A comparative Assessment of Collaborative Business Process Verification Approaches

John Paul Kasse[1], Lai Xu[1], Paul de Vrieze[1]

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

Abstract. Industry 4.0 is a key strategic trend of the economy. Virtual factories are key building blocks for Industry 4.0 where product design processes, manufacturing processes and general collaborative business processes across factories and enterprises are integrated. In the context of EU H2020 FIRST (vF Interoperation suppoRting buSiness innovaTion) project, end users of vFs are not experts in business process modelling to guarantee correct collaborative business processes for realizing execution. To enable automatic execution of business processes, verification is an important step at the business process design stage to avoid errors at runtime. Research in business process model verification has yielded a plethora of approaches in form of languages and tools that are based on Petri nets family and temporal logic. However, no report specifically targets and presents a comparative assessment of these approaches based on criteria as one we propose. In this paper we present an assessment of the most common verification approaches based on their expressibility, flexibility, suitability and complexity. We also look at how big data impacts the business process verification approach in a data-rich world.

Keywords: Petri nets, Temporal Logic, Collaborative Business Process, big data, Virtual Factory

1 Introduction

Virtual factories (vF) arise out of the amalgamation of distributed manufacturing, virtual enterprises, and business management. A vF describes a distributed and integrated computer-based model simulating total manufacturing environment. It incorporates all the tasks and resources necessary to accomplish the operation of designing, producing and delivering a product [1][2]. 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 (cBP). Individual enterprise business processes integrate into a cBP jointly designed and implemented. The pool of skills, resources and technology is exploited to support the analysis of different design alternatives, performance evaluation and reduced time-to-production.

cBPs are complex; they are dynamic, cross organizational boundaries and rely on data from partners for their design and execution. They differ from single organization business processes (sBP) in nature and structural design [3][4] more so in virtual environments where execution is automated. It is resonant therefore to verify cBPs before their implementation to avoid errors at execution. We posit that verification should be supported with canonical approaches. Literature is scanty concerning approaches and tools applicable to verify cBP models especially for vF yet sBP verification has been well addressed with various approaches [5][6][7][8][9] [3][10][11]. However, these approaches present realizable knowledge gaps; they concentrate on control flow aspects [3][10][11][9][4][4][12][14][15] and abstract from other perspectives like data which is a major input for smart devices and machines in a vF. Besides, best practice linking verification approaches to vF cBPs is missing. The EU H2020 FIRST project aims to develop a method to support non expert end users to model and verify vF cBPs.

This paper presents the state of art in business process verification approaches and makes a comparative assessment of their fitness to verify vF cBPs. The vF cBPs being data intensive, we describe their requirements and how to support their verification in a vF environment.

The rest of the paper is structured as follows; section 2 presents the requirements for cBP verification by describing their characteristics in a vF setting, section 3 presents the state-of-art in process verification. In section 4 we present a framework of cBP verification while section 5 discusses the related work. We make conclusions and outlook for future work in section 6.

2. Requirements of cBP Model Verification

For us to support cBP verification, it is objective to understand their nature and requirements. We postulate that cBPs should conform to a set of requirements described below;

Span different organisations: Collaboration involves different partners working together for a common goal. In terms of business process management the partners 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 [16]. The approach to verify such processes should consider the diversity of users, their roles and the distributed nature of the cBP.

Communication/Interaction Protocol: Typical of the cBP are the forms of communications and interactions expressed as message exchanges among partners who engage in discussions before reaching a decision. cBPs require dedicated interaction protocols through which partners can communicate. Various interaction protocols are proposed [4] [17] but they do not pass the criteria to support cBP verification.

Dynamism, Flexibility and Complexity: CBPs may be composed from services offered by partners using Service oriented architecture. The partners timelessly and continuously push in changes that impact on the process outcome. Such volatility

should be taken care or at design time through verification to prevent execution flaws and support change integration, propagation and continuous verification.

Data requirements: cBP data requirements relate to several issues to support operations and analytics for decision making. Workflow systems embraced 2 kinds of data i.e. control data (for routing purposes) and production data (information objects like documents, forms, and tables) [8]. For the smart factory data is exploited by the cyber physical systems, internet of things and cloud computing to support operations by autonomously exchanging information, triggering actions and controlling operations. Factory automation relies on intelligent data gathering and exchange between the systems. Verification should cater for data requirements and data patterns to support analytics for decision making and driving operations at the factory floor. A verification tool should be able to support verification of such requirements.

The following section presents the state of art in business process verification by discussing how existing approaches and tools compare in regard to supporting cBP model verification given the suggested requirements.

3 State of the art in Business Process Verification

3.1 Business Process Verification Approaches

Business process verification as a concept of model checking (MC) has various applications; i.e. Variability - checking to ascertain how business processes vary in behavior over a set of conditions [12] [18]. Compliance - model conformance with requirements, laws or standards [13][14][15][19][20] [21]. Compatibility - aligning partner processes to the choreography i.e. the interaction architecture through which the cBP is executed [18] [22]. Verification – checking models to correct errors. During business process design, more time is spent on verification than actual design. Formal verification leads to seminal advantages as described in [19] and [23].

Various verification approaches exist along with supporting tools. This section presents a description of some of the most commonly applied verification approaches and tools in literature. The tools are broadly categorized according to the technique or language on which they are semantically based .i.e. *Petri nets and Temporal Logic*. Petri nets describe a bipartite directed graph with two nodes i.e. Places (circles) and Transitions (rectangles) [24] connected by directed arcs. Petri nets are applied in workflows to create Workflow Nets. 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 [3] [11] [25] [26]. For details on petri nets and workflow verification the reader is referred to [6]. Classical petri nets become very large, inaccessible and difficult to interpret [3]. The color petri nets solve the limitations. A discussion of some of the most common petri net based tools follows;

Colored Petri Nets (CPN Tools); support modelling of data, objects and structures using color [27] and support verification [28][29]. Color expresses each instance as unique in a case, time captures time related information to track capacity of a process,

and hierarchy supports hierarchical design of process models and sub models. CPN tools integrate with other tools to support verification of models, for instance Protos and E-C-A [29].

Yet Another Workflow Language (YAWL); it is both a workflow modelling and verification tool based on the Petri nets [25] and workflow patterns [30]. It provides a graphical editor enhanced with built-in verification functionality supporting early time detection of model errors. It provides support for verification based on Reset nets and transition invariants (WofYAWL editor plug-in) [31]

Protos: Protos supports process model definition and analysis based on different perspectives of data and control flow. It supports simulation of models before their enactment and execution. Protos2CPN tool is an integration of Protos with CPN tools to support process model verification [29].

FlowMake: The tool supports design time identification of errors in models before implementation in WFMS [32]. Graph reduction algorithm [33] is employed to verify workflows for syntactic correctness 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.

Application Development based on Encapsulated pre-modelled Process Templates (ADEPT)/AristaFlow; a family of tools used to support modelling and verification of flexible and dynamic business processes [34] [35][36][37]. Based on clinical business scenarios, ADEPT enables process implementers, application developers and end users to model and verify models through its features like; extended graphical interfaces, on-the-fly correctness checks [37], process templates and structural transformation of processes. It supports for ad-hoc changes and their propagation.

The tools described from this point are based on Temporal Logic formalism. 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 [20][38]. It is widely applied to verify concurrent systems, distributed systems, context aware and collaborative systems. For details on temporal logic and its various branches and application the reader is referred [39].

Declarative Service Flow Language (DecSerFlow): DecSerFlow supports specification, enactment, and monitoring of service flows in a declarative nature. Verification of service workflow conformance is achieved by subjecting models to temporal logic constraints enforced by the engine and guarded against their violations. The engine monitors the violations as well [11][14] .

HYbrid TECHnology (HyTech); supports automatic verification of system models against properties specifications expressed in real time temporal logic through symbolic computation [7]. Models are verified for reachability, liveness, time boundedness and duration properties [40]. HyTECH is recommended for verification of mission critical systems. However, the tool is limited to verification of small systems [41] and linear hybrid systems [42]. Some of the limitations have been overcome by HyTECH+ tool [42] which is an extension to the classical HyTECH.

Symbolic Model Verifier (SMV); operates like HyTECH to exhaustively verify models and provide counter examples. It is limited by state explosion. NuSMV is a modified version to verify synchronous finite-state and infinite-state systems [43][44].

SPIN; supports verification of asynchronous systems by verifying for correctness. The properties are expressed as standard temporal logic while model specifications 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, partial order reduction method is employed [25][45][46][47]. However state explosion remain a challenge limiting applicability to verify cBP model.

KRONOS; applies timed automata and timed temporal logic to verify models for reachability properties [48][49] 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; 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 [5][50] [51]. State explosion remains a challenge limiting its application to cBP model verification. Table 1 presents a summary of the tools and the related properties that they verify in business process models.

Table 1 showing summary of tools and properties

Language\Technique	Tool	Properties
Petri Nets	Woflan	Soundness (deadlocks, reachability and liveness)
	YAWL	Soundness and liveness
	FlowMake	Synchronization, Deadlocks, consistency, boundedness, liveness
	CPN Tools	Performance analysis, coverability and occurrence
	Protos2CPN	Soundness and liveness
Temporal logic	SPIN	Correctness and logical consistency
	UPPAAL	Bounded liveness and deadlocks
	KRONOS	Reachability (safety and bounded response)
	SMV\NuSMV	Correctness, safety, and liveliness
	HyTECH	Reachability, safety, liveness, time-bounded, duration
	ADEPT	Semantic correctness, deadlock and Safety
Petri Nets & Temporal logic	DecSerFlow	Constraints and their variations

3.2 Limitations of the Verification Approaches to Verify cBP Models

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

Support sBP verification; existing approaches were developed to support modelling and simulation of single organization business processes, not cBPs. Simulation is not

an exhaustive way to verify models since it is based on assumptions that may deviate from actual. Some tools integrate with other tools to support verification (e.g Protos and E-C-A integrate with CPN tools) [29] while others like YAWL verify models modelled in the same language. Woflan was created as an independent verification tool to verify models developed in Staffware, COSA and MQ [52]. These tools remain limited for vF cBP verification.

The semantical and architectural structure: The tools do not support the semantical structure and architectural requirements for cBP verification i.e. the lack of interfaces or open structures to permit integration with other systems manifests their inability to support collaborative environments. YAWL avails web based plugins for integration with other systems but support for simultaneous model and sub models verification is limited. Additionally the semantical structure of other tools is ambiguous and a source of semantical errors and conflicts when models are merged for verification [53].

Support for data and data analytics: Most approaches target verification based on control flow perspective and abstract from other perspectives like data, resources, tasks and applications [6][8] [9][52][16]. The justification advanced for abstraction never anticipated future data requirements that vF processes currently present. The smart factory heavily relies on data routed between interconnected smart devices to drive the automated machines at the factory floor. Moreover, data is used to support analytics for other seminal benefits like decision making, projections and future planning. Therefore during verification data and data analytics should be supported at both design time and runtime.

4   Framework for Assessing cBP model Verification Approaches

4.1 Assessment Criteria

Language comparisons are based on different factors that may be objective or subjective [53]. A set of parameters to compose our criteria intended to assess the inherent traction and precision of the verification approaches and their appropriateness to verify vF cBP models. The following section briefly describes these parameters;

Expressibility; assesses the degree to which an approach can represent any number of models in different application domains [54][55]. In [33], 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 tool in terms of the degree to which it supports 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 or runtime, and support for scalability especially as the cBPs evolve and grow.

Suitability; describes the appropriateness of a tool to a particular application domain [54][5]. In our case we assess suitability in terms of the degree to which a tool 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 and Limitations; assesse the level of difficulty an approach presents to work with [33] or its features that make it easy to work with while being used to verify process models. The limitations refer to the different forms of inadequacies of a tool that render it inappropriate and inapplicable to verify vF collaborative business process models.

4.2 Application of the Assessment Framework
This section presents the application of the assessment framework criteria to assess the existing verifications approaches and tools. A summary of this assessment is presented in Table 2 after which we discuss the assessment results in the section that follows after;

Table 2 showing a Comparative assessment of the verification tools

Tool	Expressibility	Flexibility	Suitability	Complexity\ Limitation
Woflan	Control flow specific Non domain specific Imports models from other tools	No adhoc changes Verifies complete models.	Non-collaborative Single model verification	Graphical interface Hard to trace Errors
YAWL	Control flow specific Non domain specific Integrates data	Exception handling at Design time	Non-collaborative Single model verification	Extensible with web plugins Graphical interface
FlowMake	Control flow specific Integrates data Non-domain specific	Supports exception handling Non scalable	Non-collaborative Single model verification	Graphical interface Hard to trace Errors
CPN Tools	Non domain specific Concurrent systems Integrates data	Exception handling supported	Non-collaborative Integrates with other tools	Graphical interface
Protos2CPN	Control flow specific Integrates data	Static analysis Exception handling supported	Non-collaborative Integrates with CPN tools	Known application and user support Graphical interface
SPIN\ XSPIN	Non domain specific Viable for vF cBP Wide application	Exception handling supported	State explosion - smaller systems Non-collaborative	Complex syntax and semantics. Graphical interface Counter examples
UPPAAL	Non scalable Error traceability	Supports on-the-fly verification.	No support for data Non-collaborative	Graphical interface with supported tools
KRONOS	Unknown application to vF domain No support for data	Exception handling supported	Non-collaborative Single model verification Verifies smaller models	Graphical interface Counter examples
SMV\ NuSMV	Non- domain specific	Exception handling Supported	Non-collaborative Single model verification State explosion	Graphical interface Counter examples
HyTECH	No support for data integration	Exception handling not supported. Non scalable	Verifies smaller models State explosion Non-collaborative	Complex syntax and semantics Counter examples

| ADEPT | On-the-fly verification Integrates data | Exception handling supported | Lack of known application Single model verification | Process templates for easy creation of processes |
| DeScerFlow | Non-domain specific Control flow specific | Adhoc changes supported Plug & play style | Single model verification | Graphical interfaces Process templates for process creation |

Assessment based on the proposed criteria as summarized in table 2 and in reference to table 1 reveals various properties being checked by the existing tools. However, these properties are expressed in relation to single organization business processes. The interpretation and connotation of these properties may not be the same for cBPs: for instance having sound models for a single organization business 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 cBPs where the requirements differ. Moreover, there is no silver bullet solution; no single approach verifies all necessary properties for all situations. For example Petri net based tools like YAWL, Woflan, and CPN are lacking in terms of time based requirements for models. Temporal logic based tools like SPIN, KRONOS 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 difficulty to understand for the ordinary users. Above of all, the inability and inconsideration for data perspective leaves them inappropriate to verify cBPs that are highly data intensive. In summary, using the parameters in the proposed criteria we note the following in view of cBPs;

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 cBPs.

Flexibility - besides HyTECH, UPPAAL and Woflan, all other tools reviewed have the capability for exception handling, permitting ad hoc changes and scalability. Such attributes meet the requirements of cBPs that are highly variable and dynamic due to the diversity of process owners and environment in which they apply. However the tools verify already completely designed models. This renders them rigid and inflexible for application to cBPs [17].

Suitability - the techniques are inappropriate and not suitable for verification of vF cBP models. The tools support single model verification at a time which makes it difficult for cBPs that are composed of many sub models that are merged for verification. Lack of standardized semantics introduces semantical errors when models to be verified are developed from different tools. This further limits the application of these tools to vF cBP verification.

Complexity\ Limitations - most tools present graphical user interfaces making them easy for the non-expert users to apply. Moreover, temporal logic based tools provide counter examples where model errors exist. However, the provided counter examples are not a guarantee for correctness of the model. Besides, temporal logic expressions remain complex for non-expert users in the collaborative environments [40].

5   Related Work

This section presents work related to our study but we are keen to highlight how our work differs.

In [33] a survey of comparative business process modelling approaches is presented based on graph based vs rule based approaches. The comparison criteria included parameters like expressibility, flexibility, adaptability, dynamism and complexity, and an analysis of how the approaches score was presented. The work is presented under the umbrella of process modelling while ours is based on supporting cBP model verification in a vF environment.

In [23], a survey of formal verification approaches for business process diagrams is presented and compared with respect to motivation behind their development i.e. the aim of verification, method of formalization and logics. This survey was based on verification of single organization business process models where our work concentrates on assessment of approaches that support cBP verification. Moreover they do not assess the tools based on their application or competency but rather on what motivated their developers. In our study the assessment is based on how well the approaches can support verification in a collaborative environment

Further work by [56] presents an analysis of verification tools based on the forms and application of verification by categorizing it into variability, compliance and compatibility. The approaches are then discussed and compared under the same breadth. Our work differs in a way that we propose and present an assessment framework to analyze verification tools based on their traction, precision and competency to verify cBP in vF environment.

6   Conclusion and Future Work

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

5   References

1. 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.
2. Wenbin, Z., Xiumin, F., Yan, Juanqi, and Zhu, 2002. An Integrated Simulation Method to Support Virtual Factory Engineering. International Journal, 2 (1), 39–44.
3. van der Aalst, W. and van Hee, K., 2004. Workflow Management.
4. 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.
5. Larsen, K. G., Pettersson, P., and Yi, W., 1997. U PPAAL in a nutshell, 134–152.
6. Aalst, W. M. P. Van Der, 1997. Verification of Workflow Nets.
7. Alur, R., Henzinger, T. A., and Ho, P., 1996. Automatic Sybolic Verification of Embedded Systems. IEEE Transactions on Software Engineering, 22 (3), 2–11.
8. W.M.P. van der Aalst, 2000. Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques. Business Process Management, 1806, 19–128.
9. W.M.P. van der Aalst, Ter Hofstede, A. ., 2000. Verification of Workflow Task Structures: A Petri-net-based approach. Information systems, 43–69.
10. 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.
11. Pesic, M. and van der Aalst, W. M. P., 2006. A Declarative Approach for Flexible Business Processes Management. Business Process Management Workshops, 169–180.
12. Varea, M., 2002.   Mixed Control / Data-Flow Representation for Modelling and Verification of embedded systems.
13. Adamides, E. D. and Karacapilidis, N., 2006. A knowledge centred framework for collaborative business process modelling. Business Process Management Journal, 12 (5), 557–575.
14. van der Aalst, 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.
15. 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.
16. 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.
17. Chiotti, P. D. V. L. R., 2010. A Modeling Approach for Collaborative Business Processes Based on the UP-ColBPIP Language. Business Process Management Workshops, NA.
18. Aiello, M., Bulanov, P., and Groefsema, H., 2010. Requirements and tools for variability management. Proceedings - International Computer Software and Applications Conference, (July), 245–250.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

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

25. 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.

26. van der Aalst, W. M. P., 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.

27. 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.

28. 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.

29. 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.

30. van der Aalst, W. M. P., 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.

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

32. 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.

33. 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.

34. Reichert, M. and Dadam, P., 1998. ADEPTflex - supporting dynamic changes of workflows without losing control. Journal of Intelligent Information Systems, 10 (2), 93–129.

35. 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.

36. 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.

37. 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.

38. Lowe, G., 2008. Specification of communicating processes: Temporal logic versus refusals-based refinement. Formal Aspects of Computing, 20 (3), 277–294.

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

40. 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.

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

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

43. 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.

44. 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.

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

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

47. 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' "the actions aff.

48. 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.

49. 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.

50. 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.

51. Bengtsson, J., Rs--, B., Larsen, K. G., and Yi, W., 1996. in 1995, (December).

52. 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.

53. Koliadis, G., 2007. Verifying Semantic Business Process Models in Verifying Semantic Business Process Models in Inter-operation.

54. 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.

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

56. 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.