International Journal of Electrical and Computer Engineering (IJECE)
Vol. 4, No. 2, April 2014, pp. 285~294
ISSN: 2088-8708

Metric Suite to Evaluate Reusability of Software Product Line

Mohammad Ali Torkamani
R&D Department, Iranian Telecommunication Manufacturing Company, Shiraz, Iran

Article Info

Article history:

Received Nov 13, 2013
Revised Jan 19, 2014
Accepted Feb 4, 2014

Keyword:

Reusability
Software Product line
Software metrics

ABSTRACT

Metrics have long been used to measure and evaluate software products and processes. Software product line architecture is a field in which few metrics have been applied, a surprising fact given the important role of software product line architecture in software product line development. Recently, Some metrics have been developed to assess software product line architecture. These metrics are useful but have not been widely used in industry. In this paper, some new metrics are provided to assess reusability of Software product line architecture. Our metrics are evaluated in action.

Corresponding Author:

Mohammad Ali Torkamani
R&D Department, Iranian Telecommunication Manufacturing Company, Shiraz, Iran

1. INTRODUCTION

A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way [1]. Metrics are employed for estimating softwares and processes[2]. Available metrics in software engineering are insufficient and even are difficulty applied for estimating Product Line Architecture (PLA). PLA is a field with less defined metrics by which it is estimated. In recent years, some metrics have been introduced for estimating PLA. Although these metrics are very useful, they have not been however, widely employed in industries. For this, experts and R&D departments should pay more attention to the metrics employing in product lines. In this paper, we introduce some metrics for estimating reusability in software product lines. The rest of this paper is structured as follows. After explaining the related works in the second part, the metric suite for evaluating reusability in software product line is explained in the third part. Then in the fourth part, case study will be explained. The last part of paper has been allocated to conclusion.

2. RELATED WORKS

Most of initial work on software metrics focused on codemetrics which are derived solely from source code of aprogram, such as Lines of Code, Halstead's metrics andMcCabe's cyclomatic complexity. As the development ofobject-oriented technology, some object-oriented metrics havebeen proposed, such as CK metric, and MOOD metric.Some component metrics also are proposed to measurecomplexity, customizability, and reusability of components.Existing software metrics are inflexible and insufficient formeasuring PLA. PLA represent reference architecture ofproduct line members. Variability is basis for implementingparticularity of product line members, variability metrics areone most important part of PLA metrics. Variability also makesPLA more complex, and complexity metrics of PLA mustconsider issues of variability. PLA will be reused by productline members, reusability also should be assessed. So somenew metrics methods should be proposed to measure quality ofPLA[3]. Some references like [3-10] have proposed some metrics for measuring quality in software product lines. The most important of them are: Structure Similarity Coefficient (SSC), Component Reuse Rate (CRR), Reuse Benefit Rate (RBR), Product-related Reusability (PrR), Size of Commonality (SOC) and Percent Reuse (PR). Also, [11-16] have proposed metrics for assessing the feature model. In next section we will introduce a few metrics for estimating reusability in software product lines.

3. METRIC SUITE FOR EVALUATING REUSABILITY IN SOFTWARE PRODUCT LINES

The main object of a product line is reusability [3]. Various assets are being used in software product lines. These assets have different values. Also, the values of them differ from the value of the profit obtained by organization through employing reuse approach. Although the assets which are being used in product lines have different values, most available metrics like SOC and SSC [3-5] however, don't consider the weight values of these assets. In this paper we propose metrics which consider the weight values of assets.

3.1. Determining the Weight Value of Assets

In past years, the focus of experts was on the reusability of fine grain assets like reusability in code level. Due to this approach, we have seen fewer successes in reusability field. Currently, the concentrations have been changed towards coarse grain assets which are being uniformed by software architecture. This approach has some advantages: a) the assets would be more appropriate for offering in market, b) it increases productivity and c) it saves time. [17] Moreover, the SEI framework of product line [6] considers product line as an attempt for employing strategic plans for coarse grain reuse. For this, larger grain assets are more valuable for reusing in software product line. In order to determine the weight value of assets we should convert assets and artifacts to a common measurement unit such as "Line of code" [10]. If the number of code lines of the software assets is not available(like a situation in which an organization has purchased a commercial of the shelf (COTS)) or it is difficult to us to convert non software assets to the number of code lines, we can use an approach in order to determine the weight value of assets. Suppose that among different assets, the ak requires the minimum effort for developing. This minimum effort is shown by Ek. Now, we can calculate the weight value of the ai asset through equation (1):

$$W_i = \frac{E_i}{E_k} \quad (1)$$

It is clear that the weight value of the asset ak will be equal to one. The higher levels of effort required for developing an asset will have more costs. For this, in the equation (1) we can replace effort level by development cost. Then, we have:

$$W_i = \frac{Cost_i}{Cost_k} \quad (2)$$

3.2. Weight Percent of Reusability

We can improve the SSC metric by applying weight values. As our metric differs from SCC formula, we call it weight percent of reusability. According to equation (3), weight percent of reusability is: (the sum of common components of PLA /the sum of all components of product line)*100:

$$Wt\%R = \frac{\sum_{i=1}^{k} W_i}{\sum_{j=1}^{n} A_j} \times 100 \quad (3)$$

In which k is the number of common components of PLA, n is the total number of components of product line, W_i is the weight value of the ith common component and A_j is the weight value of the jth component. According to this formula, the higher weight values of common components of PLA will lead to the higher architectural similarity of the members of product line which in turn will lead to higher rates of profit obtaining through employing reusability approach. If we show the weight value of product line assets as Wspl, we can rewrite the equation (3) as follows:

$$Wt\%R = \frac{1}{w_{SPL}} \sum_{j=1}^{N} w_j \times 100 \quad (4)$$

Also, we can calculate weight percent of reusability for product line products through the following equation:

$$Wt\%R_P = \frac{1}{w_p} \sum_{j=1}^{N} w_j \times 100 \quad (5)$$

In which $Wt\%R_P$ is the weight percent of reusability of the product P and wj is the weight value of the jth component that reused in the product p. Wp is the weight value of the assets of product P which is derived through the following equation:

$$w_p = \sum_{k=1}^{m} w_k \quad (6)$$

In which m is the number of the assets of the product P.

3.3. Average of Rehabilitation

If Ci be the average of rehabilitation of the ith component in software product line, the average of rehabilitation of whole assets in software product line (AoRspl) would be derived from the following equation:

$$AoR_{SPL} = \frac{1}{n}\sum_{i=1}^{k} C_i \quad (7)$$

In which k is the number of reused components in the common part of PLA and n is the number of whole components in the common part of PLA. The value of $C_i$would be one if the ith component be used as Black Box. For other reusability methods like Whit Box approach, the value of Ci is obtained though the following equation:

$$C_i = 1 - \frac{Mi}{100} \quad (8)$$

In which M_i is the percent of changes applying on each component for adoption and reusability purposes.

Similarly, we can calculate the average of rehabilitation of a given product through the following equation:

$$AoR_p = \frac{1}{n}\sum_{i=1}^{k} Cp_i \quad (9)$$

In which AoR_p is the average of rehabilitation of the product p and Cp_i is the average of rehabilitation of the ith component in the product p. The value of Cp_i is calculated similar to C_i i.e. through the equation 8. In the equation (9), k is the number of the reused components in the product p and n is the total number of components in the product p. If we wish to express the average of rehabilitee in percent, it is just enough to multiply the derived numbers from the equations (7) and (9) by 100.

Example: imagine five components as C1 to C5 which have been reused in a software product line. Table 1 shows the percent of changes of these components. In this table, WB stands for White Box and BB stands for Black Box.

Table 1. The percent of change for adopting with new architecture

Row	Name	type	percent of change for adoption (M_i)	Ci
1	C1	WB	20%	0.8
2	C2	BB	0%	1
3	C3	WB	50%	0.5
4	C4	WB	35%	0.65
5	C5	WB	70%	0.3

The average of rehabilitation of software product line is expressed as follows:

$$AoR_{SPL} = \frac{1}{5}(\,0.8 + 1 + 0.5 + 0.65 + 0.3) = 0.65$$

It could be said that the average of rehabilitation for adopting with software product line is 65%.

3.4. Introducing Some Metrics for Estimating Reusability Based On the Mapping of Software Product Line as Graph

Recently, some of researchers like Mr. Burger[5] have employed the theory of sets and graph for modeling software product line and displaying the relationships of the products of product line. In this section, we introduce some metrics for software reusability. These metrics have been obtained through mapping software product line to a graph, that we call it Product-Asset graph (see fig. 1). Assume that A is the set of the assets of our product line:

$A = \{a1, a2, \dots, a_{|A|}\}$

In this set, the number of the members of the set A is shown as $|A|$. Also, assume that P is the set of the products of the software product line:

$P = \{p1, p2, \dots, p_{|P|}\}$

Again in this set, the number of the members of the set P is shown as $|P|$. Each asset can be used in every product. Assume that the percent of changes applying for adopting reusability in different products differs from asset to asset. This implies that the profit obtaining through the reusing of assets in products would be different. Assume that B_{ij} is the benefit obtaining through the reusing of asset a_i in product P_j. We define the weighted and directed graph G as follows:

G= (V, E)

$V = P \bigcup A$

$E = P \times A \times B_1$

$B_1 = \{\forall B_{ij} : i \in P, j \in A, B_{ij} = CD_{ij} - CR_{ij}\}$

$B_{ij}, CD_{ij}, CR_{ij} \in \mathbb{R}$

B_{ij} is the benefit obtaining through the reusing of asset a_j in product p_i. CD_{ij} is the cost of developing asset aj in the product p_i. CR_{ij} is the cost of reusing the asset aj in the product p_i. This graph includes the couple of edges like $e_{1,2} = (p1, a2, B12) \in E$.

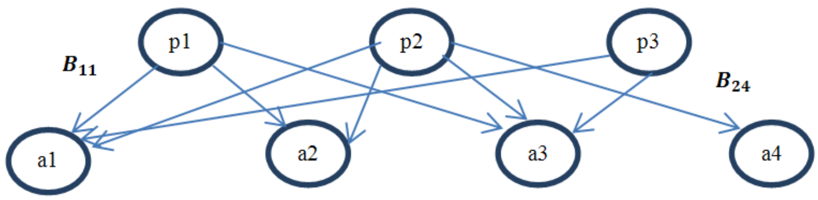

Figure 1. Product-asset graph

We will introduce some metrics based on this graph in next.

3.4.1. Calculating the Benefit Obtaining Through Reusability in Product Line

The total benefit of reusability is obtained through the following equation:

$$B = \sum_{i=1}^{|P|} \sum_{j=1}^{|A|} B_{ij} \quad (10)$$

According to this equation the total benefit obtaining through the reusing of assets in software product line is equal with the benefit obtaining through the reusing of assets in all individual products of product line.

3.4.2. The Impact of the Reusability of an Asset on Developing a Given Product

The impact of the reusability of the asset aj on the developing of the product pi is derived through the following equation:

$$I_{ij} = \frac{w(a_j) * [1 - \frac{k_{ij}}{100}]}{w(p_i)} \quad (11)$$

In which Iij is the impact of the reusability of the asset aj in developing the product pi, w(aj) is the weight value of the asset aj, kij is the percent of changes of the asset aj applying for using in the product pi and w(pi) is the weight value of total assets used in the product pi. If we wish to express the weight value of assets in terms of line of code, we can rewrite the equation (11) as follows:

$$I_{ij} = \frac{size(a_j) - [size(a_j) * k_{ij}]}{size(p_i)} \quad (12)$$

In which Iij is the impact of the reusability of the asset aj in developing the product pi, size (aj) is the number of the lines of the asset aj, kij is the percent of changes of the asset aj applying for using in the product and size(pi) is the number of the lines of the product pi.

Example: suppose that $size(p_1) = 2KLOC$ و $size(a_1) = 400LOC$ ، $k11 = 20\%$. From the equation (11) we have:

$$I_{11} = \frac{400 * [1 - \frac{20}{100}]}{2000} = 0.16$$

Also, we can calculate I11 through the equation (12) as follows:

$$I_{11} = \frac{400 - [400 * 0.2]}{2000} = 0.16$$

3.4.3. The Impact of Reusability on Developing a Given Product

The impact of the reusability of assets on developing the product pi is derived through the following equation:

$$I_i = \sum_{j=1}^{|A|} I_{ij} \quad (13)$$

In which Ii is the impact of reusability on the product p_i.

3.4.4. The Impact of Reusability on Developing All Products of Product Line

The impact of reusability on developing all products of software product line is calculated through the following equation:

$$I = \sum_{i=1}^{|P|} I_i \quad (14)$$

In which Ii is the impact of reusability on the product pi. We can rewrite the above equation as follows:

$$I = \sum_{i=1}^{|P|} \sum_{j=1}^{|A|} I_{ij} \quad (15)$$

In which Iij is the impact of the reusability of the asset aj on the developing of the product pi. Impact of Reusability measures reuse benefit of software product line. Normally software product line has more members, this metric is bigger, and product line is more economic.

4. CASE STUDY

In this section, Our Metric suite is evaluated in practice in Iranian Telecommunication Manufacturing Company (ITMC). ITMC is a company operating in Electrical engineering and ICT areas. Beside some products in electrical and communication area, ITMC is developing some software systems. In order to take advantage of Software Product Line, R&D department of ITMC has developed five Software product lines:

- SPL1: Software Product Line for Mobile Sets
- SPL2: software product line for Telecommunication Centers
- SPL3: Software Product line for ECU (and Smart control systems for cars)
- SPL4: Software Product line for ATM and Banking systems
- SPL5: ERP Software Product line

The evaluation indexes employing in this case study are SCC (Structural Similarity Coefficient), RBR (Reuse Benefit Rate) PrR (Product-related Reusability) and SOC (Size of Commonality). Tables 2 to 7 show the data belonging to the product line 1. Table 2 shows the list of the common assets of the Software product line architecture 1. Table 3 shows the list of assets reused in some products of the software product line 1. Table 4 shows the list of other new-developed assets. Table 5 shows the list of the products of the software product line 1 along with the information of every product including product name, the name of assets reused in products, the weight value of each asset, asset type (developed or reused) and percent of changes applied for adopting with new architecture and product weight (Wp). Also in these table, the calculations of metrics Wt%Rp and AoRP, the impact of the reusability of an asset on product development (Iij), the impact of reusability on developing a given product (Ii), the impact of reusability on all products and product related reusability(PrR). At first we calculated evaluation indexes and the metrics introduced in this paper. The obtained results were saved in table 6. Table 7 defines the ranks of product lines 1 to 5 for each comparison aspect. This table has been prepared using the results saved in the table 6.

Table 2. The list of common assets of the architecture of the product line 1

no	asset	wi	type	Percept of Changes
1	a1	1	Reuse	20
2	a2	2	Reuse	BB
3	a3	2	Reuse	BB
4	a4	1	Reuse	BB
5	a5	7	Reuse	40
6	a6	3	Reuse	40
7	a7	5	Develop.	
8	a8	2	Develop.	

Table 3. The list of assets reused in some products of the product line 1

no	**asset**	**wi**
1	a9	2
2	a10	1
3	a11	1
4	a12	2
5	a13	1
6	a14	1
7	a15	1
8	a16	2
9	a17	2
10	a18	2
11	a19	2
12	a20	2

Table 4. The list of new developed assets of the product line 1

no	**asset**	**wi**	**row**	**Asset**	**wi**	**row**	**asset**	**wi**	**row**	**asset**	**wi**
1	a21	1	11	a31	1	21	a41	1	31	a51	1
2	a22	1	12	a32	1	22	a42	1	32	a52	1
3	a23	1	13	a33	1	23	a43	1	33	a53	1
4	a24	1	14	a34	1	24	a44	1	34	a54	1
5	a25	1	15	a35	1	25	a45	1	35	a55	1
6	a26	1	16	a36	1	26	a46	1	36	a56	1
7	a27	1	17	a37	2	27	a47	1	37	a57	1
8	a28	1	18	a38	5	28	a48	1	38	a58	1
9	a29	3	19	a39	1	29	a49	1	39	a59	1
10	a30	1	20	a40	1	30	a50	1	40	a60	1

If you compare rows 9 and 13 in the table 7, you will find that the metric of "the impact of reusability on products" is completely in accordance with the metric of RBR. Our metric has an advantage compared with the metric of RBR. It can be calculated for each product and is not general like RBR. Comparison of the rows 10 and 14 of the table 7 reveals that the results of the metric of Wt%Rpare completely similar to the average obtained through PrR metric. Also our Wt%Rp metric gives two different values to the product lines 2 and 3 while their values in PrR metric are the same. The results of this case study show that the product line 2 has the maximum weight value in reusability as it has gained the maximum value in Wt%Rpmetric. In other words, in this product line the ratio of the weights of the components of common part of software product line architecture to the weight of whole components of the software product line gains the maximum value. For this reason, it is expected that the product line 2 will be more successful from the viewpoint of reusability. (For example compare weight value with the number of lines of the program or required effort for developing purposes.)

In order to check the accuracy of the metric of AoRP, compare two different products with each other (for example products p1 and p2 belonging to the product line 1)

As you may see in the table 5, the most components of the product p1 are Black Box type. Generally the components of the product p1 require fewer changes for adopting with the architecture of the product line compared with the product p2. The metric of AoRP shows this fact clearly. (The value of this metric is 0.854545 for the product p1 and 0.745445 for the product p2.)

This metric will work for other components too even if you select items from different product lines. The metric of AoRSPL is similar to AoRP. The only difference is that this metric estimates only the common components of the architecture of product line. According to the table 7, among various product lines, the product line 2 has the maximum AoRSPL. This means that the common components of the architecture of the product line 2 require fewer changes for adopting with new architecture compared with other product lines.

Table 5. The products of the product line 1

Product	asset	wi	Type R: Reuse D:Develpment	Percent of Changes	$1-\frac{M_i}{100}$	Wp	$\sum_{i=1}^{n} w_i$	Wt%R_p	AoR_P	I_{ij}	I_i	I	PrR
p1	a1	1	R	20	0.8	29	22	75.86206897	0.854545455	0.027586207	0.593103448	2.143200067	0.615384615
	a2	2	R	BB	1					0.068965517			
	a3	2	R	BB	1					0.068965517			
	a4	1	R	BB	1					0.034482759			
	a5	7	R	40	0.6					0.144827586			
	a6	3	R	40	0.6					0.062068966			
	a7	5	D										
	a8	2	D										
	a11	1	R	20	0.8					0.027586207			
	a12	2	R	BB	1					0.068965517			
	a13	1	R	BB	1					0.034482759			
	a14	1	R	BB	1					0.034482759			
	a15	1	R	40	0.6					0.020689655			
p2	a1	1	R	20	0.8	45	38	84.44444444	0.745454545	0.017777778	0.404444444		0.347826087
	a2	2	R	BB	1					0.044444444			
	a3	2	R	BB	1					0.044444444			
	a4	1	R	BB	1					0.022222222			
	a5	7	R	40	0.6					0.093333333			
	a6	3	R	40	0.6					0.04			
	a7	5	D										
	a8	2	D										
	a16	2	R	10	0.9					0.04			
	a17	2	R	60	0.4					0.017777778			
	a18	2	R	25	0.75					0.033333333			
	a19	2	R	40	0.6					0.026666667			
	a20	2	R	45	0.55					0.024444444			
	a21	1	D										
	a22	1	D										
	a23	1	D										
	a24	1	D										
	a25	1	D										
	a26	1	D										
	a27	1	D										
	a28	1	D										
	a29	3	D										
	a30	1	D										
p3	a1	1	R	20	0.8	30	18	60	0.95	0.026666667	0.45		0.571428571
	a2	2	R	BB	1					0.066666667			
	a3	2	R	BB	1					0.066666667			
	a4	1	R	BB	1					0.033333333			
	a5	7	R	40	0.6					0.14			
	a6	3	R	40	0.6					0.06			
	a7	5	D										
	a8	2	D										
	a9	2	R	15	0.85					0.056666667			
	a31	1	D										
	a32	1	D										
	a33	1	D										
	a34	1	D										
	a35	1	D										

p4	a1	1	R	20	0.8	38	21	55.26315789	0.78	0.021052632	0.4	0.470588235
	a2	2	R	BB	1					0.052631579		
	a3	2	R	BB	1					0.052631579		
	a4	1	R	BB	1					0.026315789		
	a5	7	R	40	0.6					0.110526316		
	a6	3	R	40	0.6					0.047368421		
	a7	5	D									
	a8	2	D									
	a9	2	R	40	0.6					0.031578947		
	a10	1	R	40	0.6					0.015789474		
	a11	1	R	30	0.7					0.018421053		
	a12	1	R	10	0.9					0.023684211		
	a36	1	D									
	a37	2	D									
	a38	5	D									
	a39	1	D									
	a40	1	D									
p5	a1	1	R	20	0.8	46	19	41.30434783	0.775	0.017391304	0.295652174	0.266666667
	a2	2	R	BB	1					0.043478261		
	a3	2	R	BB	1					0.043478261		
	a4	1	R	BB	1					0.02173913		
	a5	7	R	40	0.6					0.091304348		
	a6	3	R	40	0.6					0.039130435		
	a7	5	D									
	a8	2	D									
	a10	1	R	40	0.6					0.013043478		
	a17	2	R	40	0.6					0.026086957		
	a41	1	D									
	a42	1	D									
	a43	1	D									
	a44	1	D									
	a45	1	D									
	a46	1	D									
	a47	1	D									
	a48	1	D									
	a49	1	D									
	a50	1	D									
	a51	1	D									
	a52	1	D									
	a53	1	D									
	a54	1	D									
	a55	1	D									
	a56	1	D									
	a57	1	D									
	a58	1	D									
	a59	1	D									
	a60	1	D									

Table 6. Calculation of metrics and comparison of the results obtained for product line 1 to 5

no	Comparison aspect	SPL1	SPL2	SPL3	SPL4	SPL5
1	Total number of components	60	40	40	44	16
2	The number of components used in common architecture section	6	10	10	15	5
3	number of products	5	6	6	10	5
4	SOC	8	10	10	15	5
5	SSC	0.133333333	0.25	0.25	0.340909	0.3125
6	Wspl	89	72	96	105	44
7	The weight of the members of product line	188	206	233	725	104
8	The weight of the components of common Part of architecture	23	31	29	61	13
9	RBR	2.112359551	2.861111	2.427083	6.904762	2.363636
10	The average of PrR	0.454378835	0.65702	0.65702	0.727546	0.660714
11	The average of AoR_P	0.821	0.706401	0.841657	0.744731	0.658333
12	AoR_{SPL}	0.625	0.875	0.725	0.716667	0.66
13	I	2.143200067	3.788222	3.566379	7.007818	2.941345
14	The average of the weight percent of reusability of products (Wt%Rp)	63.37480383	82.67909	76.1477	96.50385	83.95562
15	Wt%R	12.23404255	15.04854	12.44635	8.413793	12.5

Table 7. The ranks of different aspects of the product lines 1 to 5

No	Comparison aspect	Rank1	Rank2	Rank3	Rank4	Rank5
1	Total number of components	1	4	2,3	5	
2	The number of components used in common architecture section	4	2,3	1	5	
3	number of products	4	2,3	1,5		
4	SOC	4	2,3	1	5	
5	SSC	4	5	2,3	1	
6	Wspl	4	3	1	2	5
7	The weight of the members of product line	4	3	2	1	5
8	The weight of the components of common Part of architecture	4	2	3	1	5
9	RBR	4	2	3	5	1
10	The average of PrR	4	5	2,3	1	
11	The average of AoR_P	3	1	4	2	5
12	AoR_{SPL}	2	3	4	5	1
13	I	4	2	3	5	1
14	The average of the weight percent of reusability of products (Wt%Rp)	4	5	2	3	1
15	Wt%R	2	5	3	1	4

5. CONCLUSION

We argued that the most of available metrics employing for estimating product line architecture are insufficient and also employing these metrics are difficult. Product line architecture is a field with fewer metrics. In recent years some new metrics have been proposed for estimating product line architecture. Although the proposed metrics are useful they have not been widely used in industries. For this, experts and R&D departments should pay more attention to the metrics employing in product line architecture.

In software product line, various types of assets are being used. The values of these assets are different and also the profit obtaining through using these assets is different. Despite of this fact, most available metrics don’t consider the weight values of the assets of software product line. We proposed in our paper some new metrics for estimating reusability in software product line. These metrics consider the weight values of assets.

Our Metric suite is evaluated in practice in Iranian Telecommunication Manufacturing Company. Along with other metrics, our proposed metrics can help us to estimate the quality of software product line.

REFERENCES

[1] Bass L, Clements P and Kazman R. *Software Architecture in Practice*. Second Edition, Addison Wesley. 2003.
[2] van der Hoek, A Dincel E and Medvidoviü N. “*Using Service Utilization Metricsto Assess the Structure of Product Line Architectures*”. Proceedings of the Ninth International Software Metrics Symposium (METRICS’03), IEEE. 2003.
[3] Zhang T, Deng L, Wu J, Zhou Q and Ma C. “Some Metrics for Accessing Quality of Product LineArchitecture”. *IEEE*. Vol 27, Issue 3, pp 35 – 41, 2008.
[4] Berger C And Busse C. “Product Line Metrics for Legacy Software in Practice”. The 14th International Software Product Line. 2010.
[5] Berger C, Rendel H and Rumpe B. “*Measuring the Ability to Form a Product Line from Existing Products*”. Variability Modelling of Software-intensive Systems (VaMos). 2010.
[6] SEI, A Framework for Software Product Line Practice, Version 5.0, 2009, Available: http://www.sei.cmu.edu/productlines/frame_report/miningEAs.htm
[7] Aldekoa G, Trujillo S, Sagardui G and Díaz O. EXPERIENCE MEASURING MAINTAINABILITYIN SOFTWARE PRODUCT LINES. *XV Jornadas de Ingeniería del Software y Bases de Datos (JISBD)*. 2006.
[8] Rahman A. Metrics for the Structural Assessment of Product Line Architecture. MSC Thesis, School of Engineering at Blekinge Institu te of Technology, Sweden. 2004.
[9] Junior E, Gimenes I and Maldonado J. “A Metric Suite to Support Software Product Line Architecture Evaluation”. *XXXIV ConferenciaLatinoamericana de Informática (CLEI 2008)*. 2008.
[10] Zubrow D and Chastek G. Measures for Software Product Lines. Technical Report, Carnegie Mellon University. 2003.
[11] Benavides D, Trinidad P and Ruiz-cortés A. “Automated Reasoning onFeature Models”. LNCS, Advanced Information Systems Engineering: 17thInternational Conference, CAISE. 2005.
[12] Mendonca M, Wasowski A, Czarnecki K and Cowan D. “*Efficient compilation techniques for large scale feature models*”. In Generative Programming and Component Engineering, 7th International Conference,GPCE , Proceedings. pages 13–22, 2008.
[13] Fernandez-Amoros D, Gil R and Somolinos J. “Inferring Information from Feature Diagrams to Product Line Economic Models”. ACM International Conference Proceeding Series Vol. 446, Proceedings of the 13th Internationsl Software Product Line Conference, pp. 41-50, 2009.

[14] Czarnecki K and Kim P. "Cardinality-based feature modeling and constraints: A progress report". In Proceedings of the International Workshop on Software Factories At OOPSLA 2005. 2005.
[15] Kasikci B and Bilgen S. "Scalable modeling of software product line variability". 13th International Software Product Line Conference, SPLC 2009, SanFrancisco. August 2009.
[16] Benavides D, Segura S and Ruiz-Cortés A. "Automated analysis of feature models 20 years later: A literature review". *Information Systems.* 2010.
[17] Jones LG. Product Line Acquisition in the DoD: The Promise, The Challenges, Technical Note, Carnegie Mellon University,1999.

BIOGRAPHY OF AUTHOR

Mohammad Ali Torkamaniwas born in Iran, Shiraz City, in 1975. He received the M.S. degree in software engineering from the Shahid Beheshti University, in 2011. He is the author of 15 books (in Persian), more than 35 articles. His research interests include software architecture, Ultra Large Scale systems, cryptography and Network security and holds one patent.
He is working in R&D Department of Iranian Telecommunication Manufacturing Company now. Also, he is currently teaching at the University of Applied Science and Technology in Shiraz.