A Multi-Objective Hybrid Particle Swarm Optimization-based Service Identification

Mohamed Merabet
Higher National School of Computer Sciences,
Algiers, Algeria
m_merabet@esi.dz

Sidi Mohamed Benslimane
Computer Sciences Department,
DjillaliLiabes University, Sidi Bel Abbes, Algeria
benslimane@univ-sba.dz

Abstract – Service identification step is a basic requirement for a detailed design and implementation of services in a Service Oriented Architecture (SOA). Existing methods for service identification ignore the automation capability while providing human based prescriptive guidelines, which mostly are not applicable at enterprise scales. In this paper, we propose a top down approach to identify automatically services from business process. We use for clustering a hybrid particle swarm optimization algorithm and several design metrics for produce reusable services with proper granularity and acceptable level of cohesion and coupling. The experimental results show that our method HPSOSI (Hybrid Particle Swarm Algorithm for Service Identification) can achieve a high performance in terms of execution time, and significant quality in terms of high modularization, strong cohesion, and weak coupling of the identified services..

Keywords – Service Identification, Hybrid Particle Swarm Optimization, Service Oriented Architecture, Business Process Modeling

1. INTRODUCTION

Frequent changes in the business environment and user demands are two important challenges in developing large-scale software systems. Service-Oriented Architecture (SOA) is one of the promising methods to address these challenges [17]. In service-oriented architecture, the first phase of the SOA Lifecycle is identification of services. This phase not only determines the services that must be implemented, but also define the logic that must encapsulate each service. Service is the basic unit in Service Oriented Computing. In design methodologies, service identification usually plays a critical role. The Lifecycle of SOA delivery projects is simply comprised of a series of steps that need to be complete to construct the services in a given service-oriented solution, one of the most important steps is to identify services [4].

Existing identification methods can be classified into three categories based on the automation point of view: prescriptive, semi-automated, and fully-automated. Because SOA generally is used to develop large-scale software systems, hence, using prescriptive methods may lead to low of identifying architectural elements.

The need for an automated approach to identify services is recognized to make service identification step more reliable.

Most of automated approaches use Meta-heuristic algorithms for service identification, such as Simulated Annealing [6], Genetic Algorithm [10], [15], Combinatorial particle swarm optimization [8].

In this paper, a new identification approach is proposed, which aims to resolve the above problems by supporting automation capabilities, adopting technical metrics, and using business process as input. This method generates candidate software services using multi-objective hybrid particle swarm optimization algorithm in order to group them into distinct services represented as clusters by analyzing the dependencies between business activities and business entity. The proposed approach combines two Meta-heuristic algorithms that are PSO (particle swarm optimization) and GA (Genetic Algorithm). We use a hybrid algorithm by embedding the crossover and mutation operators of Genetic Algorithm into the PSO.

The rest of this paper is organized as follows. In section 2, the most related work is briefly reviewed. In Section 3 introduces to the basic concepts the particle swarm optimization. In section 4,a novel method is proposed in order to solve the service identification problem based on hybrid Particle Swarm Optimization. Section 5 provides an implementation and experimental results to demonstrate the performance of our approach. Finally, section 6 concludes the paper and point directions for future work.

2. Model-Driven service identification process

The literature provides several works in service identification. Who uses various levels of automation: prescriptive, semi-automated, and fully-automated.

In this section, the existing approaches for service identification can broadly be separated into one of the following categories:

2.1 Non-automated Methods (*prescriptive, semi-automated*):

[5]mentioned some best practices, wherever appropriate, to point out the vagueness involved in service identification. A top-down and bottom-up techniques for service identification has proposed in this methodology.

[9] presented a method of service identification using an ontology for the product line. Primary, Semantic relationship was derived through the mapping between feature modeling and ontology. Second, both service and service boundary was defined by semantic distance. Third, the method was proposed for feature grouping and candidate service refining service candidate which is the fittest service granularity.

[14] proposed a generic ontology-based framework, BPAOnto-SOA, for the derivation of software service oriented models from a given business process architecture relying on two ontologies. This framework utilizes an adapted clustering algorithm based on affinity analysis of business process functions in order to group them into services along with their associated NFRs to ensure conformance of these services with SOA principles.

[2] proposed a top-down approach for services identification from business process models, applying heuristics to define services for the semantic analysis of process elements such as business rules and business requirements, and from a syntactic analysis of process models according to its corresponding structural patterns.

[16] introduced a service identification method based on scenario modeling and a conceptual framework to elicit possible business changes. Traceability among business requirements, business changes and the identified services are also supported by their method.

[19] presented a new method to identify services from business process models. The contribution of this approach is to show how the business architecture can be used to drive the organization of the information systems architecture and to ensure its alignment with business requirements.

2.2 Automated Methods

In [10], an automated method for identifying business services by adopting design metrics based on top-down decomposition of processes is presented. This method takes a set of enterprise business processes as input and produces a set of non-dominated solutions representing appropriate business services using a multi-objective genetic algorithm.

[6] introduced a novel approach called ASIM for automatically identifying and partly specifying enterprise-level software services for business models using best practices and principles of model-driven software development. They formulated service identification as a multi-objective optimization problem and solved it by a novel meta-heuristic optimization algorithm that derives appropriate service abstractions by using appropriate quantitative measures for granularity, coupling, cohesion, reusability, and maintainability.

In [15], a meta-heuristic approach called MOOSI for deriving service-oriented architectures from annotated business process model is proposed. They generate candidate software services using multi-objective evolutionary algorithm that analyses dependencies between business activities in order to group them into distinct clusters, each cluster must groups one or more closely related activity to form a future software service.

[8] Proposed a top down approach to identify automatically services from business process by using several design metrics. The approach called CPSO produce services from business processes as input and use a clustering combinatorial particle swarm optimization algorithm.

In [18], the P2S (Process-to-Services) methodology for service identification is applied in a top-down context or in any case in which a portfolio of available services is not present. The methodology is designed to implement the guidelines for service design, ensuring at the same time proper metrics to automate service identification.

3. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO), is a new intelligent global optimization algorithm, it was proposed in 1995 by Eberhart and Dr. Kennedy [11]. It is inspired by the swarming behavior of animals and human social behavior. They developed this method for optimization of continuous non-linear functions. The algorithm is similar to other population-based algorithms like Genetic algorithms, but there is no direct combination of individuals of the population. Instead, it relies on the social behavior of the particles. In every generation, each particle adjusts its trajectory based on its best position (local best) and the position of the best particle (Global best) of the entire population. This concept increases the stochastic nature of the particle and converges quickly to a global minimum with a reasonable good solution.

3.1 Standard *PSO Algorithm*

PSO uses to solve an optimization problem, a swarm of computational elements, called particles to explore the solution space for an optimum solution. [11] Each particle represents a candidate solution and is identified with specific coordinates in the N dimensional search space. The position of the i'th particle is represented as $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$. The velocity of a particle is denoted as $V_i = (v_{i1}, v_{i2} \ldots\ldots v_{in})$. The fitness function is evaluated for each particle in the swarm and is compared to the fitness of the best previous result for that particle and to the fitness of the best particle among all particles in the swarm. After finding the two best values, the particles evolve by updating their velocities and positions according to the following equations :

$$\upsilon_i^{k+1} = \mathrm{w} \times \upsilon_i^k + c_1 r_1 \left(p_{i\_best} - x_i^k\right) + c_2 r_2 \left(p_{g\_best} - x_i^k\right)$$

$$\mathrm{x}_i^{k+1} = \mathrm{x}_i^k + \upsilon_i^{k+1}$$

where i = (1, 2, …, swarm_*size)*; p_{i_best} is the particle best reached solution and p_{g_best} is the global best solution in the swarm, k represents the k-generation algebra evolution, w is the inertia weight, c_1 and c_2 are the acceleration coefficient, r_1 and r_2 are random numbers between 0 and 1 used to maintain the diversity of the population, and v_{ij} is a linear combination of three parts.

3.2 *Hybrid PSO algorithm*

PSO is suitable for continuous areas. [7], proposed HPSO (Hybrid Particle Swarm Algorithm) for solving problems in discrete areas by introducing the crossover strategy and the mutation strategy of the genetic algorithm.

$\mathrm{c}_1\mathrm{r}_1\left(p_{i\_best} - \mathrm{x}_\mathrm{i}^\mathrm{k}\right) + \mathrm{c}_2\mathrm{r}_2\left(p_{g\_best} - \mathrm{x}_\mathrm{i}^\mathrm{k}\right)$ is viewed as a crossover operator that make the current solution do crossover operation on individual extremum, and make the result do a crossover operation on global extremum.

$\mathrm{w} \times \upsilon_\mathrm{i}^\mathrm{k}$ can be viewed as a mutation operator of genetic algorithm. The variation of the above results is a new location[7].

For our problem, we will use hybrid PSO by introducing the crossover and mutation operators as follow:

(1) Crossover operator

We propose crossover operators that ensure consistency of solutions as follows:

1. Select a particle with a given crossover probability $\mathrm{P_c}$.

2. For the current particle and a business process which is composed of n activities and m business entity, randomly select a position i, (where i = 1,…,n or i = 1,…,m).

3. The value of the selected position in the first sequence will exchange with the value of corresponding position in the second sequence.

(2) Mutation operator

In order to guarantee the diversity of the population, we introduce the mutation strategy in genetic algorithm into PSO by randomly select x positions from the sequence, and replace it by a random number ranging in the interval [0...k], where k is the number of clusters.

4. THE PROPOSED APPROACH

Referring to the SOMA [1], different approaches can be adopted to identify services, namely top-down (domain decomposition), bottom-up (existing system analysis) and meet-in-the-middle. The Top-Down approach consists

International Conference on Advanced Aspects of Software Engineering
ICAASE, November, 2-4, 2014, Constantine, Algeria.

mainly in decomposing business processes into finer business tasks. The bottom-up approach is about analyzing existing IT assets and finding functionality that could be exposed as services.

The last one is about to conduct both a bottom-up analysis and top-down analysis and to correlate the services identified by each of these approaches.

The goal of service identification from business process model is organizing a set of business activities into significant clusters (high cohesion and low coupled). The activities of a cluster are considered as the operations of a candidate service.

In our approaches, we begin with the business process model as input (Top-Down) and derive the effective service set based on this model by combining three techniques for grouping business activities in order to form software services:

Business Entity-Centric Technique: it consists in putting all the activities associated with a particular business entity into a service. In this case, a CRUD (Create Read Update Delete) Matrix (CRUDM) is used as input in the clustering algorithm.

Actor-Centric Technique: it consists in putting all the activities performed by a particular actor into a service. In this case, an Activity-Actor Matrix (AAM) is used as input in the clustering algorithm.

Business Goal-Centric Technique: it consists in putting all the activities those participate in the achievement of a particular business goal into a service. For this Technique, we use an Activity Goal Matrix(AGM) as input data in the clustering algorithm.

We use a directed graph to represent dependencies between business activities in order to make the structure of a complex business process more understandable.We call such graphs Activity Dependency Graph (ADG).

In the ADG, the business functions (Business Activities, Sub Process) are represented as nodes and their relationships as edges that connect the nodes. Thus, an Activity A is connected with an Activity B if they manipulate the same Business Entity and/or are performed by the same Actor and/or they participate in the achievement of the same Business Goal.

One way of making ADGs more available is to partitioned them by grouping closely related activities into clusters.

Definition 1 (Cluster)

Let $A_s = (A_1, A_2, , A_n)$, a set of Business Activities of a business information system. Each activity manipulates

one or more Business Entity and is performed by one or more Actor and participate in the achievement of one or more Business Goal.

Formally, a cluster Ci can be defined as:

$C_i = \{A_j\} \mid 1 \leq j \leq |A_s|$ where $A_j \in A_s$.

Definition 2 (Partition)

A partition is a set of non-empty subsets of.

Formally, a partition π can be defined as:

$\pi = \{C_k\} \mid 1 \leq k \leq |A_s|$

Where $\cup_{i=1}^{k} C_i = A_s$ and $\cap_{i=1}^{k} C_i = \emptyset$

In addition, a partition of A_s into k non-empty clusters is called a *k-partition* of A_s. Given a set A_s that contains *n* elements, the number $S_{n,k}$ called *Stirling numbers* of distinct *k-partition* of A_s satisfies the recurrence equation presented by [20].

$$S_{n,k} = \begin{cases} 1 & if\ k = 1\ or\ k = n \\ S_{n-1,k-1} + k * S_{n-1,k} & otherwise \end{cases} \quad (1)$$

Creating a meaningful partition of an ADG is difficult because the number of possible partition is very large even for a small graph. $S_{n,k}$ grow exponentially with respect to the size of A_s. For example, a 7-node ADG would have 877 distinct partitions, while a 25- node ADG would have 4.638.590.332.229.999.353 distinct partitions.

4.1 Design Metrics for Service Development:

An effective way of increasing the quality of software products consists of using metrics to guide the development process. For evaluating and measuring the quality of the services produced by our clustering technique in each iteration, we use three design metrics introduced by [15],[22] and [23] which are: coupling, cohesion, and modularization quality.

Definition 3 (Coupling between two cluster)

The Coupling Between two Clusters (CoupBC) measures the degree of connectivity between two distinct clusters. A high degree of inter-connectivity is undesirable because it indicate that services are highly dependent on each other.

Formally, $CoupBC_{ab}$ (coupling between cluster a and cluster b) measures the ratio of inter-dependencies between cluster a and cluster b and the maximum possible number of inter-dependencies between those clusters.

$$CoupBC_{ab} = \begin{cases} 0 & if\ a = b \\ \dfrac{2 * \sum_{i=1}^{N_a} \sum_{j=1}^{N_b} TDM_{AC_{a_i},\ AC_{b_j}}}{2 * N_a * N_b} & if\ a \neq b \end{cases} \quad (2)$$

Where $N_a$is the number of activities of cluster *a*, N_b is the number of activities of cluster *b*, and the value of $TDM_{AC_{a_i},\ AC_{b_j}}$ (Total Dependency Matrix) represents the degree of dependency between the i^{th} activity of cluster *a* (AC_{a_i}), and the j^{th} activity of cluster *b* ($\mathrm{AC_{b_j}}$).

Definition 4 (Coupling Of a given Cluster)

The Coupling Of a given Cluster (*CoupOC*) measures the degree of connectivity between a given cluster and all other clusters of the system.

Formally, a Coupling Of a given Cluster *a* ($CoupOC_a$)

is defined as:

$$CoupOC_a = \begin{cases} 0 & if\ N_c = 1 \\ \frac{\sum_{b=1}^{N_c} CoupBC_{ab}}{N_c - 1} & otherwise \end{cases} \quad (3)$$

Where Nc is the number of cluster in the system.

Definition 5 (Cohesion Of a Cluster)

Service cohesion refers to the degree of the strength of functional relevance of activities carried out by a service to realise a business process [21].

The degree of cohesion depends on the strength of dependencies between activities of the appropriate cluster (service).

Formally, a Cohesion Of a Cluster *a* ($CohOC_a$) is defined as:

$$CohOC_a = \begin{cases} 0 & if\ n = 1 \\ \dfrac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (2 * TDM_{AC_{a_i},\ AC_{a_j}})}{n * (n-1)} & if\ n > 1 \end{cases} \quad (4)$$

Where *n* is the number of activities in the cluster *a*, *TDM* is the Total Dependency Matrix, AC_{a_i} is the i^{th} Activity of the Cluster *a*, and AC_{a_j} is the j^{th} Activity of the Cluster *a*.

Definition 6 (Modularization Factor)

Modularization Factor (*MF*) is the ratio of the cohesion and the coupling of each cluster.

Formally, the Modularization Factor of a cluster *a* (MF_a) is defined as follows:

$$MF_a = \begin{cases} 0 & if\ CohOC_a + CoupOC_a = 0 \\ \dfrac{CohOC_a}{CohOC_a + \left(\frac{CoupOC_a}{2}\right)} & otherwise \end{cases} \quad (5)$$

Where $CohOC_a$ is the cohesion of the cluster *a*, and $CoupOC_a$ is the coupling of the cluster *a*.

Definition 7 (Modularization Quality)

Modularization Quality (*MQ*) determines the quality of an ADG partition perfectly as the trade-off between interconnectivity (i.e., dependencies between the activities of two distinct services) and intra-connectivity (i.e., dependencies between the activities of the same service).

This trade-off is based on the hypothesis that well-designed service-oriented software systems are organized into cohesive subsystems that are loosely interconnected. Therefore, a high MQ allows the creation of highly cohesive clusters that are not coupled strongly.

Formally, The Modularization Quality (MQ) is defined as the average of the MF's of all clusters:

$$MQ = \frac{1}{n} * \sum_{i=1}^{n} MF_i \quad (6)$$

Where n is the total number of cluster in the system. And MF_i is the Modularization Factor of the i^{th} cluster.

4.2 Service Identification Process

The service identification process starts by querying the Business Process Ontology to extract the existing dependencies between Activity, Business Entity, Actor, and Goal concepts. Three matrices are created to represent the extracted dependencies: (i) the CRUD Matrix (CRUDM) which describes the relations between the Activities and the Business Entities, (ii) the AAM (Activity-Actor Matrix) which describes the relations between the Activities and the Actors, (iii) and the AGM (Activity-Goal Matrix) which describes the relations between the activities and Business Goals (cf. Figure 1).

4.2.1 Activity Relationships Matrices

The CRUD Matrix. For grouping Business Activities into clusters using the Business Entity Centric Technique, we need a rate that represent the degree of semantic relationships between Activity and Business Entity. Four operations can be applied on each Business Entity by the Business Activity. These operations are called CRUD (C:Create, R:Read, U:Update, D:Delete).

In order to compute the value of each technical metrics, such as Cohesion, Coupling, and Modularization Quality, it is necessary to associate each operation with a value between 0 and 1. The majority of existing works on CRUD semantic relationships adopts the following substitutions: C=1; U=0.75, D=0.50, R=0.25. These values represent the cost of the operation on each Business Entity.

We store these values in a CRUD Matrix (CRUDM), which has Business Activities as its rows, Business Entities as its columns, and semantic relationships (”C”, ”U”, ”D”, ”R”, with the distinct intensity 1 => C > U >D > R > 0) as its cells. Each column in this matrix must have exactly one create operation and each row must have no conceptual (operational) activities. Table 1 shows an example of CRUDM.

Activity-Actor Matrix. In order to regroup activities using Actor-Centric Technique, we represent the degree of semantic relationships between Activity and Actor by using a second matrix called Activity-Actor Matrix (AAM). The value of each cell of AAM is taken between 0 and 1. For example the value AAM1,3 = 0.75 indicates that the Actor 3 perform 0.75% of the Activity1. The sum of each row of AAM must be equal 1 to indicate that the corresponding activity is performed completely.

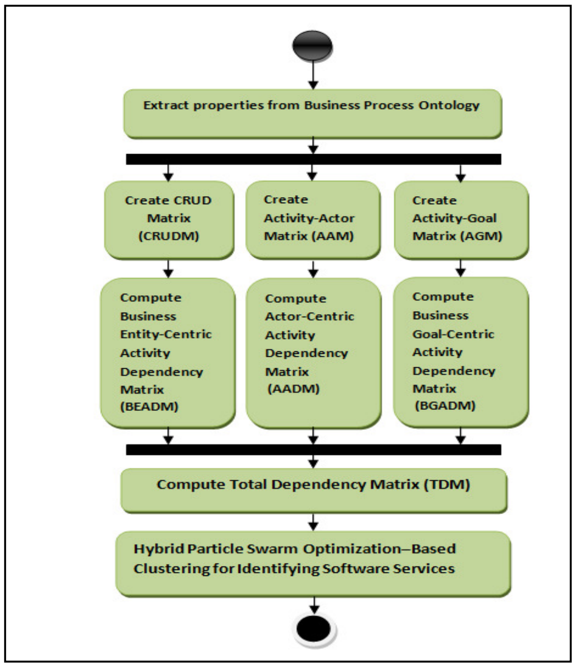

Extract properties from Business Process Ontology
Create CRUD Matrix (CRUDM)
Create Activity-Actor Matrix (AAM)
Create Activity-Goal Matrix (AGM)
Compute Business Entity-Centric Activity Dependency Matrix (BEADM)
Compute Actor-Centric Activity Dependency Matrix (AADM)
Compute Business Goal-Centric Activity Dependency Matrix (BGADM)
Compute Total Dependency Matrix (TDM)
Hybrid Particle Swarm Optimization–Based Clustering for Identifying Software Services

Figure 1: Service Identification Steps

Table 1: CRUD Matrix Example

	BE1	BE2	BE3	BE4	BE5	BE6
Activity 1		C	U	D	C	D
Activity 2	R	D	C	R		U
Activity 3	C				D	
Activity 4			D	C		C
Activity 5	D	R			R	

Activity-Goal Matrix. In order to group Business Activities based on Goal-centric Technique, we use a third matrix called Activity-Goal Matrix (AGM). Each cell of AGM represents the degree of semantic relationships between Activity and Business Goal. The value of each cell of AGM is taken between 0 and 1. It indicates the achievement rate of a Business Goal after the execution of a Business Activity. For instance the value AGM2,4 = 0.50 indicates that the activity 2 Achieve 50% of the Goal 4. The sum of each column of the AGM must be equal 1 to indicate that the corresponding Business Goal is achieved completely.

4.2.2 Total Dependency Matrix derivation

In order to enhance the quality of service cluster, three square matrices are derived from CRUDM, AAM, and AGM respectively. The first one is called Business Entity centric Activity Dependency Matrix (BEADM), the second one is called Actor-centric Activity Dependency Matrix

(AADM), and the third one is called Business Goal centric Activity Dependency Matrix (BGADM). These square matrices represent the degree of dependency between each activity and other activities of the annotated business process. To compute dependencies between two activities, we take the minimal value of relationships that relies the both activities with the same Business Entity.

For instance, the degree of dependency between Activity 1 that read Business Entity 5 and Activity 3 that create the same Business Entity 5 is equal 0.25 (we use the minimal value between R:0.25 and C:1). If no operation is applied by the Activity 2 on Business Entities that are manipulated by Activity 1, we conclude that there is zero dependency between Activity 1 and Activity 2.

The three matrices BEADM, AADM, and BGADM are integrated into one matrix named Total Dependency Matrix (TDM) that represents the average dependency between each couple of activities. Each cell of TDM is calculated as follows:

$$\mathrm{TDM}_{i,j} = \frac{\mathrm{W}_1 * \mathrm{BEADM}_{i,j} + \mathrm{W}_2 * \mathrm{AADM}_{i,j} + \mathrm{W}_3 * \mathrm{BGADM}_{i,j}}{\sum_{i=1}^{3} \mathrm{W}_i}$$

Where W1,W2, and W3 are weighting factors corresponding to BEADM, AADM, BGADM respectively.

4.2.3 Hybrid Particle Swarm Optimization-Based Clustering.

Matrix clustering is considered as NP-complete problem, we have to use Meta-heuristics to solving it.

We propose Hybrid Particle Swarm Optimization [7] for clustering the TDM matrix.

The steps of HPSOSI(Hybrid Particle Swarm Optimization for Service Identification)is mainly divided into three parts:

- First, initialize the particles ;
- Second, calculate the fitness value for each particle and update pbest and gbest;
- Third, use the crossover and mutation operation to improve PSO.

The specific steps are shown in Figure 2.

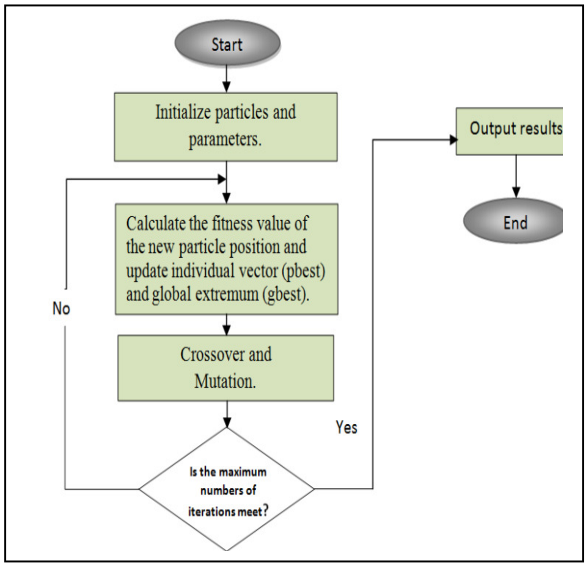

Start
Initialize particles and parameters.
Calculate the fitness value of the new particle position and update individual vector (pbest) and global extremum (gbest).
Crossover and Mutation.
Is the maximum numbers of iterations meet?
No
Yes
Output results
End

Figure 2: The flow chart of HPSOSI

Algorithm: HPSOSI-Hybrid Particle Swarm for Service Identification

Input: CRUD Matrix, Activity-Actor Matrix, Activity-Goal Matrix.
Output: a set of clusters
Step 1: Initialize the particles, make particle dimension as equal to the number of rows of the CRUD Matrix and set the maximum number of iterations.

Step 2: For each particle, calculate it fitness value according to Equation (2), if the fitness value is better than the previous best pbest, set the current fitness value as the new pbest, and for all particles, select the best particule as gbest.

Step 3: For each particle, based on crossover strategy and mutation strategy to update the fitness value of particle and the new location.

Step 4: if the maximum number of iterations is meet, Output the solution (gbest) and its fitness

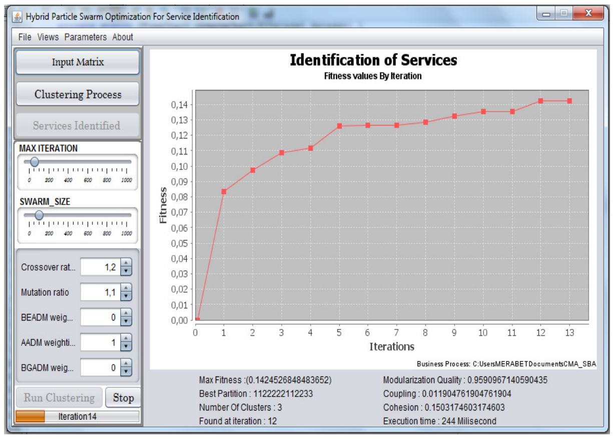
The fitness of a particle is calculated according to the three following objectives: (i) the quality of modularization (ii) the sum of intra-connectivities (iii) the sum of the inter-connectivities.

$$fitness = MQ * \left(\frac{1}{n} * \sum_{i=1}^{n} CohOC_i\right) * (1 - (\frac{1}{n} * \sum_{j=1}^{n} CoupOC_j))$$

Where n is the total number of clusters represented by the current particle, $CohOC_i$ is the cohesion of the the ith cluster, and $CoupOC_j$ is the coupling of the jth cluster. A better particle is a particle that has a strong quality of modularization, a strong cohesion, and a weak coupling which implies a high fitness.

5. IMPLEMENTATION AND EXPERIMENTAL RESULTS

In order to evaluate the performance of our approach we implemented a tool called HPSOSI, Figure 3 is shown a printout screenshot of this latter, which was developed in order using Java programming language with NetBeans IDE 7.0. All experiments run in Windows 7 on a desktop PC with Intel Dual Core, 2.3 GHz processors, 2GB of RAM.


Hybrid Particle Swarm Optimization For Service Identification
File Views Parameters About
Input Matrix
Clustering Process
Services Identified
Identification of Services
Fitness values By Iteration
Fitness
Iterations
Run Clustering
Stop
Iteration14

MAX ITERATION
SWARM_SIZE
Crossover rat... 1,2
Mutation ratio 1,1
BEADM weig... 0
AADM weighti... 1
BGADM weig... 0

Business Process: C:UsersMERABETDocumentsCMA_SBA

Max Fitness :(0.1424526848483652)
Best Partition : 1122222112233
Number Of Clusters : 3
Found at iteration : 12
Modularization Quality : 0.9590967140590435
Coupling : 0.011904761904761904
Cohesion : 0.1503174603174603
Execution time : 244 Milisecond

Figure 3: Interface of the proposed tool.

The HPSOSI parameters were set experimentally as follows: crossover probability P_c and mutation probability P_m we reset to 0.8 and 0.4 respectively and weighting factors W1=1,W2=2,and W3=1. The swarm comprises of 1000 particles and the number of iteration k = 100.

To evaluate the performance of our tool HPSOSI (Hybrid Particle Swarm Optimization for Service Identification), we applied the proposed clustering algorithm on three business processes with different combination of weighting factors. The provided results are shown in Figure 4.

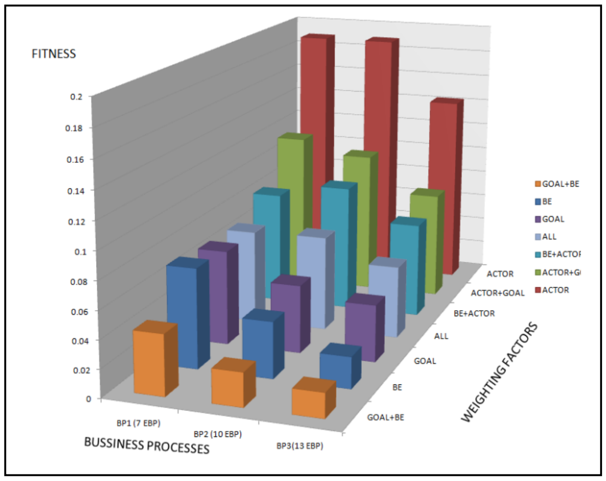


Figure 4: Experimental Results Using Different BP

We note that some factor influences more than others on the fitness value, because the summation of total semantic relationships for matrices CRUDM, AAM, and AGM are different.

There is especially for total semantic relations (TSR) in the matrix AAM is still a higher summation of total semantic relations for CRUDM or AGM matrices. They can be formulated as follows:

$$TSR(AAM) > TSR\ (CRUDM)\ > TSR\ (AGM).$$

Where:

$$TSR(AAM) = \sum_{i=1}^{\#EBP} SR(EBP_i)$$

$$TSR(\mathrm{CRUDM}) = \sum_{i=1}^{\#BP} SR(BP_i)$$

$$TSR(\mathrm{AGM}) = \sum_{i=1}^{\#GOAL} SR(GOAL_i)$$

$SR(EBP_i) = SR(BP_i) = SR(GOAL_i)$ = 1.

$SR(EBP_i)$: Summation of semantic relationships in EBP$_i$ row for AAM.
$SR(BP_i)$: Summation of semantic relationships in BP$_i$ Column for $CRUDM$.

$SR(GOAL_i)$: Summation of semantic relationships in GOALi Column for AGM.

The business process of the case study is shown in Figure 5.

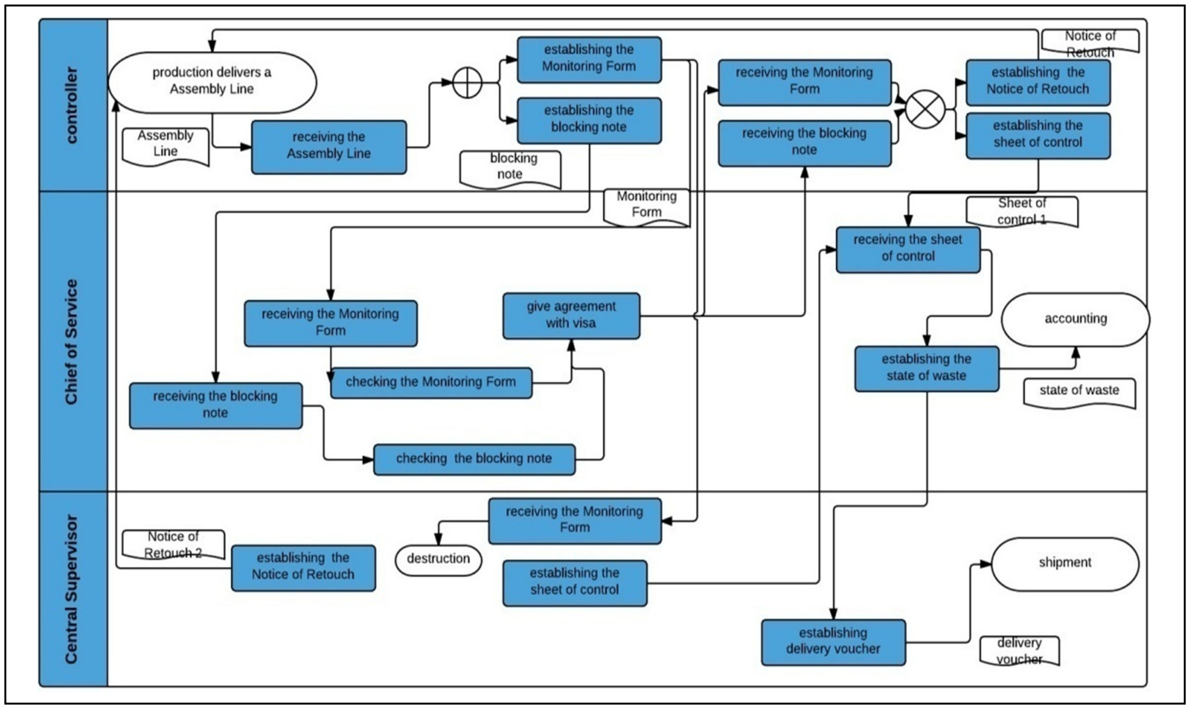

controller
Chief of Service
Central Supervisor
production delivers a Assembly Line
Assembly Line
receiving the Assembly Line
establishing the Monitoring Form
establishing the blocking note
blocking note
Monitoring Form
receiving the Monitoring Form
receiving the blocking note
Notice of Retouch
establishing the Notice of Retouch
establishing the sheet of control
Sheet of control 1
receiving the sheet of control
receiving the Monitoring Form
give agreement with visa
checking the Monitoring Form
receiving the blocking note
checking the blocking note
accounting
establishing the state of waste
state of waste
receiving the Monitoring Form
Notice of Retouch 2
establishing the Notice of Retouch
destruction
establishing the sheet of control
shipment
establishing delivery voucher
delivery voucher

Figure 5: the business process of the case study

We run our tool on the same business process used in [6], [8] and [15]. We note, as shown in Figure 6, that we have widely improved by reducing the number of iterations to achieve the same results as in [6], [8], and [15].

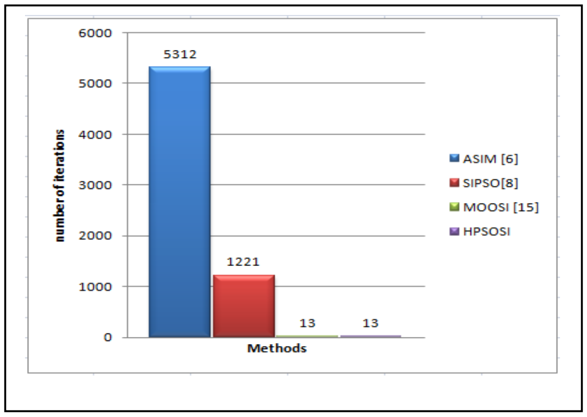


Figure 6: HPSOSI COMPARED TO ASIM [6], SIPSO [8], AND MOOSI [15]

	HPSOSI	ASIM [6]	SIPSO[8]	MOOSI [15]
Number of Activities	13	13	13	13
Input model	CRUD Matrix, Activity-Actor Matrix, Activity-Goal Matrix.	CRUD Matrix	CRUD Matrix	CRUD Matrix
Optimization algorithm	Hybrid Particle Swarm Optimization	simulated annealing	Combinatorial Particle Swarm Optimization	genetic algorithm
Number of Clusters found	3	4	4	5
Found at iteration	13	5312	1221	13

Table 2: HPSOSI COMPARED TO ASIM [6], SIPSO [8], AND MOOSI [15]

The difference found between the results is explained by the use in clustering step a Total Dependency Matrix (TDM) in our approach instead of using the CRUD matrix by approaches ASIM [6], SIPSO [8].

Although we have the same number of iterations in [15], a better solution is found with the highest fitness value thanks to the use of BEADM, AADM, and BGADM.

6. Conclusion and outlook

Matrix clustering for services identification is an NP-complete problem. This paper presents HPSOSI a multi-objective hybrid particle swarm optimization algorithm, where the crossover and mutation of a genetic algorithm is embedded into the PSO to improve the local search ability and to maintain the diversity of the population in order to achieve better performance than standard evolutionary algorithm. The proposed method identifies automatically candidate SOA services from business process models.

The experimental results show that our approach achieves high performance in term of convergence speed and execution time compared with other solution.

As future work, we are studying introducing others semantic annotation to fully consider semantic relationships between business elements, and hence transaction and semantic integrity can be guaranteed.

7. REFERENCES

[1] Arsanjani. Service-Oriented Modeling and Architecture (SOMA). IBM developer Works, 2004.

[2] Azevedo Leonardo Guerreiro, Flávia Santoro, Fernanda Baião, Jairo Souza, Kate Revoredo, Vinícios Pereira, and IsoldaHerlain. “A Method for Service Identification from Business Process Models in a SOA Approach“. CAiSE, pp. 99-112, 2009.

[3] Alejandro Cervantes. Inés Galván. Pedro Isasi.Binary particle swarm optimization in classification. Neural Network World 3(05), pp. 229-241,2005.

[4] Erl T. Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR, 2005, 792 p.

[5] Inaganti S. and Behara G. K .Service Identification: BPM and SOA Handshake. White Paper,2007.

[6] Jamshidi P. Mansour S. Sedighiani K. Jamshidi S. and Shams. “An Automated Service Identification Method“.Technical Report. Shahid Beheshti University.2012.

[7] Sheng-Jun Xue, Wu Wu. Scheduling Workflow in Cloud Computing Based on Hybrid Particle Swarm Algorithm. Nanjing University of Information Science & Technology, School of Computer and Software, Nanjing, China.TELKOMNIKA, Vol.10, No.7, November 2012, pp. 1560~1566.

[8] Chergui Mohamed El Amine, Sidi Mohamed Benslimane. Using Combinatorial Particle Swarm Optimization to Automatic Service Identification. 13th International Arab Conference on Information Technology ACIT’2013, December 17-19, 2013, Khartoum, Sudan.

[9] Kang DongSu, Chee-yang Song, Doo-Kwon Baik. “A Method of Service Identification for Product Line“. Proc. Of Third 2008 International Conference on Convergence and Hybrid Information Technology, pp. pp. 1040–1045, 2008.

[10] Kazemi Ali, Ali Rostampour, PooyanJamshidi, Eslam Nazemi, Fereidoon Shams, Ali Nasirzadeh Azizkandi. “A Genetic Algorithm Based Approach to Service Identification“.IEEE World Congress on Service Computing, Washington DC, USA, July 4-9, 2011

[11] Kennedy .J, R.C. Eberhart. “Particle swarm optimization“. Proceedings of the IEEE International Conference on Neural Networks, vol. IV, 1995, pp. 1942–1948.

[12] Martin, J. “ Information Engineering“, Prentice Hall, 1990.

[13] Magnus Erik Hvass Pedersen .Good Parameters or Particle Swarm Optimization. Hvass Laboratories Technical Report no. HL1001, 2010.

[14] Rana Yousef, Mohammad Odeh, David Coward, Ahmad Sharieh. “A Semantically Enriched Framework to Derive Software Service Oriented Models from Business Process Architectures“. International Journal of Information Studies, Volume 1, Issue 4 , October 2009.

[15] Soltani M. and Benslimane S. M. “From A High Level Business Process Model To Service Model Artifacts: A Model-Driven Approach“. 14-th International Conference on Enterprise Information Systems (ICEIS 2012), 28 June – 1 July, 2012, Wroclaw, Poland.

[16] Suntae Kim, Minseong Kim, VijayanSugumaran, Sooyong Park. “A Scenario Based Approach for Service Identification“.34th Annual IEEE Computer

Software and Applications Conference Workshops, p.237-238, 2010.

[17] Bhaskar MM, Benerji M, Sydulu M. A Hybrid Genetic Algorithm Approach for Optimal PowerFlow.TELKOMNIKA Indonesian Journal of Electrical Engineering. 2011; 9(2): 211-216.

[18] Bianchini, D.; Cappiello, C.; De Antonellis, V.; Pernici, B., "Service Identification in Inter-Organizational Process Design,

[19] " Services Computing, IEEE Transactions on , vol.PP, no.99, pp.102,2013.

[20] Birkmeier, Dominik Q.; Gehlert, Andreas; Overhage, Sven; Schlauderer, Sebastian, "Alignment of Business and IT Architectures in the German Federal Government: A Systematic Method to Identify Services from Business Processes," System Sciences (HICSS), 2013 46th Hawaii International Conference on , vol., no., pp.3848,3857, 7-10 Jan. 2013.

[21] Mancoridis S. et al, Using Automatic Clustering to Produce High-Level System Organization of Source Code, In Proceeding of the International Workshop on Program Understanding, June 24-26,1998, Ischia, Italy, pp 45–53,(1998).

[22] Qian Ma et al, Evaluating Service with Design Metrics On Business Process Decomposition, IEEE International Conference on Services Computing, pp. 160-167, (2009).

[23] S. Mancoridis, B. S. Mitchell, Y. Chen, and E. R. Gansner. Bunch: A clustering tool for the recovery and maintenance of software system structures. In Proceedings of the IEEE International Conference on Software Maintenance, pages 50 59. IEEE,1999.

[24] S. Mancoridis, B. S. Mitchell, C. Rorres, Y. Chen, and E. R. Gansner. Using automatic clustering to produce high-level system organizations of source code. In Proceedings of the Sixth International Workshop on Program Comprehension, pages 45 52. IEEE, 1998.