Interactive Preference-based Approach to Optimal Feature Configuration of Multi-dimensional Service Platforms

Research Proposal · December 2017

DOI: 10.13140/RG.2.2.18809.36966

Azubuike Ezenwoke
Covenant University Ota Ogun State, Nigeria

Interactive Preference-based Approach to Optimal Feature Configuration of Multi-dimensional Service Platforms

Azubuike Ezenwoke
Department of Computer and Information Sciences,
Covenant University
KM10 Idiroko Road, Ota, Nigeria
+234-803-062-6181
azu.ezenwoke@covenantuniversity.edu.ng

ABSTRACT
A multi-dimensional service delivery platform (MDSP) supports development, deployment and management of services in multiple business domains, serves multiple consumers with different functional and non-functional requirements and integrates services from diverse external collaborators to actualize the platform's business objective. Consumers of product line services have variant needs that are based on the specific requirements of their business objectives, which demands optimal configuration of the MDSP. Optimal configuration of the MDSP connotes the existence of the most appropriate set of features on the MDSP that best approximates the consumer's requirements, in the face of multiple conflicting objectives. So far, solutions proposed in the literature have mainly used either a priori or a-posterior methods. In prior methods, the requirements and preference information is provided before the configuration process begins; while a set of possible configurations is first generated and preferred selection is made from the set in a-posterior methods. These methods lack the kind of flexibility afforded by interactive methods in an attempt to generate satisfactory results. The aim of this research is to develop an approach that engenders the derivation of optimal configurations from a multi-dimensional service platform (MDSP), in a manner that is interactive and meets the needs of the consumer.

Categories and Subject Descriptors
D.2.13 [**Software Engineering**]: Reusable Software- Domain engineering, Reuse models

General Terms
Algorithm, Performance, Design.

Keywords
Feature Modeling, Variability Modeling, Software Product Line, Optimization, multi-objective optimization, Automated Analysis, Cloud computing, Platform as a service, Interactive Configuration, service delivery platform

1. INTRODUCTION
Service delivery platforms comprise a set of software and hardware components (databases, servers, network resources, integrated development environment) that provide mechanisms for development, deployment and management of end-user services. Typical service delivery platforms include Google App engine[1] and force.com[2]. A multi-dimensional service delivery platform (MDSP), supports development, deployment and management of services in multiple business domains (which includes business, health, education), serves multiple consumers with different functional and non-functional requirements. It also integrates many services from diverse external collaborators to actualize the platform's business objective e.g. the platform described in the Indenica project[3]. A possible and useful adaptation of MDSP is providing support for product line development, where consumers that share common requirements but yet having variant specific needs can leverage optimal configuration of the MDSP to serve their specific needs. In such a scenario a MDSP that serves a product line context would be capable of presenting variants views to consumers based on the specific requirements of their business objectives. Also, consumers can configure the features of the MDSP, to derive an instance to perform consumer-specific business tasks [1].

Hence, the configuration of a MDSP that is adapted for the product line context can be abstracted in terms of feature models. Feature models [2,3] are the most widely used representations for variabilities and commonalities in software product lines. Capturing the variability information of MDSP, in a single feature model could be very complex [4]. The features (and their attributes) of a MDSP could be very large, and an attempt to manually derive meaningful compositions from these models is time-consuming and error-prone [5,6,7].

This challenge necessitates automated approaches to extract useful information from feature models, during product configuration [3,8,9,10]. Several studies on automated analysis of feature models have been conducted and reported in the literature [3,8]. The search for valid configurations from the feature models ranges from just any configuration with no particular preference to one or more optimal configurations that satisfy specific objective functions based on specific criteria [11]. How to optimize the inclusion of features in a configuration, in the face of multiple constraints in a context such as MDSP is still an open problem in the literature [12]. Searching for an optimal feature set

[1] https://developers.google.com/appengine/
[2] http://www.force.com/
[3] http://www.indenica.eu/

should involve the use of some preference constraints to optimize search results. The inputs to such optimization operation are the feature models and objective function(s), while the output is a product configuration that satisfies the criteria defined by the function(s) [3]. The optimization operation, most suitable on extended feature models [9], is such that a set of features can be selected by maximizing or minimizing the values of given feature attributes.

2. BACKGROUND AND MOTIVATING EXAMPLE

For example, consider a database feature from the feature model of an eShop product line (see Figure 1). A decision to select a database facility could be evaluated in terms of its cost, performance, security and memory, and the cost is implicitly a function of performance, security and memory. As can be seen from figure 1, the cheapest database facility to manage the data of an eShop instance is Shared-tables, which costs 20; compared to Shared-database (cost=30) or Isolated Database (cost=50). Therefore, a consumer that desires to maximize Performance, Memory, and Security while at the same time minimize Cost in selecting a database facility for its eShop services initiates a multi-objective optimization problem (MOP) scenario. Multi-objective optimization scenario occurs when there are two or more conflicting objective functions that must be simultaneously optimized in the face of a given set of constraints [13].

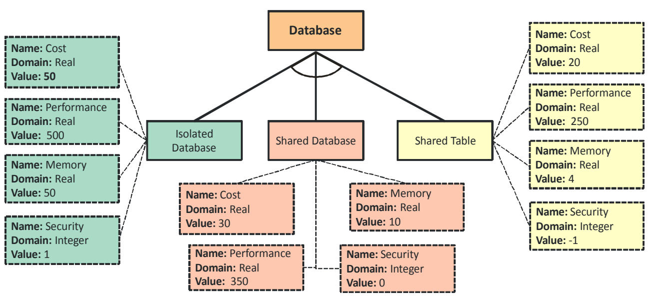

Figure 1: A view of the Database Feature from the Feature Model for an eShop productline

A MOP problem has to be solved in order to derive the optimal configuration of MDSP that best approximates consumers' requirements. Unfortunately, there is no 'best' solution in solving MOP, but rather a list of 'equally good' or optimal trade-off solutions, called the Pareto optimal set from which an optimal solution, called the Pareto-optimal, is selected. Even though it is important to allow the consumer to specify preference information, in order to determine the most desirable feature combinations; arriving at a single solution from the Pareto optimal set is a non-trivial process. Therefore the how and when preference information is incorporated into the search process, has significant impact on the time it takes to arrive at that optimal solution. Hence, the methods used to solve MOP in the context of MDSP should satisfy the following criteria: 1) generate Pareto optimal set of feature combinations in a reliable manner; 2) provide the consumer with an overview of the available Pareto optimal configurations; 3) arrive at the most preferred configuration in a reasonable time; 4) ensure uncomplicated mode of exchanging information between the consumer and the platform; and 5) supports the consumers to convincingly determine the most preferred configuration solution as the final one [14].

The three main methods of incorporating preference information for solving multi-objective optimization problem are: a priori, interactive and a-posterior methods [14]. In a priori methods, the multiple objectives are combined into a single objective function through a process called scalarization. Also, the consumer first provides the requirements and preference information and the platform's configurator, in a search process, attempts to find a combination of feature that approximates, as much as possible, the requirements and preference information. The drawback of this approach is that the consumer may cut off the possibilities of arriving at 'more satisfactory' feature combinations, due to the constraints imposed by the consumers' preference, defined a priori. Therefore, the opportunity cost of missing out on a more approximate feature configuration that satisfies consumer's requirements and preference is very high.

In a-posterior methods, a set of Pareto optimal configurations is first generated and the consumer is expected to select the most preferred. In spite of the consumer being exposed to an overview of available Pareto optimal feature combinations, the search process of a-posterior methods could be increasingly complex and computationally expensive. Furthermore, if the search process is terminated too early the feature combinations presented may not be the most optimal.

With interactive methods, the interaction between the consumer and the platform's configurator increases the possibilities of arriving at more desirable feature configurations and generates better search results. As shown in figure 2, the decision making process of the consumer, and optimization activities of the platform's configurator, are interlaced together; such that the preferences specified by the consumer at a given instance is what determines the optimal configuration generated by the platform. The partial results of the search are revised again by the consumers, and the search process progresses. This process continues until the final solution is reached.

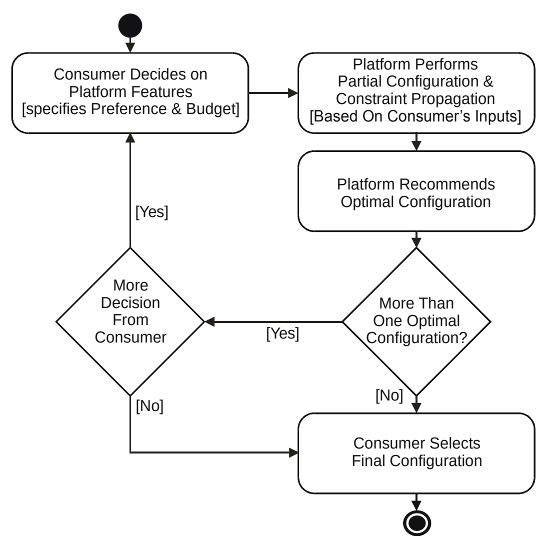

Figure 2: Interactive Configuration Process

Furthermore, it is possible that the results of a multi-objective optimization operation may not be desirable at the first iteration. For example, a consumer of the database service (Figure 1), as part of a larger composite eShop service, may desire to increase his budget by a certain amount in other to get additional value for database performance and security. Therefore, the iterative refinement of criteria by re-specifying preference information should be allowed until desired feature combinations are obtained.

Similarly, a consumer may desire to relax or tighten its preferences during the configuration search process based on current partial search results. In making such decisions during configuration, the configurator should automatically propagate the effect of such decisions to ensure their consistency with previously made features selections. So far, configuration solutions proposed in the literature, to derive optimal configurations, have mainly used either a priori or a-posterior methods, which lack the kind of flexibility afforded by interactive methods. Hence, a vital issue that will be addressed in this research is: How to derive optimal configurations by allowing iterative refinement of preference information during the platform configuration process in a manner that accurately approximate consumer satisfaction at minimal computational cost.

The envisioned multi-dimensional service platform (MDSP) for the introduction of interactive preference-based optimization for feature configuration is the product line as a service (PLaaS) context (See Figure 3). Akin to the PaaS model of Cloud Computing [15,16], PLaaS is conceptualized as a web-based on-demand integrated development environment for the design, development, deployment and management of a product line of service-based applications. A PLaaS offering would comprise databases, middleware, product line development tools, run-time and execution environments [17], and would host and serve multiple tenants per time, each having a variant view of the platform. The overriding objective of pursuing a PLaaS initiative is to create a viable platform for very small software enterprises (VSSE) to readily access core software product line services – such as variability modeling, and configuration solution for web application- without incurring the huge cost associated with adopting an independent SPL initiative.

The use of an interactive preference-based optimization approach for feature configuration in the PLaaS context, as proposed in this research, will facilitate the generation high-quality PLaaS configurations that best meet consumers (i.e. VSSE) requirements and preferences in an effective manner.

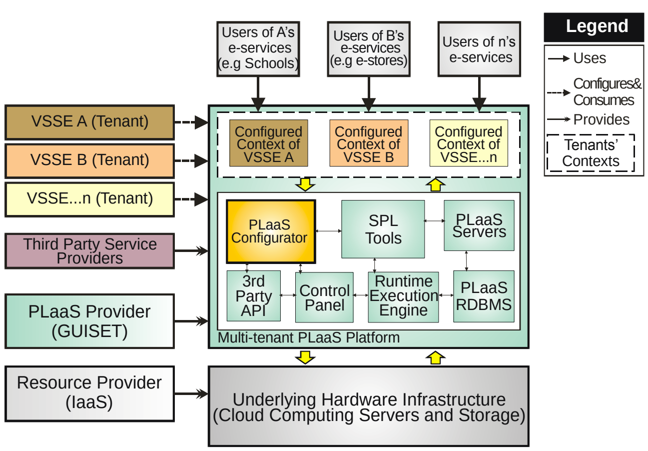

Figure 3: A view of PLaaS platform and stakeholders

The core motivation for PLaaS is the need to serve under-resourced VSSE with meager budget, but desire to adopt software product line initiatives in their software development projects. Cloud computing is potentially the most convenient way for these VSSE to adopt advanced technology [15]. PLaaS, as a cloud computing model, would enable the adoption of SPL practices by VSSE, who lack SPL proficiency and cannot afford the cost of adopting mainstream SPL. PLaaS provides a platform for VSSE to leverage, on-demand, product line technologies and services without incurring the total cost of ownership.

PLaaS can be achieved via a multi-tenancy arrangement [17]. Multi-tenancy would enable the provision of dynamically customizable development environments to satisfy multiple and diverse tenants' requirements in a cost effective manner. In the literature, the use of variability management techniques from the SPL domain has been proposed for the realization of multi-tenancy [18,19] and hence, SPL engineering, as an engineering approach, is suitable to engineer a multi-tenant PLaaS platform. In a PLaaS platform, variability could be expressed in various dimensions [4,20], by separating the feature set, such as the hardware features, external software features, internal software features, security requirements [4], and software context [21].

3. RESEARCH ISSUES, OBJECTIVES, AND QUESTIONS

Multi-tenant cloud-based environment, such as the context of provisioning PLaaS, requires computationally efficient product configurator (in terms of CPU and memory consumption) for automated configuration of multi-dimensional feature models [19]. Importantly, the platform should comprise a self-optimizing mechanism that balances the goals of maximizing the business value derived from consuming PLaaS and the optimal utilization of cloud infrastructure and resources. From the PLaaS provider's viewpoint, an efficient way to multiplex tenants' activities and demands to reduce operational cost is desirable [23].

The process of selecting the most optimal set of features that satisfies multiple optimization goals during PLaaS configuration in order to consume PLaaS is the focus of this research. Being an instance of a multi-criteria decision making problem (MCDM) [27,28], the consumer (also the tenant and/or VSSE) becomes the decision maker (DM) [29], from whom preference information is elicited to determine the best available combination of features that approximates the tenant's requirements (functional and non-functional). The PLaaS platform configurator should comprise multi-objective optimization mechanisms that incorporates the consumer's preference information in the decision making process. This would be done in such a manner that optimizes the utilization of the cloud infrastructure, and resources in satisfying the consumer requirements and preferences. This means that techniques for configuring the PLaaS variability should support preference-based multi-objective optimization and should derive optimal (valid and satisfactory) configurations in reasonable time; valid and satisfactory configuration in the sense that, the configuration is correct, complete, and most approximates the tenant's requirements and preferences.

An open problem in the literature is the need to determine which features to include in a particular configuration in the face of certain resource constraints, imposed by multiple and conflicting optimization goals [12]. Existing optimal configuration solutions in the literature have used either a priori or a-posterior approaches, which do not allow iterative refinement of consumer preferences in a flexible way that engenders the generation of optimal configurations. Hence, there is need for an efficient approach that will facilitate the derivation of optimal feature set that accurately approximates consumers' requirements and preferences at minimal operational cost and in a flexible way. Proposed in this research is an automated configuration approach that optimizes multiple conflicting objectives by incorporating interactive preference information during PLaaS (See figure 2).

To this end, this research will attempt to answer the research question below: *How can one derive optimal configurations of a multi-dimensional service platform, such as a PLaaS context, in a manner that is interactive and achieves consumer satisfaction?*

The aim of this research is to develop an approach that engenders the derivation of optimal configurations from a multi-dimensional service platform (MDSP), in a manner that is interactive and achieves consumer satisfaction. In order to achieve this aim, the following objectives were formulated:

- To understand what constitutes variability in a multi-dimensional PLaaS context, and why existing configuration techniques are not suitable for automated interactive configurations.
- To experiment with current automated configuration techniques and discover how to create an automated configuration approach that is interactive and considers consumer preferences for optimal configuration of PLaaS.
- To develop an automated configuration approach that supports interactive preference articulation for use in a PLaaS context.
- To evaluate the performance of the proposed approach with usability, user experience and accuracy metrics via a case study in an experimental context.

4. RELATED WORK

The multiple variability dimensions of a PLaaS, separated into several feature models, enable on-demand composition and analysis [4,23]. The language used to model such multi-dimensions must have robust formal semantics that facilitates automated analysis and configuration [25]. Automated analysis of feature models uses computer-aided mechanisms to extract important information from feature models [3,26]. The automated approach entails mapping the feature models into a specific formal representation as inputs and solvers are used to perform analysis operations to obtain results. A solver is a software package that accepts formal representations as inputs and determines some satisfiability criteria [3]. Formal configuration techniques used to provide automated support for analysis operations have been classified into four categories: Propositional Logic (PL), Description Logic (DL), Constraint Programming (CP), and Ad-hoc algorithms [3,8].

In the PL approach, the feature models is translated into a propositional formula (e.g. conjunctive normal form-CNF), then solvers such as satisfiability solvers (SAT solvers) are used to perform analysis operations on the formula. Other solvers used in PL approaches are Alloy[4] and SMV[5] – a symbolic model checker for verifying logic systems. DL approaches map feature models into description logic and logic reasoners such as RACER or Pellet are used for analysis. The CP approach represents feature models as a Constraint Satisfaction Problem (CSP) and CSP solvers use constraint programming to find the solution to the problem. Approaches classified under other studies used ad hoc algorithms and tools for automated analysis, e.g. a Prolog-based prototype used for analysis in FODA [2].

[4] http://alloy.mit.edu/

[5] www.cs.cmu.edu/~modelcheck

In extended feature models, variability models are annotated with quality information (non-functional requirements e.g. memory consumption, cost etc). Analysis of feature models does use qualities as basis in specifying preferred configuration requirements. Roos-Frantz et al., [30] developed an approach for quality-aware analysis in software product lines based on the orthogonal variability model (OVM). Quality-centric variability information was translated into a CSP and a prototype tool (FaMa-OVM) was used to perform verification task that meets certain quality conditions. However, the approach does not support interactive configuration and the optimization operation. Karataş et al. [9] introduced a way to map extended feature models to constraint logic programming over finite domains. This approach enabled the use of CLP (FD) solvers (a class of CSP solvers) to analyze the models with complex cross-tree inter-attribute relationships. The authors of [36] used Hierarchical Task Network (HTN), a preference-based artificial intelligence planning technique to represent feature model's functional and non-functional requirements. In the approach, feature models and stakeholder's preferences (based on non-functional qualities) were transformed into HTN planning problem, and SHOP2 was used to derive an optimal plan. Guo et al., [12] proposed the use of a genetic algorithm, called GAFES, to optimize feature selection during configuration by maximizing or minimizing an objective function. The use of genetic algorithm is classified as a posterior [14]. Also, Mendonca et al., [32] proposed SPLOT, a web-based system and configuration system developed in Java. It uses HTML-based template engine to create interactive Ajax-based reasoning, based on BDD and SAT solvers, to reason on feature models. The online usage context and support for interactive configuration of the solution proposed in this research is similar to SPLOT. However, none of these approaches consider iterative inputs during configuration in order to generate the most optimal results. Even though SPLOT is not used within a service delivery context like PLaaS, our approach would be a web-based tool that can be integrated to the PLaaS platform, where prospective VSSE could 'shop' for preferred 'slice' of the platform in an interactive manner. Attempts have been made to optimize multiple non-functional or quality attributes in deriving optimal solutions e.g. [31,33,34,35]; but, none of these proposals, except [31], considered integrating users' preference information in the configuration process and the preference information was specified a priori.

5. RESEARCH METHODOLOGY

The research approach adopted in this research is classified as Design Science research [36]. Based on the research framework proposed by Hevner et al., [36], this research would progress in two phases- Engineering by discovery of a automated configuration approach that uses interactive preference articulation method for solving MOP and, Justification of the discovered approach. Based on the Hevner's research framework, the environment (See figure 4) for this research is identified as the business needs in the technology domain that impact on infrastructure, applications, and development capabilities. The knowledge base provides the repository of applicable knowledge in terms of foundations and methodologies to give rigor to the research process. In this study, literature survey would be embarked on to identify requirements, frameworks, tools, methods, and experiment designs useful to both phases.

First, the knowledge gained from the literature would be used to design a multi-objective optimization algorithm that uses interactive preference articulation methods, as part of a configuration solution. The proposed approach would accept formal representations of multi-dimensional variability models of the PLaaS in a specific modeling language. The proposed approach would be implemented as a tool fragment with Java in Eclipse IDE, and deployed in an online web-based environment to demonstrate how the approach supports configuration of PLaaS platforms.

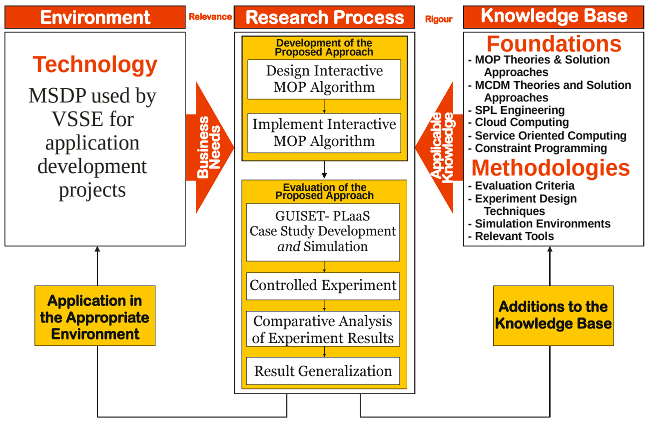

Figure 4: Doctoral Research Methodology, adapted from [36]

Secondly, the case study for evaluating the performance of the proposed approach would be carried out in the context of the GUISET [37] project. GUISET is envisioned as both an enabling infrastructure and a suite of on-demand Service-Based application. As a cloud-computing model, GUISET is aimed at offering affordable e-enabling and “appliance-like” technology services (PLaaS) through the internet to lower the total cost of ownership. The GUISET infrastructure would provide developers the required tools and environments for consuming PLaaS on a pay-as-you-go basis. These services are aimed at e-enabling the activities of under-resourced VSSE. The primary motivation for the GUISET project is economic advantages of enterprise clusters over stand-alone organization such as resource sharing, cost reduction, and ability to compete with larger firms. A GUISET PLaaS case study would be used to evaluate experimental performance of the proposed approach and corresponding tool-support.

Controlled experiments would be performed to compare the performance in terms of usability, user experience and accuracy of the proposed approach against existing configuration solutions that implement a priori and a posteriori methods for multi-objective optimization in the SPL context. This would be done to prove the efficiency and correctness of search results of the proposed approach. Precision and recall metrics would be used to measure the accuracy of the results. Comparative analysis would be carried out on the performance results obtained from the proposed approach and existing approaches. Based on the results of the comparative analysis, arguments would be used to justify the performance of the proposed approach.

The GUISET case study would be deployed on a simulated cloud computing environment and the experiments would be performed in the same environment using BeTTy [38]. BeTTy is a benchmarking and testing platform that supports the generation of test data useful for evaluating the performance of feature model analysis tools in both average and pessimistic cases. Simulation was adopted because of the limitations associated with the use of real cloud infrastructures such as Amazon EC2 or Rackspace, which are: real infrastructures would limit the experiments to the scale of the infrastructure, and results would jeopardize the reproduction of the results. Also, the existing conditions of the Internet-based environment are beyond the control of developers and finally, real infrastructure requires real financial commitments. The use of a simulator enables deployment in a cost-free, repeatable, and controlled environment [39].

6. PRELIMINARY KEY RESULTS

6.1 Overview of Solution Approach

The approach proposed in this work would be to use a preference-based multi-objective optimization algorithm that allows for several intermediate inputs, by the consumer, during the runs of the algorithm, to obtain the most preferred optimal configuration in a computationally efficient manner. The three ways to specify preference information in interactive methods are trade-off information, reference points and classification of objective functions [14]. Critical conceptual analysis studies would be carried out to determine which of these approaches would be best suitable in achieving the goals of this research.

The expected results of this research would be an approach to derive optimal configurations from a MDSP, based on the complex multi-dimensional variability model of such platforms. This would be done in a manner that integrates interactive preference articulation method optimizing multiple and conflicting objectives during platform configuration (see figure 2). The optimization approach would solve the multi-objective optimization problem (maximizing and minimizing), enabling interactive preference articulation to obtain most preferred optimal configuration. Together with a Constraint Satisfaction Problem (CSP) solution approach, this optimization solution would be implemented as a tool fragment for configuration of multi-dimensional PLaaS platform. The research objectives highlighted were formulated to achieve this aim.

Other key characteristics of the proposed approached are means to: model platforms functional and non-functional requirements and consumers’ preference, support interactive configuration based on consumers’ requirements and preferences, support optimization operation in a time efficient manner, support analysis of extended feature models based on the multi-dimensional variability model of the MDSP, provide an automated on-line web-based tool support for interactive configuration for use in the envisioned PLaaS context, and the use of interactive approaches from Multi-Criteria Decision Making Domain in the context of SPL.

6.2 Expected contribution

The expected contributions of this research work are highlighted as follows:

- Creation of a novel automated configuration approach that incorporates interactive preference articulation methods for optimizing multiple and conflicting consumer’s objectives towards the derivation of optimal configurations from multi-dimensional service platforms (MDSP).

- Demonstrate the application of multi-criteria decision making interactive approaches to solve the multi-objective optimization problem in the context of SPL.

The relationship between the key topical context of this proposal and proposed contribution is shown in Figure 5. The concept graph consists of seven nodes and edges point from nodes exerting an impact on those they influence.

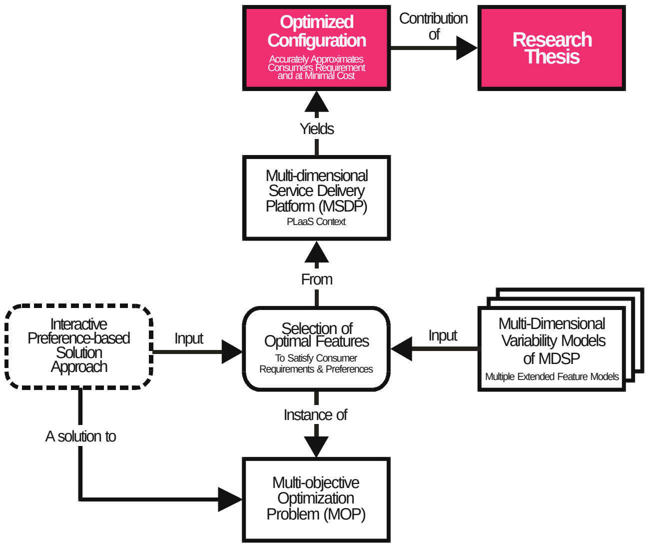

Figure 5: Relationship between the key topical context and proposed contribution

Table 1. Summary of work plan

	Activity	**Time**
1	Literature Review: Automated Analysis of Feature Models	**First and Second Quarter**
2	Literature Review: Multi-Objective Optimization	
3	Identifying the variability requirements of multi-dimensional PLaaS context	
4	Literature review on existing configuration techniques suitable for automated interactive configurations in multi-dimensional service context.	
5	Prepare paper on results of reviews.	
6	To experiment with current automated configuration techniques and discover how to create an automated configuration approach that supports interactive preference articulation.	**Third and Fourth Quarter**
7	Engineer an automated configuration approach that incorporates the interactive preference articulation concept for use in a PLaaS context.	
8	Prepare paper on solution engineering approach.	
9	Perform evaluation of the proposed engineering approach via a case study in an experimental context.	
10	Prepare paper on evaluation results of the approach.	**Fifth Quarter**
11	Compile preliminary thesis report	

7. CONCLUSION

There is need for approaches that automates the generation of feature sets that satisfies certain collection of criteria (requirements and preferences) in an interactive manner. In a PLaaS context, providers can benefit from such techniques to minimize the operational cost for CPU, Memory and bandwidth required to attract and retain potential consumers; also, consumers can obtain satisfactory results by a flexible process that considers specific preferences. A desirable approach that optimize configuration based on preference information provided by the consumers is the goal of this doctoral research. Table 1 presents the summary of the work plan for the next 20 Months (five quarters).

8. REFERENCES

[1] Schmid, K and Rummler, A. Cloud-based Software Product lines. In *SPLC' 12* (2012), 164-170, ACM.

[2] Kang, K., Cohen, S., Hess, J., Novak, W., and Peterson, S. *Feature–Oriented Domain Analysis (FODA) Feasibility*. Software Engineering Institute, Carnegie Mellon University. 1990.

[3] Benavides, D., Segura, S., and Ruiz-Cortes, A. Automated analysis of feature models 20 years later: A literature review. *Information Systems* (2010), 615-636.

[4] Rosenmüller, M., Siegmund, N., Thüm, T., and Saake, G. Multi-Dimensional Variability Modeling. In *VaMoS '11* (Numar, Belgium 2011), ACM.

[5] Deelstra, S., Sinnema, M., and Bosch, J. Product derivation in software product families: a case study. *Journal of Systems and Software* (2005), 173-194.

[6] White, J., Schmidt, D. C., Benavides, D., Trinidad, P., and Ruiz–Cortés, A. Automated Diagnosis of Product-line Configuration Errors in Feature Models. In *Proceedings of 12th International Software Product Line Conference* (2008), IEEE Computer Society, 225 -234.

[7] Rabiser, R., Wolfinger, R., and Grunbacher, P. Three-level Customization of Software Products Using a Product Line Approach. In *42nd International Conference on System Sciences HICSS'09* (Hawaii 2009), IEEE, 1-10.

[8] Benavides, D., Ruiz–Corte´s, A., Trinidad, P., and Segura, S. A Survey On The Automated Analyses Of Feature Model. *Jornadas de Ingenier´ıa del Software y Bases de Datos* (2006).

[9] Karataş, A. S., Oğuztüzün, H., and Doğru, A. From extended feature models to constraint logic programming. *Science of Computer Programming* (2012), In-Press.

[10] Elfaki, A. O., Abouabdalla, O. A., Fong, S, Johar, Md Gapar Md, A., Kevin L.T., and Bachok, R. Review and Future Directions Of The Automated Validation In Software Product Line Engineering. *Journal of Theoretical and Applied Information Technology* (2012), 75-93.

[11] Trinidad, P., Benavides, D., Duran, A., Ruiz-Cortes, A., and Toro, M. Automated error analysis for the agilization of feature modeling. *The Journal of Systems and Software*, 81 (2008), 883-896.

[12] Guo, J., White, J., Li, J., and Wanga, Y. A genetic algorithm for optimized feature selection with resource constraints in software product lines. *The Journal of Systems and Software* (2011), 2208– 2221.
[13] Sinha, A., Korhonen, P., and Wallenius, J. An Improved Progressively Interactive Evolutionary Multi-objective Optimization Algorithm with a Fixed Budget of Decision Maker Calls.
[14] Miettinen, K., Ruiz, F., and Wierzbicki, A. P. Introduction to Multiobjective Optimization: Interactive Approaches. In *Multiobjective optimization: Interactive and evolutionary approaches*. Springer-Verlag, Berlin Heidelberg, 2008.
[15] Chong, F. and Carraro, G. *Architecture Strategies for Catching the Long Tail*. 2006.
[16] Rimal, B. P., Jukan, A., Katsaros, D., and Goeleven, Y.. Architectural Requirements for Cloud Computing systems: An Enterprise Cloud Approach. *Journal of Grid Computing*, 9, 1 (March 2011), 3-26. Doi: 10.1007/s10723-010-9171-y.
[17] Duijvestijn, L., Fernandes, A., Isom, P., Jewell, D., Jowett, M., Stahl, E., and Stockslager, T. R. *Performance Implications of Cloud Computing*. 2010.
[18] Shi, Y., Luan, S., Li, Q., and Wang, H. A Multi-Tenant Oriented Business Process Customization System. In *International Conference on New Trends in Information and Service Science, NISS'09.* (2009), IEEE, 319-324.
[19] Mell, P. and Grance, T. *The NIST Definition of Cloud Computing*. 2011.
[20] Krebs, R., Momm, C., and Konev, S. Architectural Concerns in Multi-Tenant SaaS Applications. In *2nd International Conference on Cloud Computing and Service Science (CLOSER'12)* (2012), SciTePress, 426-431.
[21] Ruehl, S. T. and Andelfinger, U. Applying Software Product Lines to create Customizable Software-as-a-Service Applications. In *Proceedings of the Software Product line Conference (SPLC)* (Munich, Germany 2011), ACM.
[22] Mietzner, R., Metzger, A., Leymann, F., and Pohl, K. Variability Modeling to Support Customization and Deployment of Multi-Tenant-Aware Software as a Service Applications. In *ICSE Workshop* (Vancouver, Canada 2009), IEEE, 18-25.
[23] Schroeter, J., Cech, S., Götz, S., Wilke, C., and Aßmann, U. Towards Modeling a Variable Architecture for Multi-Tenant SaaS-Applications. In *Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive System (VaMoS '12)* (2012), ACM New York, NY, USA, 111-120.
[24] Hartmann, H. and Trew, T. Using Feature diagrams with Context Variability to model Multiple Product Lines for Software Supply Chains. In *12th International Software Product Line Conference* (2008), IEEE, 12-21.
[25] Classen, A., Boucher, Q., and Heymans, P. A text-based approach to feature modelling: Syntax and semantics of TVL. *Science of Computer Programming* (2011), 1130-1143.
[26] Batory, D., Benavides, D., and Ruiz-Cortes, A. Automated analysis of feature models: challenges ahead. *Communications of the ACM* (December 2006), 45-47.
[27] Ho, W., Xu, X., and Dey, P. K. Multi-criteria decision making approaches for supplier evaluation and selection: A literature review. *European Journal of Operational Research* (2010), 16-24.
[28] Triantaphyllou, E. *Multi-criteria decision making methods: a comparative study*. Kluwer Academic Publishers, Dordrecht, 2000.
[29] Lin, Y., Ye, H., and Tang, J. An Approach to Efficient Product Configuration in Software Product Lines. In Bosch, Jan and Lee, Jaejoon, eds., *Software Product Lines: Going Beyond*. 2010.
[30] Roos-Frantz, F., Benavides, D., Ruiz-Cortes, A., Heuer, A., and Lauenroth, K. Quality-aware analysis in product line engineering with the orthogonal variability model. *Software Quality Journal* (2011). Available at: DOI 10.1007/s11219-011-9156-5.
[31] Soltani, S., Asadi, M., Gaševi, D., Hatala, M., and Bagheri, E. Automated Planning for Feature Model Configuration based on Functional and Non-Functional Requirements. In *Software ProductLine Conference* (Salvador - Brazil 2012), ACM , 56-65.
[32] Mendonca, M., Branco, M., and Cowan, D. SPLOT: software product lines online tools. In *24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications* (2009), ACM, 761-762.
[33] Siegmund, N., Rosenmuller, M., Kuhlemann, M., Kastner, C., Apel, S., and Saake, G. SPL Conqueror: Toward Optimization of Non-functional Properties in Software Product Lines. *Software Quality Journal* (2012), 487-517.
[34] Olaechea, R., Stewart, S., Czarnecki, K., and Rayside, D. Modelling and Multi-Objective Optimization of Quality Attributes in Variability-Rich Software. In *NFPinDSML2012* (2012).
[35] Siegmund, N., Kuhlemann, M., Apel, S., & Pukall, M. Optimizing non-functional properties of software product lines by means of refactorings. In *VaMoS' 10* (2010), 115-122.
[36] Hevner, A.L., March, S.T., Park, J., and Ram, S. Design Science in Information Systems Research. *MIS Quaterly*, 28, 1 (2004), 75-105.
[37] Adigun M.O., Emuoyibofarhe, O.J, and Migiro S.O. Challenges to Access and Opportunity to use SMME enabling Technologies in Africa. In *All Africa Technology Diffusion Conference* (June 2006).
[38] Segura, S, Galindo, J, Benavides, D, Parejo, J, and Ruiz-Cortés, A. BeTTy: Benchmarking and Testing on the Automated Analysis of Feature Models. In *VAMOS '12* (Leipzig, Germany 2012), ACM, 63-71.
[39] Buyya, R., Ranjan, R., and Calheiros, R. N. Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities. In *International Conference on High Performance Computing & Simulation* (2009), IEEE, 1-11.