2012 4th International Conference on Computer Engineering and Technology (ICCET 2012)
IPCSIT vol.40 (2012) © (2012) IACSIT Press, Singapore

A Genetic Algorithm for Configuration of the Business Process Families

Ivana Ognjanović[1] and Ramo Šendelj[2]

[1,2] University Mediterranean, Montenegro

Abstract. Business process model families (BPMF) capture all of the possible business processes that can be defined for a target domain of interest. Configuration problem which arises is defined as the problem of both, unique business process model derivation, and services selection optimally implementing the activities of the process model.There is increasing number of available services per each activity in the BPMF, with the same functionality which differentiates on QoS characteristics. We propose use of Genetic Algorithms (GA) as a technique which finds near optimal solution of combinatorial problems, adopt it to the problem of BPMF configuration and simulation analyze its efficiency for considering domain.

Keywords: genetic algorithm, configuration, business process model families

1. Introduction

Software Product Line Engineering (SPLE) is a systematic approach to enabling the development of a family of software systems. One of the key concepts in SPLE is variability modeling, which aims at capturing how different options for a decision point can impact the way a software system is designed. Several researchers have proposed to integrate Service-Oriented Architecture (SOA) and SPLE paradigms into the new Service-Oriented Software Product Lines (SOSPLs) idea in order to allow for the development of customizable service-oriented products, and hence, benefit from the synergies of both paradigms [1,2].

Within the context of SOSPL, a reference process model is a template that represents a Business Process Model Family (BPMF), which can be instantiated, customized and tailored in order to meet specific needs. A BPMF comprises of a careful collection of business processes for an entire target domain. Business process families capture all of the possible business processes that can be defined for a target domain of interest. This paper deals mainly with BPMF configuration problem from the aspect of quality-of-service (QoS) induced by selection of appropriate services from the set of candidate, all of which provide equivalent functionalities but with different degree of quality characteristics related to the service quality specification. Since, there may be a large number of available services delegated to each activity, combinatorial size of the configuration problem imposes the use of optimization approaches.

In the literature, widely adopted approach in Integer Linear Programming (ILP) which is, later in the Related Work section, shown to have limitations for this purpose. As a response, this paper uses Genetic Algorithms (GA) as a technique which finds near optimal solution of combinatorial problems, and the closeness to the optimal solution is used as an efficiency criterion which is simulation estimated. The paper is organized as follows: background information are reported in Section. 2. The adoption of GA is presented

[1]Ivana Ognjanović. Tel.+38267819450
E-mail address: ivana.ognjanovic@unimediteran.net

in Section 3 and simulation results are reported in the Section 4. Section 5 concludes the paper with related work and future research directions.

2. Families of business processes

Variability can occurs in different phases of development of an SOSPL lifecycle. Design time (architectural) variability defines variability at the design time of a system. Runtime (behavioural) variability manages variation during execution. The approach which will be used through the paper, encode the architectural and behavioral variability by means of two models, i.e., feature models and business process models, respectively.

2.1 Feature Model

In SPLE, feature modelling is a technique for scoping and representing the commonality and variability of features in a product family, for instance, deciding which features should be supported by the common assets and which not, identifying architectural variation points, and in system customization [6]. A feature model has a formal semantics and graphical representation which are visually represented as a rooted directed acyclic graph (i.e., tree-like structure), where nodes represent the features and the edges illustrate the relationship among them (e.g., Figure 1(a)). Feature models define and represent at least the following relationships among features [5]: a) Optional: if a feature is defined as optional, it can be optionally included in product if its parent is selected; b) Mandatory: a mandatory feature is included in all products if its parent is selected; c) Alternative: a set of features are defined as alternative if only one feature is selected when its parent is selected to be included in products; d) Or: a set of features are defined as or-relation with their parent when one or more of them based on defined cardinality can be included in products in which their parents is selected.

In addition to parental relationships, a feature model also can contains integrity constrains between features. These constraints can be specified in order to describe inclusion or mutual exclusion relations between features. For instance, if feature f_i requires f_j, the inclusion of fi in a product line implies the inclusion of f_j. In contrast, if feature f_i excludes f_j, both features cannot be included in the same product. Figure 1(a) shows a part of simplified feature model for an e-shop family.

2.2 Reference Business Process Model

A business process model comprising behavioral variability (runtime variability) includes process specifications, a control flow over a set of activities and events where an activity represents a functional abstraction of services. An activity can be 1) atomic (a.k.a, task) or 2) nonatomic (a.k.a, sub-process). Each activity is delegated and bound to one or more services which provide the required functionality with different quality characteristics.

A reference process model represents a Business Process Model Family (BPMF) as a template which can be instantiated, customized and tailored in to meet situation specific needs. A BPMF comprises a union of the business processes for the entire family in a superimposed way. In our approach, the various parts of the reference business processes are organized in variation points, which can be further managed and configured by means of feature model. For instance, some activities should be included in some products and some activities are required to be excluded for a specific application according to functional and quality requirements. Hence, we will assume that there is a mapping model available which interconnects these two models. This injective mapping (i.e., one-to-one) reciprocally links each feature in the feature model (e.g., Figure 1(a)) to the corresponding activity in the reference business process model (e.g., Figure 1(b)).

3. Adoption of GA for BPMF configuration

The ***optimal configuration problem of the business process model family*** is defined in [13] as a problem of unique business process model derivation, such that overall quality characteristics are maximized. For the derived model, the services should be selected optimally implementing the activities of the process model.

3.1 QoS degree measurements

In our approach we use model proposed in [7] to analyse and monitor QoS characteristics. This model makes it possible to compute the aggregated quality of each configured BPM from the family, based on atomic task QoS attributes and workflow composition patterns. Hence, given a combination of services which further configure a BPM, the overall QoS, can be computed by applying the aggregation rules recursively to compound nodes of BPM structure.

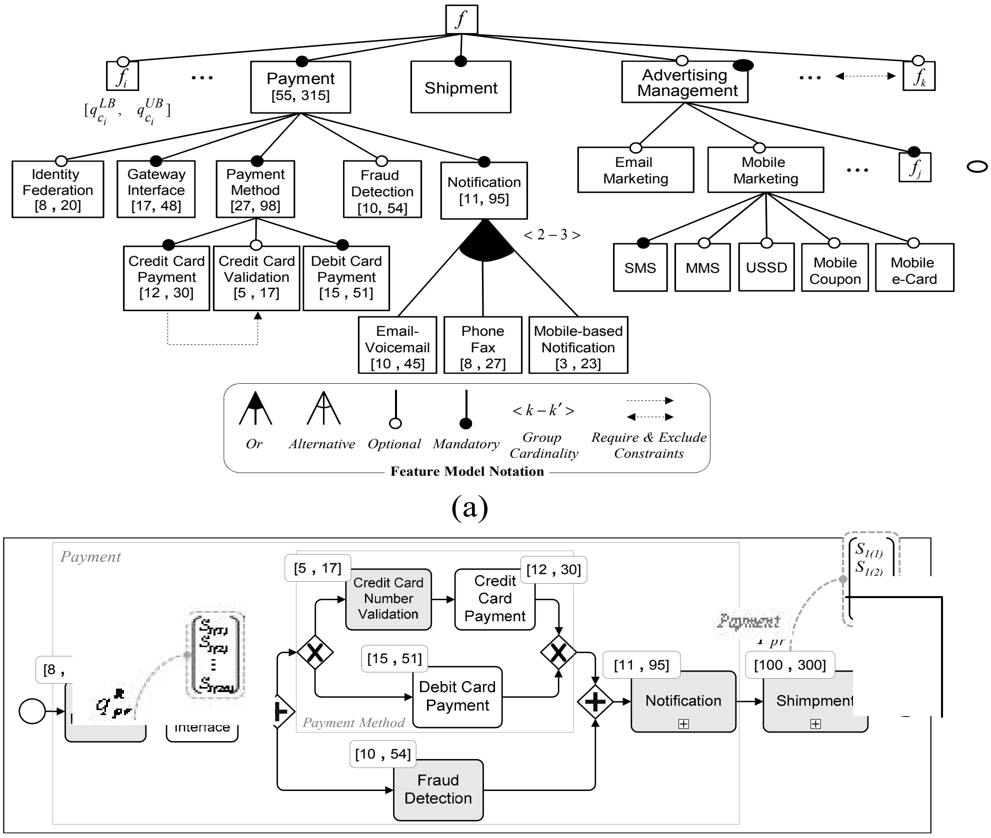

Fig.1.a) High-level representation of feature modelof e-shop. b) A part of business process model family representing the realization and composition of payment and shipment features[8]

Furthermore, quality evaluation is a challenging task even more complex when it comes to SOSPL as it needs to analyze the attributes of a family of SOA systems [8]. A framework for evaluating the quality ranges in an SOSPL proposed in [8] and takes into account both variability and composition patterns during architecture quality aggregation. This method estimates the range values of each QoS dimension for the whole family reachable by different combinations of services. To exemplify, let us consider that estimated range values of Cost for the BPMF represented on the Fug.1.are [100, 300]. Furthermore, let us suppose that we have a combination of services which aggregated Cost value is 150. Based on these values and decreasing tendency of Cost characteristic, we can assign the quality degree to considered configuration as follows: $1-(150-100)/(300-100)$=0,75. On the other side, if we consider Availability as increasing QoS characteristic, with range [0.7, 0.9], quality degree of the configuration which gives aggregated value of 0.80 is calculated as follows:$(0{,}8-0{,}7)/(0{,}9-0{,}7)=0{,}50$.

Proposed measurement allows us to define quality degree of considered combination of services as the value of predefined function (e.g. mean, sum, min) of quality degrees among each QoS dimension. So, the problem of BPMF configuration is reformulated into finding the valid configuration of BPMF which maximizes the total quality degree. We use Genetic algorithms (GA), in order to evaluate different combinations of services that satisfy the constraint defined in the BPMF and its corresponding feature model, measure their quality degrees and select the best one as follows.

3.2 GA adoption

GAs are a form of metaheuristic search that solve problems using algorithms inspired by the processes of the neo-Darwinian evolutionary theory [4]. They are an iterative procedure based on a constant-size population (called “chromosomes”). Each individual population describes a solution and new ones are produced by applying mutation or crossover operators. The crossover operator takes two individuals (called “parents”) of the old generation and combines their bits (called “genomes”) to produce one or more individuals (called “offsprings”). The mutation operator has been introduced to prevent convergence to local optima, in that it randomly makes modifications on individual genomes. New sets of chromosomes representing the possible solution to the optimization problem are constantly generated through different generations and are evaluated for their suitability using a fitness function. This process often continues until an adequate solution is found [4].

Here we demonstrate the adoption of each element of GA, as follows.

Service Chromosome Encoding. An n-dimensional array encoding is used to represent a potential solution (i.e. one combination of services) as a chromosome (n- the number of atomic activities in the BPMF). If the ith activity is mandatory, the set of possible values for i-thelement in the array is the set of available services. On the other side, the ith activity or any of its ancestors is optional, the set of possible values for i-thelement in the array additionally includes 0 (representing that activity is not included in the BPMF).

Initial Population. To employ a GA for service selection in the BPMF, a group of service sets must be generated to obtain an initial population. They are generated randomly, and hence, initial service sets may not be valid service combinations that confirm to the optionality and integrity constraints defined by the feature model and parental relationships in the business process model. In order to solve the problem of generating invalid elements in the population, we use the approach similar to [3] which in predefined number of steps transforms randomly generated sets into valid service selections.

Fitness Evaluation. In order to evaluate the solutions and select the best one, a fitness function which represents the performance of each individual is needed [4]. To generate and select the set of services with the best performance, we need to optimize the overall function. As fitness function should measure the quality of each element of the population, we use proposed quality degree $f(\langle e_1,...,e_k \rangle)$. Additionally, it is necessary to ensure that each individual satisfies problem constraints, i.e. individuals which do not satisfy cannot be obtained as a final result. Stakeholders are allowed to define structural constraints for special sub-processes limiting the corresponding quality properties, in the form $cl_i(e_1, \dots, e_k) > u_i$. Thus, the distance from constraint satisfaction, according to [12], might be defined as: $D(e_1, \dots, e_k) = \sum_{i=1}^{l} cl_i(e_1, \dots, e_k) y_i$, where$y_i = 0, cl_i(e_1, \dots, e_k) \leq u_i$ and $y_i = 1, cl_i(e_1, \dots, e_k) > u_i$.

In the literature [2,9], there is recommended to have penalty factors dynamically increased with the number of generations. The main reason is in the risk that individuals will not be discarded although they violate the constraints [2]. So, by advancing in the iteration process, penalty weights are getting increased which ensures that fitness function (which should be maximized) is decreased for individuals which violate constraints. Finally, our fitness function is defined as:$Fitness(e_1, \dots, e_k) = f(e_1, \dots, e_k) - w_1(gen) D(e_1, \dots, e_k)$.

Crossover and Mutation. Because of the non-binary nature of our genomes, we use standard k-point crossover and random point mutation operators [4]. As a result of both operations, invalid chromosomes might be generated. To handle this, we need a reparation method that restores feasibility in the chromosome and for this purpose we employ the same algorithm from [3] that was introduced earlier.

Termination Criterion. Iterations are performed until the best fitness individual remains unchanged for a given number of iterations. The adoption of GA is presented on the Figure below.

4. Simulation Analysis

As our approach uses GAs, it is necessary to analyze how it is efficient and effective for the problem of BPMF configuration. We have chosen the closeness to an optimal solution as the criterion which is used to estimate the efficiency of our approach.

Algorithm. *GA adoption for BPMF configuration*

Input: Feature model FM, BPM family, set of available services S_{a_k} and theirQoSvalues
Output. Configured BPM

```
Begin
estimate ranges of QoS (FM, BPMF, S_ak );
  Begin    // Genetic algorithm (FM, BPMF, S_ak );
    initialize population;
    evaluate population with calculation of QoS measurements;
  while TerminationCriteriaNotSatisfied
   do
    select parents for reproduction;
    perform crossover and mutation;
    evaluate population with calculation of QoS measurements;
   end while;
  end
End.
```

As we previously indicated, one of the major characteristics of BPMG configuration problem is induced by variability and integrity constraints defined in both models (FM and BPMF). In that sense, we decided to analyze how different distributions of these constraints (defined by percentage ratio values of their occurrences) influence closeness to an optimal solution. We use FM generator developed in [4] and implement our framework as an Eclipse plug-in. Also, we implemented brute-force algorithm in order to obtain the optimal solution and compare it with the solution obtained by our approach.For estimating the average distance to an optimal solution, we performed random experiment. We use standard descriptive statistics (as reported in [11] to be a common practice) including mean (M) and standard deviation (SD) values for the analyses of experimentally collected data.

In the previous empirical research of feature models [10], we found that the number of features in the feature models was in the range between 14 and 287 where M=76.86 and SD=96.25. Accordingly, we performed our experiments with FMs and BPMFs of the size equal to this empirically found mean values. The maximum number of available services per activity was set to 100 and their values of QoS characteristics are generated randomly. Each simulation is performed 1000 times and percentage ratio values of each pattern in both models are generated randomly. The mean values of the obtained relative distances in the configuration is calculated and is equal to 10.78% (SD=1.087%). Thus, the optimality of our approach is around 90%, i.e., the rest of our approach has approximately 10% lower estimated quality as compared to the optimal solution.

5. Related Work

Maximizing the quality of service composition and configuration is essentially a multi-objective optimization problem, which is known to be NP-hard [12]. AI planning and software synthesis techniques have received much attention recently among semantic web researchers as promising ways to plan service composition without human intervention [17].

GAs are used for a wide range of different problems and GAs often work well it is shown to work well for highly constrained problems, such as the Traveling Salesman [14] and Satisfiability problems [15].Furthermore, they are already adopted for the problem of optimal feature selection [3] and web service selection [12].

As an alternative approach for optimization problems is Linear Integer Programming [16]. It suffers from linearity requirements in each element, which is hardly to be met in the problem of BPMF configuration. More precisely, the linearization approaches should be applied for representation of integrity constraints in the FM and non sequential patterns in the BPM. Furthermore, increasing number of available services imposes huge number of variables in the LIP, which represents its limitation in performance characteristics [12,13].

To the best of our knowledge, none of the existing approaches perform service selection in presence of variability within composition specifications, as we will discuss in this paper in the context of business process families (and SPLE).

6. Conclusions

This paper has presented a novel approach for configuration on business process families based on the use of genetic algorithms to maximize the overall quality degree. It is simulation shown that this approach gives effective near optimal solution, but it is needed to further analyze how different input parameters of the whole approach influence on its performance characteristics. Furthermore, in our future research, we will allow stakeholders to define their own preferences about the most preferable quality characteristics which should be incorporated in the configuration problem. We will also (empirically) compare the proposed genetic algorithm-based approach to alternative based on integer linear programming.

7. Reference

[1] Rosa, M.L., Dumas, M., ter Hofstede, A.H.M., Mendling, J.:Configurable multi-perspective business process models. *Information Systems*36(2) (2011) 313–340

[2] Rosemann, M., van der Aalst, W.M.P.: A configurable reference modelling language. *Inf.Systems*32 (2007) 1–23

[3] Guo, J., White, J., Wang, G., Li, J., Wang, Y.: A genetic algorithm for optimized feature selection with resource constraints in software product lines. J. on *Systems and Software*84 (December 2011) 2208–2221

[4] Srinivas, M., Patnaik, L.: Genetic algorithms: a survey. Computer 27(6) (1994) 17–26

[5] Batory, D. "Feature Models, Grammars, and Propositional Formulas,“ *In Proc. of the 9th Int. Conf. on SPLs*, 2005, pp. 7-20.

[6] Lee, K., Kang, K. C., and Lee, J., "Concepts and Guidelines of Feature Modeling for Product Line Software Engineering," *In Proc. of the 7th Int. Conf. on Software*, pp 62-77, 2002.

[7] Cardoso, J., Sheth, A.P., Miller, J.A., Arnold, J., Kochut, K.: Quality of service for workflows and web service processes. J. Web Sem. 1 (2004) 281–308

[8] B.Mohabbati, D.Gašević, M.Hatala, M.Asadi, E.Bagheri, M.Bošković, "A Quality Aggregation Model for Service-Oriented Software Product Lines based on Variability and Composition Patterns, " In *Proceedings of the 9th International Conference on Service Oriented Computing,* Paphos, Cyprus, 2011

[9] Hsiao-LanFang,Genetic Algorithms in Timetabling and Scheduling, *PhD thesis, University of Edimburg*, 1994

[10] Bagheri, E., Noia, T.D., Gasevic, D., Ragone, A.: Formalizing interactive staged feature model configuration. J. of Software Maintenance and Evolution: *Research and Practice* (2011)

[11] Blaikie, N.: Analysis of Quantitative Data: From Description to Explanation. *SAGE* Publications Ltd (2003)

[12] G.Canfora, M.Penta, R. Esposito, M.Villani, An Approach for QoSaware Service Composition based on Genetic Algorithms, *GECCO'05*, USA

[13] Mohabbati, B., Asadi, M., Bošković, M., Gašević, D., Hatala, M., "Development and configuration of service-oriented software product lines," In *Proceedings of the 26th ACM Symposium on Applied Computing*, TaiChung, Taiwan, 2011, pp. 1606-1613

[14] Muhlenbein, H., 1989. Parallel Genetic Algorithms Population Genetics and Combinatorial Optimization. In: *Proceedings of ICGA'89*, George Mason University, Fairfax, Virginia, USA, pp. 416-421.

[15] De Jong, K. A., Spears, W. M., 1989. Using Genetic Algorithms to Solve NP-Complete Problems. In*: Proceedings of ICGA'89*, George Mason University, Fairfax, Virginia, USA, pp. 124-132.

[16] Alexander Schrijver, Theory of linear and integer programming, *John Wiley & Sons*, 1998

[17] M. Tang, L. Ai, A Hybrid Genetic Algorithm for the Optimal Constarined Web Service Selection Problem in Web Service Composition, *Proc. of the 2010 World Congress on Computational Intelligence*, *IEEE*, Barcelona.