*, 155–169.

Telang, P. R. and Singh, M. P., 2012. Specifying and verifying cross-organizational business models: An agent-oriented approach. *IEEE Transactions on Services Computing*, 5 (3), 305–318.

Thomas, R. K. and Sandhu, R. S., 1997. Task-based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management. *Database Security* [online], 11, 166–181. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.54.6227&rep=rep1&type=pdf.

Varea, M., 2002. Mixed Control / Data-Flow Representation for Modelling and Verification of embedded systems.

Verbeek, H. M. W., Basten, T., and Van Der Aalst, W. M. P., 2001. Diagnosing workflow processes using Woflan. Computer Journal, 44 (4), 246–279.

Wasim, S. and Maria, O., 2000. Applying Graph Reduction Techniques for Identifying Structural Conflicts in Process Models. Advanced Information Systems Engineering, 1789 (November 2016), 431–445.

Weber, B., Reichert, M., and Rinderle-Ma, S., 2008. Change patterns and change support features - Enhancing flexibility in process-aware information systems. Data and Knowledge Engineering, 66 (3), 438–466.

Weber, B., Reichert, M., Rinderle-Ma, S., and Wild, W., 2009. Providing Integrated Life Cycle Support in Process-Aware Information Systems. International Journal of Cooperative Information Systems, 18 (1), 115–165.

Wenbin, Z., Xiumin, F., Yan, Juanqi, and Zhu, 2002. An Integrated Simulation Method to Support Virtual Factory Engineering. International Journal, 2 (1), 39–44.

Wu, M., 2007. Role and Task Based Authorization Management for Process-View. *Proceedings of the Second International Conference on Security and Cryptography* [online], (707), 85–90. Available from: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0002126300850090.

Yuan, E. and Tong, J., 2005. Attributed Based Access Control (ABAC) for Web Services. *In*: *The IEEE International Conference on Web Services*. 561–569.

Yovine, S., 1997. Kronos: A verification tool for real-time systems. International Journal on Software Tools for Technology Transfer, 1 (1–2), 123–133.or Technology Transfer, 1 (1–2), 123–133. Yovine, S., 1997.

Ziemann, J. and Matheis, T., 2007. Modelling of cross-organizational business processes-current methods and standards. *Proc. EMISA '07* [online], 2 (2), 87–100. Available from: http://doc.utwente.nl/64399/1/EMISA-Proceedings.pdf#page=87.