# AN EFFIENT BPaaS MODELING APPROACH FOR THREE-STEPS CONFIGURATION AND TRANSACTIONAL BEHAVIOR VERIFICATION PROCESS

**Mr. Muthukuri Chiranjeevi, Mr. Katti Jaya Krishna[2*]**

*1 Final Year MCA Student, QIS College of Engineering and Technology, Ongole*
*2* Assistant Professor, MCA Dept., QIS College of Engineering and Technology, Ongole*

## Abstract

*Business Process as a Service (BPaaS) is a rising kind of cloud service that offers configurable and executable business processes to customers over the Internet. As BPaaS is still in the early years of research, many open issues remain. Managing the configuration of BPaaS expands on areas, for example, software product offerings and configurable business processes. The issue has worries to consider from several points of view, for example, the distinctive kinds of variable features, constraints between configuration choices, and satisfying the requirements given by the customer. In our approach, we utilize temporal rationale templates to inspire transactional requirements from customers that the arranged service must adhere to. For formalizing constraints over configuration, feature models are utilized. To manage all these worries amid BPaaS configuration, we build up an organized process that applies formal methods while coordinating customers through determining transactional requirements and choosing configurable features. The Binary Decision Diagram (BDD) analysis is then used to check that the chose configurable features don't violate any constraints. Finally, model checking is applied to confirm the arranged service against the transactional prerequisite set. We demonstrate the feasibility of our approach with several validation scenarios and performance evaluations.*

*Index Terms-- Business Process as a Service, formal methods, verification, transactional requirements, model checking.*

## I. INTRODUCTION

THIS In recent years, cloud services have had dramatic impacts in both the research [1] and industry [2] land-scapes of service-oriented computing. Cloud computing has become a popular paradigm for delivering a wide range of services, such as software applications, computing capacity, storage, and virtual platforms [3]. Cloud service providers can offer these utilities to clients over the Internet in a pay-by-use manner. The distinctive properties of cloud services include:

- On-demand availability through public or private network access, most commonly the Internet [3].
- Utilization of pooled resources such as servers, applications, CPU time, or storage.
- Dynamic response to workload by elastically pro-visioning and releasing resources [3][4].
- Configurability of service behavior of properties to meet individual client requirements [5][6][7].

The traditional hierarchy of cloud service types is comprised of three layers, where each layer can provide the base (infrastructure or platform) for running services within the layer above [3]. Infrastructure as a Service (IaaS) is the bottom service layer, providing access to virtualized physical resources, such as storage and computation capacity. Computing capacity offered by Amazon EC21 or IBM Smart Cloud Enterprise+[2] are examples of IaaS offerings. *Platform as a Service* (PaaS) provides access to utilities such as software development and hosting frameworks. For example, Google App Engine[3] and Microsoft Azure[4] both contain PaaS features for web application

development and hosting. Finally, *Software as a Service* (SaaS) is software applications deployed in a way that is Internet accessible, automatically scaling and multi-tenant. SaaS enables clients to remotely use soft-ware complex systems, such as customer relationship management through Salseforce[5].

### *A. BPaaS (Business Process as a Service)*

BPaaS is an emerging cloud service that provides business service to clients over the internet. As the service provided by companies on the internet is rapidly measuring, there is a need for a dynamic business environment. Since BPaaS is still in years of research, there are services like IaaS, PaaS, SaaS. BPaaS is considered as the next level of SaaS. It is a configurable and executable business process which provides software to the user as well as being the logic and control flow. In the rapidly changing ideas of economy, BPaaS has the potential to play a significantly bigger side. BPaaS has to address both business and IT alignment. In this system, BPaaS is used as a third party application. It has logical economic benefits as well as the ease of use. In this project, while dealing with money, allocation of cash for each user is done by this BPaaSutility.

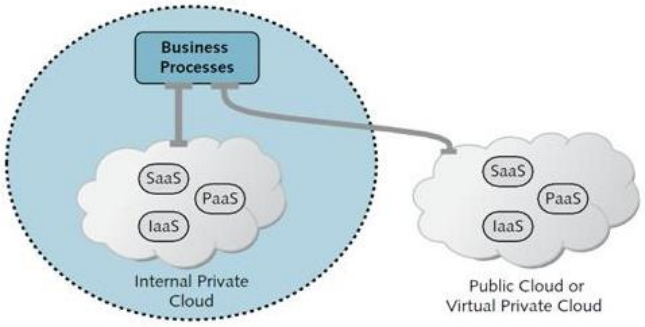


Fig. 1. An abstract BPaaS example

Figure 1 shows an abstract example that demonstrates the structure and variety of services and resources that a BPaaS can utilize. In this example, the BPaaS is composed of heterogeneous component services from the service provider and third parties. Two SaaS services (i.e., SaaS 1 and SaaS 2) used by the BPaaS are hosted and managed by the same provider. Private internal software exclusive to the BPaaS is also required. Two of the SaaS services are from external sources - SaaS 3 is from a third party, while SaaS 4 is another service of the BPaaS provider, but hosted on an external PaaS.

However, when a BPaaS has a large number of configurable components, the verification that the behavior is correct and/or meets client requirements can be challenging, as state space explosion hinders the verification of large models [17][18], and the requirements provided by clients can be complex [19]. Existing approaches in managing business process configuration ensure *domain constraints* over configuration choices while allowing basic client requirements such as selected features or control flow variations. One area that has yet to receive research attention is ensuring both *domain constraints* and client *transactional requirements* during BPaaS configuration. These requirements can include conditions for accept-able process commit or abortion, required recovery operations for key activities, or valid forms of process compensation, and are difficult to verify in a cloud-based scenario where multiple stakeholders are involved. A configuration method that ensures complex requirements within a feasible runtime will be able to provide service clients with increased trust for outsourcing potentially sensitive business operations.

To address these problems, we propose a three-step configuration and verification process which relies on a modeling paradigm. Such a paradigm allows us to capture transactional requirements and subsequently verify them. Our approach is expressive and relatively easy to use by stakeholders, while at the same time being sufficiently rigorous to allow us to apply formal methods for verification.

## II. RELATED WORK

In recent years, numerous works are focused. Scott Bourne, Claudia Szabo, Quan Z. Sheng introduced a new concept called BPaas(Business process as a service) as the next level of Saas cloud service. BPaas is an emerging cloud service that provides business service to clients over the internet. As the services provided by companies on the internet is rapidly measuring, there is a need for a dynamic business environment[1].

RajuBraskar, AnjanaJayanthdeenEt. Al provides an insight of E-Commerce offers the managing an accounting industry incredible opportunity, yet additionally makes an arrangement of new dangers and weakness, for example, security dangers. Data security, in this way, is a fundamental administration and specialized prerequisite for any proficient and successful Payment exchange exercises over the web. In any case, its definition is a perplexing undertaking because of the steady mechanical and business change and requires an organized match of calculation and specialized arrangements. Web-based business isn't fitting to all business exchanges and, inside a web-based business, there is nobody innovation that can or ought to be proper to all necessities. Web-based business isn't another marvel; electronic markets, electronic information exchange, and client internet business. The utilization of electronic information trades as a general and non-restrictive method for working together. Through the electronic exchange, the security is the most critical wonders to improve the keeping money exchange security by means of installmentexchange[2].

Robert Woitsch, Knut Hinkelmann, Ana Maria Juan Ferrer. Al proposed a new approach and that explains Business Processes as a Services (BPaaS) that is worked out in the H2020 venture CloudSocket. Idea models and semantics are utilized to adjust space particular business forms with executable work processes that are sent and underway in a multi-cloud condition. The Business Process Management System Paradigm (BPMS) is asking for the useful capacities of the purported BPaaS Environments (I) plan, (ii) designation, (iii) execution and (iv) assessment, which in fact make the CloudSocket Broker stage. This paper presents the first discoveries of adjusting clients' business needs with BPaaS cloud contributions utilizing a model-based approach[3].

XiweiXu, Ingo Weber, Mark Staples, Liming Zhu, Jan Bosch, Len Bass, CesarePautasso, Paul Rimba introduced Blockchain is a rising innovation for decentralized and transactional data sharing across a large network of a participant who does not need to trust each other. Blockchain, as a product connector with a complex inner structure, has different setups and diverse variations. Utilizing blockchain in various situations requires the examination of blockchain alternatives and items with various usage and arrangements. The scientific categorization can be utilized when looking at block chains and aid the outline and assessment of programming structures utilizing blockchain innovation. The scientific classification catches the major engineering attributes of blockchains and the effect of various choice choices. This scientific categorization is proposed to help with essential compositional contemplations about the execution and quality characteristics of blockchain based frameworks. Other than scientific classification, designs are additionally an instrument to order and compose the current arrangements [4].

ZHANG Yifeiproposes the countermeasures of featuring on the online payment security at the current circumstance and calls attention to a few parts of key developments in the development of China's protected payment framework. Bank card payment is the most as often as a possible utilized mode to shop online in the nation as of now. With the steady advancement of web-based business, outsider payment stage has likewise been quickly created. Taobao, eBay, Dangdang have quickly involved the residential market and host turn into the most favored third-get-together payment stage of the Internet clients and dealers[5].

## III. EXISTING METHODOLOGY

Existing approaches in managing business process configuration ensure domain constraints over configuration choices while allowing basic client requirements such as selected features or control flow variations. One area that has yet to receive research attention is ensuring both domain constraints and client transactional requirements during BPaaS configuration. These requirements can include conditions for acceptable process commit or abortion, required recovery operations for key activities, or valid forms of process compensation, and are difficult to verify in a cloud-based scenario where multiple stakeholders are involved. A configuration method that ensures complex requirements within a feasible runtime will be able to provide service clients with increased

trust for outsourcing potentially sensitive business operations.

The problem has concerns to consider from several perspectives, such as the different types of variable features, constraints between configuration options, and satisfying the requirements provided by the client.

*1) Business Process Management and Modeling*

Business Process Management (BPM) aims to (i) identify internal business processes of an organization, (ii) design new process models, and (iii) be able to manage and optimize business process execution by monitoring and reengineering. BPM lifecycle is an iterative process in which all the BPM aspects are covered. It consists of the following stages :

Design. Business process design consists of identifying existing processes and designing new process models using BPEL27 or BPMN28. The main objective of this step is to ensure that correct and efficient theoretical designs are prepared.

Modeling and Implementation. Processes previously designed are now modeled, then implemented in an executable process language.

Enactment. At this stage, the business processes are deployed and monitored using a Business Process Management System (BPMS).

Evaluation. The business process evaluation encompasses both business process optimization and reengineering.

## IV. PROPOSAL METHODOLOGY

**We propose a three-step configuration and verification process BPaaS modeling approach** which relies on a modeling paradigm. Such a paradigm allows us to capture transactional requirements and subsequently verify them. Our approach is expressive and relatively easy to use by stakeholders, while at the same time being sufficiently rigorous to allow us to apply formal methods for verification. We propose a BPaaS configuration process that applies formal methods to ensure that

the configuration is valid with respect to provider domain constraints, and

the process satisfies transactional requirements drawn from the business rules of the client. First, we provide an overview of the process which guides clients through BPaaS configuration, and then we provide details on how the Binary Decision Diagram (BDD) analysis and model checking are used at certain steps.

***a) Modeling Domain Constraints***

Domain constraints are rules that allow providers to restrict BPaaS configuration to valid choices. For example, several credit card transaction management resources may be available for a given payment operation, but at least one must be selected in any configuration. Our BPaaS modeling approach adapts *feature models* from the software product line engineering domain [20]. Feature models are tree-like structures being able to express domain constraints formally and visually. Typically, feature models are used to express variability in a configurable system, by modeling the constraints between optional *features*. In our approach, we adapt them to formalize constraints between the configurable activities, resources, and data objects of a BPaaS. Furthermore, by using one feature model to define all BPaaS domain constraints, we are able to define constraints that cross these configuration perspectives. For example, certain configurable activities may require the selection of specific resources.

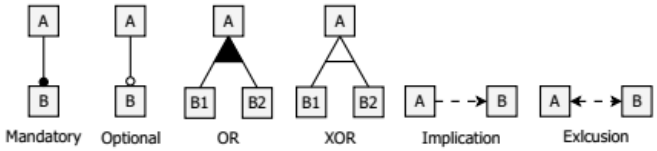


Fig. 2. Feature model constraints used in our approach

We apply six feature model relationship structures, shown in Figure 2, to model domain constraints.

The first four relationships apply to one or more leaf features if the head feature is selected. For example, the *Mandatory* and *Optional* structures define that if feature A is selected, then feature B is essential or optional respectively. *Implication* and *Exclusion* can be defined between any two features in the model, regardless of their level in the tree structure.

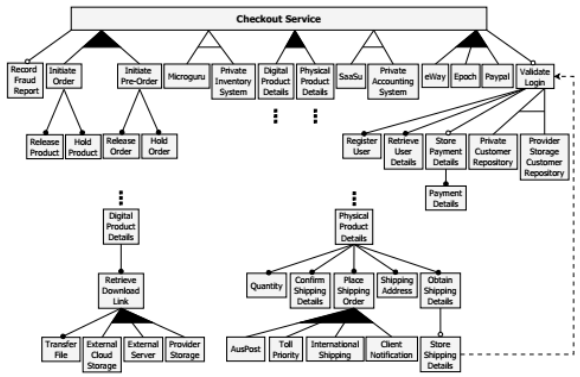


Fig. 3. Feature model of the domain constraints of our example BPaaS

A feature model capturing the domain constraints for the configuration of the checkout BPaaS is shown in Figure 3. The root Checkout Service feature contains all other features as children and allows constraints to cross between activity, resource, and data object perspectives. For example, Validate Login is an optional feature, but it requires Register User, Retrieve User Details, either Private Customer Repository or Provider Storage, and enables Store Payment Details to be selected. A selection of features that satisfy all constraints in this model, therefore conforming to all configuration requirements of the provider, is a *valid configuration* of the BPaaS.

### *b) Modeling Activities and Control Flow*

We use BPMN for modeling activities and control flow, as it formalizes the BPaaS structure while remaining easily readable for clients (see Figure 4 for checkout BPaaS). Furthermore, BPMN is a widely used notation for formalizing and executing business processes, which increases the potential client and provider base of our approach.

The configurable resources include Microguru9 for inventory management, SaaSu10 for accounting, and FTP or FTPS for digital product file transfer. Cloud storage is offered by the provider for a customer repository and digital product hosting. Data object configurability includes physical and digital product details, enabling product quantities, and payment and shipping details.

### *c) Modeling Transactional Requirements*

While BPMN provides lifecycle statecharts that represent the transactional state of individual activities, a view of the transactional state of the entire process is necessary for verification against process-level transactional requirements. Such requirements include specifying the activities critical for successful execution, necessary activities to execute prior to aborting, or requirements for valid process compensation. We adopt the separated behavior model of our previous work in transactional Web service compositions [19] and use a transactional behavior model to represent the global transactional state of the process.

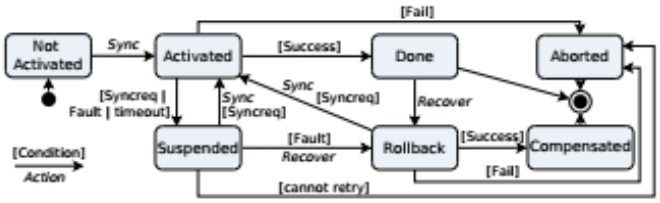


Fig. 5. Transactional behavior model of the BPaaS

The transactional behavior model of the BPaaS is represented using a statechart, as shown in Figure 5. This model contains various transactional states the BPaaS can be in at a given point, from prior to activation by a client (Not Activated), to termination through the Done, Aborted, or Compensated states. A Compensated state occurs after the effect of the operation is undone through a successful Rollback.

**Cloud** service providers can indicate changes in the transactional state of the BPaaS by modeling *inter-behavior messages* [19] between transactional behavior states and BPMN activities. These messages are exchanged between the transactional and BPMN activity models and are used for their coordination and to facilitate transactional behavior verification:

*Sync* to trigger an activity execution,
*Recover* to initiate activities for fault recovery,
*Delay* indicating that an unacceptable delay has occurred during execution,
*Ping* to test the liveness of an activity,
*Success* to commit a successful execution of the BPaaS,
*Fail* to abort execution,
*Syncreq* to request a *Sync* message for a retrial following a fault or recovery,
*Fault* to indicate the presence of a fault that requires recovery, and
*Ack* to acknowledge a received *Ping* message.

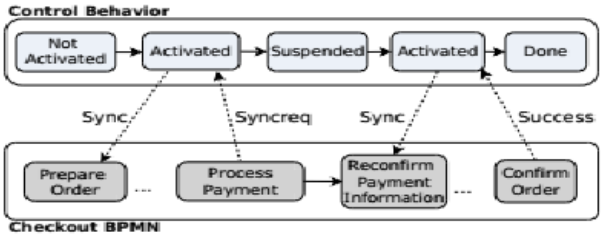


Fig. 6. Inter-behavior messages enabling communication between BPMN and transactional behavior models

Figure 6 shows an example of how the control behavior model can direct and communicate with the checkout BPMN using inter-behavior messages. In this conversation session, the process becomes sus-pended after processing payment fails, but the process is able to commit successfully after the user is asked to reconfirm their payment information.

## V. Result Analysis

There are several reasons that our configuration process must be able to handle large and complex scenarios in an efficient way. Firstly, the impact state space explosion has on model checking performance is exponential as the size of the model increases. Clients may also have large and complex sets of transactional requirements to be verified. Furthermore, model checking may need to be applied by the client several times, if a configuration solution is difficult to obtain. Therefore, long verification times are likely to compound and become a bigger problem for clients

BDD Analysis The first verification step in our approach identifies all features required for the client transactional require ments and determines whether they can all be selected while satisfying the feature model constraints. This implies that at least one valid configuration must exist using the activities, resources, and data objects specified in the requirements set, or extra features required by the client. We employ BDD-based analysis, which has been proven as an effective method for determining feature model satisfiability [21].

A BDD is an acyclic graph visualization of a propositional logic formula. Variables of the formula are represented as nodes with two outgoing branches, indicating their true or false assignment. The graph is constructed in such a way that each complete path from the head node to the terminal node represents the assignment of boolean variables. All paths terminate at a final true or false node, which determines whether the variable assignments of that path satisfy the propositional logic formula.

There are several reasons that our configuration process must be able to handle large and complex scenarios in an efficient way. Firstly, the impact state space explosion has on model checking performance is exponential as the size of the model increases. Clients may also have large and complex sets of transactional requirements to be verified.

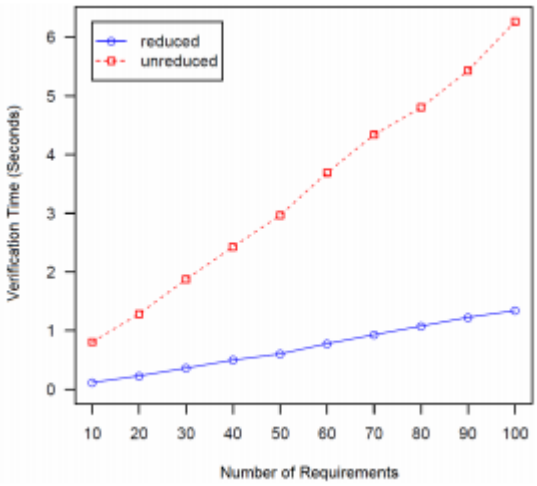


Fig. 7. Verification times during configuration with and without reduction for 10 to 100 requirements

Furthermore, model checking may need to be applied by the client several times, if a configuration solution is difficult to obtain. Therefore, long verification times are likely to compound and become a bigger problem for clients.

- To the best of our knowledge, transactional requirements important to clients, such as those supported by our template set, are not yet supported by any business process configuration method, and this is one of the major contributions of this work compared to existing works.
- This increases client trust that the service will behave in a manner consistent with internal business policies and requirements, without having to perform their own analysis of the service behavior.
- Our BPaaS model enables configuration from numerous perspectives important to BPaaS clients, namely, activities, resources, and data objects.
- Our configuration method aims to elicit and ensure complex transactional requirements

from clients, by adapting the temporal logic template set.

- It has the advantage of a reduced runtime when configuring services with many configuration options and values.

## VI. CONCLUSION AND FUTURE ENHANCEMENT

The increase in cloud computing adaptations in recent years has produced the concept of Business Process as a Service (BPaaS), whereby service providers are able to offer common or proven business processes to clients looking to automate and/or outsource parts of their operations. We address the problem of managing BPaaS configuration in a way to ensure that the resulting service i) is valid with respect to configuration constraints of the provider, and ii) satisfies transactional requirements drawn from the business rules of the client. Our approach utilizes several modeling techniques, including BPMN for business process structure, statecharts for the transactional state, feature models for configuration constraints. Using these models, we develop a BPaaS configuration process that applies a Binary Decision Diagram (BDD) analysis and model checking. BDD analysis ensures that BPaaS features selected during configuration do not violate the domain constraints of the service provider, while model checking verifies the configured BPaaS against transactional requirements provided by the client. To reduce the impact of the state-space explosion, we employ a state-space reduction algorithm and split the model checking into two phases. These phases verify differ-ent configuration perspectives separately and allow for the state space and temporal logic properties to be reduced further. Our performance analysis shows that the proposed configuration method is capable of verifying models with hundreds of activities, resources, data objects, and requirement sets within seconds.

## *VII. REFERENCES*


*[1] S. Bouchenak, G. Chockler, H. Chockler, G. Gheorghe, N. Santos, and A. Shraer, "Verifying cloud services: present and future," SIGOPS Operating Systems Review, vol. 47, no. 2, 2013.*

*[2] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghal-sasi, "Cloud Computing The Business Perspective," Decision Support Systems, vol. 51, no. 1, pp. 176–189, 2011.*

*[3] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud Computing: State-of-the-Art and Research Challenges," Journal of Internet Services and Applications, vol. 1, no. 1, pp. 7–18, 2010.*

*[4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A View of Cloud Computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.*

*[5] I. Kumara, J. Han, A. Colman, and M. Kapuruge, "Runtime Evolution of Service-Based Multi-Tenant SaaS Applications," in Service-Oriented Computing. Springer, 2013, pp. 192–206.*

*[6] Y. Liu, B. Zhang, G. Liu, M. Zhang, and J. Na, "Evolving SaaS Based on Reflective Petri Nets," in Proceedings of the 7th Workshop on Reflection, AOP and Meta-Data for Software Evolution. ACM, 2010, pp. 7:1–7:4. [Online]. Available: http://doi.acm.org/10.1145/1890683.1890690*

*[7] J. Schroeter, S. Cech, S. Gotz,¨ C. Wilke, and U. Aßmann, "Towards Modeling a Variable Architecture for Multi-Tenant SaaS-Applications," in Proceedings of the 6th International Workshop on Variability Modeling of Software-Intensive Systems. ACM, 2012, pp. 111–120.*

*[8] R. Accorsi, "Business Process as a Service: Chances for Remote Auditing," in The 35th Annual Computer Software and Applications Conference Workshop. IEEE, 2011, pp. 398–403.*

*[9] S. Bourne, C. Szabo, and Q. Sheng, "Managing Configurable Business Process as a Service to Satisfy Client Transactional Requirements," in Proceedings of the 11th International Conference on Services Computing. IEEE, 2015, pp. 154–161.*

*[10] T. Lynn, J. Mooney, M. Helfert, D. Corcoran, G. Hunt, L. Van Der Werff, J. Morrison, and P. Healy, "Towards a Framework for Defining and Categorising Business Process-As-A-Service (BPaaS)," in 21st International Product Development Management Conference, 2014.*

*[11] M. P. Papazoglou and W. van den Heuvel, "Blueprinting the Cloud," IEEE Internet Computing, vol. 15, no. 6, pp. 74–79, 2011.*

*[12] J. Mendling, J. Recker, M. Rosemann, and W. M. P. van der Aalst, "Generating Correct EPCs from Configured C-EPCs," in Proceedings of the Symposium on Applied Computing. ACM, 2006, pp. 1505–1510.*

*[13] W. M. P. van der Aalst, M. Dumas, F. Gottschalk, A. H. M. ter Hofstede, M. La Rosa, and J. Mendling, "Preserving Correct-ness During Business Process Model Configuration," Formal Aspects of Computing, vol. 22, no. 3-4, pp. 459–482, 2010.*

*[14] G. Groner,¨ M. Boskovi˘c,´ F. S. Parreiras, and D. Gasevi˘c,´ "Modeling and Validation of Business Process Families," Information Systems, vol. 38, no. 5, pp. 709–726, 2013.*

*[15] A. Kumar and W. Yao, "Design and Management of Flexible Process Variants using Templates and Rules," Computers in Industry, vol. 63, no. 2, pp. 112–130, 2012.*

*[16] M. La Rosa, M. Dumas, A. H. M. ter Hofstede, J. Mendling, and F. Gottschalk, "Beyond Control-Flow: Extending Business Process Configuration to Roles and Objects," in Proceedings of the 27th International Conference on Conceptual Modeling, 2008, pp. 199–215.*

*[17] W. M. P. van der Aalst, N. Lohmann, M. Rosa, and J. Xu, "Correctness Ensuring Process Configuration: An Approach Based on Partner Synthesis," in Proceedings of the 8th International Conference on Business Process Management (BPM 2010), ser. Lecture Notes in Computer Science, vol. 6336, pp. 95–111.*

*[18] C. Baier and J.-P. Katoen, Principles of Model Checking. MIT Press, 2008.*

*[19] S. Bourne, C. Szabo, and Q. Sheng, "Verifying Transactional Requirements of Web Service Compositions using Temporal Logic Templates," in Proceedings of the 14th International Conference on Web Information System Engineering. Springer, 2013.*

*[20] K. C. Kang, J. Lee, and P. Donohoe, "Feature-Oriented Product Line Engineering," IEEE Software, vol. 19, no. 4, 2002.*

*[21] D. Benavides, S. Segura, and A. Ruiz-Cortes,´ "Automated Analysis of Feature Models 20 Years Later: A Literature Review," Information Systems, vol. 35, no. 6, pp. 615–636, 2010.*

*[22] W. Zhang, H. Yan, H. Zhao, and Z. Jin, "A BDD-Based Approach to Verifying Clone-Enabled Feature Models Constraints and Customization," in High Confidence Software Reuse in Large Systems. Springer, 2008, pp. 186–199.*

*[23] E. Clarke, "Model Checking," in Foundations of Software Technology and Theoretical Computer Science. Springer, 1997.*

*[24] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella, "NuSMV 2: An Opensource Tool for Symbolic Model Checking," in Computer Aided Verification. Springer, 2002, pp. 241–268.*

*[25] S. Kripke, "Semantical Considerations on Modal Logic," Acta Philosophica Fennica, vol. 16, no. 1963, pp. 83–94, 1963.*

*[26] R. Mietzner, A. Metzger, F. Leymann, and K. Pohl, "Variability Modeling to Support Customization and Deployment of Multi-Tenant-Aware Software as a Service Applications," inProceedings of the ICSE Workshop on Principles of Engineering Service-Oriented Systems. IEEE, 2009, pp. 18–25.*

*[27] J. Schroeter, P. Mucha, M. Muth, K. Jugel, and M. Lochau, "Dynamic Configuration Management of Cloud-based Applications," in Proceedings of the 16th International Software Product Line Conference (SPLC 2012), pp. 171–178.*

*[28] B. F. van Dongen, M. H. Jansen-Vullers, H. M. W. Verbeek, and W. M. P. van der Aalst, "Verification of the SAP Reference Models using EPC Reduction, State-Space Analysis, and Invariants," Computers in Industry, vol. 58, no. 6, 2007.*

*[29] M. La Rosa, W. M. P. van der Aalst, M. Dumas, and A. H. M. ter Hofstede, "Questionnaire-Based Variability Modeling for System Configuration," Software & Systems Modeling, vol. 8, no. 2, pp. 251–274, 2009.*

*[30] L. Cui, H. Wang, J. Lin, and H. Pu, "Customization Modeling Based on Metagraph for Multi-Tenant Applications," in The 5th International Conference on Pervasive Computing and Applications. IEEE, 2010, pp. 255–260.*

*[31] A. Banerjee, "A Formal Model for Multi-Tenant Software-as-a-Service in Cloud Computing," in Proceedings of the 5th ACM COMPUTE Conference: Intelligent & Scalable System Technologies. ACM, 2012, p. 18.*

*[32] W. M. P. Van Der Aalst, M. Dumas, F. Gottschalk, A. H. M.ter Hofstede, M. La Rosa, and J. Mendling, "Correctness-Preserving Configuration of Business Process Models," in Fundamental Approaches to Software Engineering. Springer, 2008.*

*[33] M. La Rosa, M. Dumas, A. H. M. ter Hofstede, and J. Mendling, "Configurable Multi-Perspective Business Process Models," Information Systems, vol. 36, no. 2, pp. 313–340, 2011.*

*[34] M. X. Wang, K. Y. Bandara, and C. Pahl, "Process as a Service Distributed Multi-Tenant Policy-Based Process Runtime Governance," in The International Conference on Services Computing. IEEE, 2010, pp. 578–585.*

*[35] W. M. P. van der Aalst, "Business Process Configuration in the Cloud: How to Support and Analyze Multi-tenant Processes?" in Proceedings of the 9th European Conference on Web Services, 2011, pp. 3–10.*

*[36] W. Tsai and X. Sun, "SaaS Multi-Tenant Application Customization," in The 7th International Symposium on Service-Oriented System Engineering. IEEE, 2013, pp. 1–12.*

*[37] J. M. Jiang, S. Zhang, P. Gong, and Z. Hong, "Configuring Business Process Models," SIGSOFT Software Engineering Notes, vol. 38, no. 4, pp. 1–10, 2013.*

*[38] A. Hallerbach, T. Bauer, and M. Reichert, "Guaranteeing Soundness of Configurable Process Variants in Provop," in IEEE Conference on Commerce and Enterprise Computing. IEEE, 2009, pp. 98–105.*

*[39] F. Gottschalk, W. M. P. van der Aalst, and M. H. Jansen-Vullers, "Configurable Process Models A Foundational Approach," in Reference Modeling. Springer, 2007, pp. 59–77.*

*Mr.**MuthukuriChiranjeevi** pursuing MCA 3rd year in Qis College and Engineering and Technology in Department of Master of Computer Applications, Ongole*

*Mr. **Katti Jaya Krishna** is currently working as an Assistant Professor in Department of Master of Computer Applications in QIS College of Engineering & Technology.*