Efficiency of Graph for the Remodularization of Multi-Level Software Architectures

Lala Madiha HAKIK

madiha.hakik@isga.ma

***Higher Institute of Engineering and Business ISGA*, Casablanca, Morocco**

Abstract

In a previous study we proceeded to the remodularization architecture based on classes and packages using the Formal Concept Analysis (FCA)[13] [14] [30]. we then got two possible remodularized architectures and we explored the issue of redistributing classes of a package to other packages, we used an approach based on Oriented Graph to determine the packages that receive the redistributed classes and we evaluated the quality of a remodularized software architecture by metrics [31] [28] [29]. In this paper, we will address the issue of the efficiency of the Oriented Graph in the remodularization of software architectures compared to the Formal Concept Analysis FCA method. The formal method of FCA concept is not popularized among scientists as opposed to the use of the labeled directed graph. It is for this reason that our directed graph approach is more effective in its simplicity and popularity.

Keywords:

Software modularization, Formal Concept Analysis (FCA), Oriented graph, Metrics, Coupling and cohesion.

1. Introduction

Great software systems based on approaches, the object consist of classes grouped into packages, forming a modular structure. The dependency relationships between classes in the same package (internal dependencies), and between classes of different packages (external dependencies generate complexity making it difficult to understand and maintain the system. In addition, the modular structure tends to degrade over time, making necessary an expert intervention for modernization. Many researchers make proposals on this subject using technical visualization, algorithms of remodularization or Exploring the Redistribution Classes of a Package with an Approach Based on Formal Concept Analysis.[13][14] [31]. In this paper, we study a particular declination, cf. the problem presented by H. Abdeen et al. [2] [1] [31], which is about the redistribution of classes from one system to existing packages. Namely, we consider in this paper more precisely the redistribution of classes in a package to other packages. This package may be a very small and in fact we want to balance the sizes of packages in the system, or it was artificially created to contain added classes to the system and the designer considers that there is no consistency semantics. In a context of restructuring architecture or development, it can also be linked to the disappearance of a node in a distributed architecture or recentralization and / or refactoring for maintenance reasons. We have explored a solution using Oriented Graph based on the technique of shortest path and illustrated our proposal with a theoretical example [31]. Section II presents our example, then we describe the approach in Section III. Section IV presents validation metrics of cohesion and coupling measure and we discuss our main results. Related work is presented in Section V, and then we conclude in Section V I.

2. Illustration

This section presents the problem of software architecture remodularization on an example. We will use the architecture shown in Figure 1 consists of five packages A,
B, C, D and E. Packages A, B, C, D, E are expected c contain more classes that are not shown for simplicity. Dependencies linking classes: they correspond for Example to call a method or use of a type. External dependency relationships link classes of package E to classes of other packages.
Internal dependency relationships connect classes E between. Internal dependencies of A, B, C and D are not presented[13] [14].

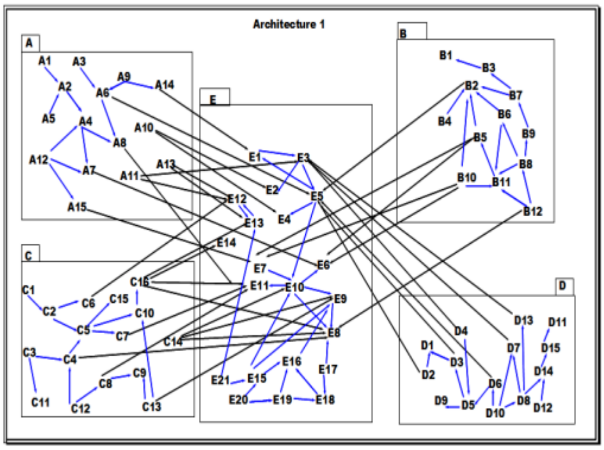

Fig. 1 An initial architecture composed of classes and packages

We are interested in the redistribution of classes E to

Manuscript received May 5, 2024
Manuscript revised May 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.5.4

Other packages with an exploratory method, whose proposals for redistribution are then presented to an expert. These proposals are based on the idea that the expert, while checking the semantic classes, could search for the increase of the cohesion (within the meaning of the coupling of classes in a package) and reduce the coupling between classes in different packages. To do this, we believe it is appropriate to encourage the following two trends:

1. Classes in a package attract them to classes of E,
2. If classes of E are interconnected, it is better to redistribute in the same package.

We believe that the Oriented Graph based on the technique of shortest path can bring interesting ways to solve this problem because this technical method allows the group to connect classes identically. We are looking here to propose a solution to an expert.

3. Proposed approach

In our approach, we are inspired of the notion of graph to present the original architecture of Figure 1 as nodes relative to classes and arcs relative to the relationship between these classes. Figure 2 illustrates this vision.

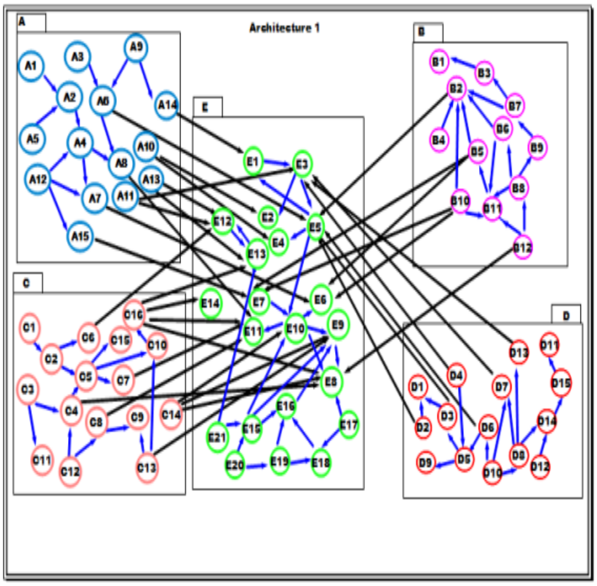

Fig. 2 Original architecture of Figure 1 as nodes relative to classes and arcs relative to the relationship between these classes

3.1 Formalization

In a second step we focus on the classes of package E, to be deleted related with classes of other packages considered in this case as nodes shown in figure 3.

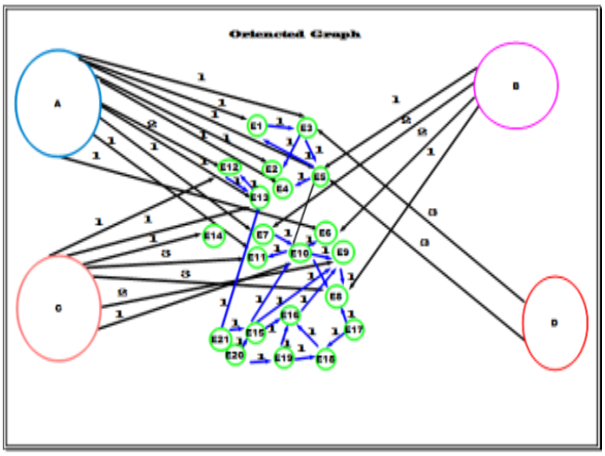

Fig. 3 Oriented Graph result of figure 2

- The relationship between classes and packages are represented by edges connecting each pair of nodes as an example the nodes A and E1 are connecting by the edge (A,E1) image of couple (**Package, Class**).

It is found that all the conditions are met to define a graph oriented, object of Figure 3.

Definition 1 (Oriented Graph) [15]:
A graph G is a mathematical structure defined by a pair (N, E) where N is a set of objects called nodes or vertices and E part of N * N which represents a set of arcs (also called edges) each connecting a pair of nodes. This general definition is a directed graph distinguishes two vertices s1 and s2 the edge (s1, s2) of the edge (s2, s1). The number of connections available to each class of package E with classes of packages A, B, C and D and mentioned on the arcs.

Example of procedure:
For the choice of the allocation of classes E Package to one of the other packages A, B, C and D, we adopted an approach advocating the use of directed graph and the technique based on the definition of the shortest path.

For examples:

1. In Figure 2, the class E1 of package E has an external connection with class A14 of package A, therefore the corresponding arc in figure 3 between class E1 and package A is the number 1. where the idea of cost of a shortest path [1]. By applying this principle class E1 of package E is affected into package A.
2. The class E12 of package E has three external relations, both with package A (one with class A13 of package A, the other with the class A11 in the same package) and the third class C16 of package C (Figure3). Under the definition of the shortest path the class E12 go to Package C.

Special case:

3. The class E8 of package E has three internal relations with the classes E9, E10 and E17 of package E and two external relations, one with class B12 of package B and the other with class C4 of package C, since the classes E9 and E10 will be affected by the principle of shortest path, to the package C therefore E8 will go also to the package C dominant.

Thus all the classes in the package E are redistributed according to the methodology listed above, and thereby the package E has been deleted to arrive on remodularized architecture (figure 4) .

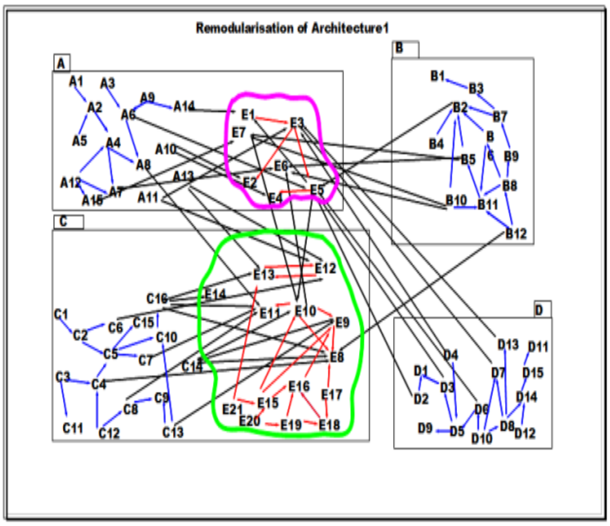

Fig. 4 One possibility of remodularization

6. Results and discussion

6.1 Validation metrics

For validatiton of metrics cohesion and coupling, our calculations were based on figures 1 and 4 with an architecture comprising 5 packages A, B, C, D and E by redistribution classes of package E (figure 1) using oriented graph and the technique based on the definition of the shortest path, which resulted one possible remodularized architecture (figure4). The package E is removed during this operation.

6.2 Cohesion metrics

Table 1. Cohesion metrics: Index of Package Goal Focus and Index of Package Services Cohesion.

	PF	IPSC
Package E of the original architecture 1	0.5	0.0116
Package A of the original architecture 1	0	1
Package C of the original architecture 1	0	1
Package A of the remodularization 1	0.25	1
Package C of the remodularization 1	0.46	1

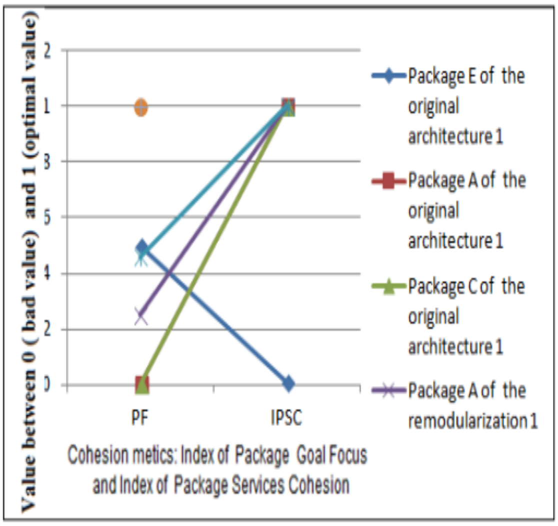

Fig. 5. Graphic representation of Cohesion metrics: Index of Package Goal Focus PF and Index of Package Services Cohesion IPSC (table1).

The Cohesion metrics: Index of Package Goal Focus (PF) and Index of Package Services Cohesion (IPSC) take their values from 0 to 1, where 1 is the optimal value and 0 is the wrong value.

Figure 5 gives the values of indices PF and IPSC for:
- Package E of Original Architecture 1 whose indexes are bad values because they are lower than 1.
- Packages A and C of remodularizated architecture whose the index IPSC is optimal value 1 therefore very good.

6.2 Coupling metrics

Table 2. Coupling metrics: Index of Inter-Package Interaction (IIPU and IIPE)

	IIPU	IIPE
The original architecture 1	0.588	0.333
Architecture of the remodularization 1	0.811	1

The coupling metrics: Index of Inter-Package Interaction (IIPU and IIPE) object of the figure 6, it is observed an improvement indexes IIPU and IIPE at remodularization 1 architecture compared to indexes of the original architecture 1 therefore a trend to optimality.

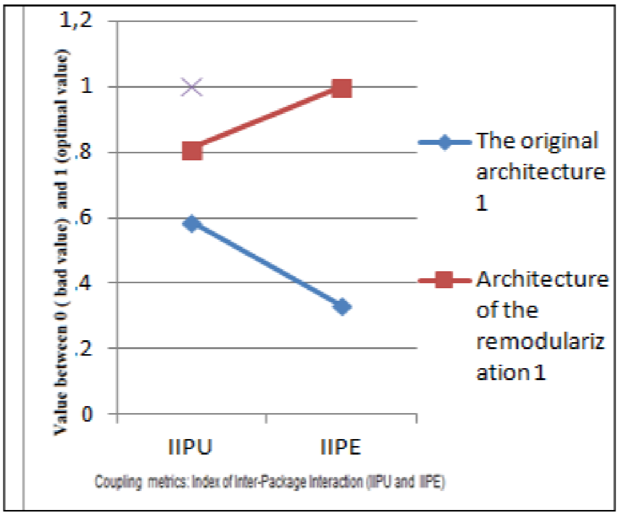

Fig. 6 Graphic representation of Coupling metrics: Index of Inter-Package IIPU and IIPE (table2).

Table 3. Coupling metrics: Index of Package changing Impact IPCI; Index of Package Communications Diversion (IIPUD and IIPED).

	IPCI	*IIPUD*	*IIPED*
Package E of the original architecture 1	0	0.271	1
Package A of the original architecture 1	1	1	1
Package B of the original architecture 1	1	1	1
Package C of the original architecture 1	1	1	1
Package D of the original architecture 1	1	1	1
The original architecture 1	0.8	0.854	1
Package A of the remodularization 1	0	0.38	0.38
Package B of the remodularization 1	1	0.583	0.583
Package C of the remodularization 1	1	0.541	0.541
Package D of the remodularization 1	1	1	1
Remodularization 1	0.75	0.626	0.626

Concerning the coupling metrics: Index of Package changing Impact (IPCI) and Index of Package Communications Diversion (IIPUD and IIPED) presented in figure 7, the results obtained for remodularization 1 approximate from those of the original architecture 1 extend to a higher interesting value 0.626

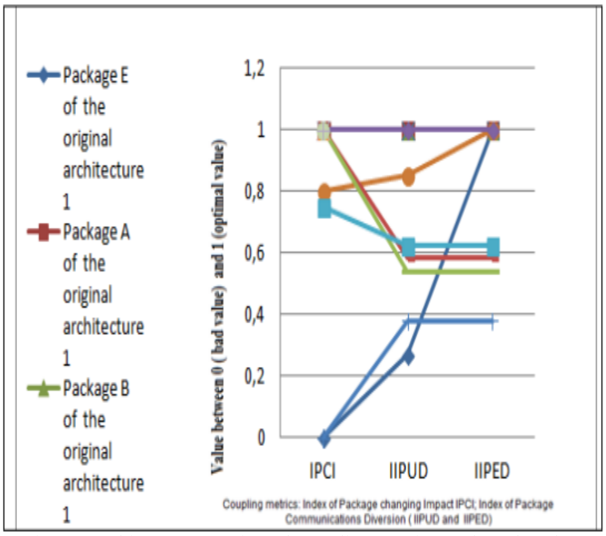

Fig. 7 Graphic representation of Coupling metrics: Index of Package changing Impact; Index of Package Communications Diversion (table3).

The results obtained at the level of the cohesion for the remodularized architecture 1 provides an optimum value 1 compared to the original architecture 1. The results of the coupling have an improvement at the level of remodularized architecture 1 compared to the original architecture 1.

The results of the cohesion and coupling metrics were conclusive for both approaches, namely:

- Remodularization of architecture 1 by the FCA formal concept method [30] [13] [14].

- Remodularization of architecture 1 by this method based on the directed graph [28] [31].

The formal method of FCA concept is not popularized among scientists as opposed to the use of the labeled directed graph.

It is for this reason that our directed graph approach is more effective in its simplicity and popularity.

The directed graph approach easily leads to the desired result because it is easily understood by a large scientific majority as opposed to the FCA method [30] known by a scientific minority results that also conclude that.

7. Related work

Different automated approaches have been proposed to restructure object systems. We cite three: the clustering algorithms, algorithms based on meta -heuristics and those based on the FCA[6]. The first aim to restructure system by the distribution of some elements (eg classes, methods, attributes) in groups such that the elements of a group are more similar to each other with elements of other groups [3] [7] [5]. Approaches to restructuring based on meta_heuristic algorithms [9] [8] are generally iterative stochastic algorithms, progressing towards a global optimum of a function by evaluating a certain objective function (eg characteristics or quality metrics). Finally, the approaches based on FCA [10] [12] provide an algebraic derivation of hierarchies of abstractions from all entities of a system. Reference [4] presents a general approach for the application of the FCA in the field of object-oriented software reengineering. In previous work, we added the dimension of exploration using the FCA[13] [14].

In this paper we explore the issue of redistributing classes of a package to other packages. We use an approach based on Oriented Graph to determine the packages that receive the redistributed classes and we have evaluate the quality of a remodularized Software Architecture by metrics for measuring Coupling and Cohesion of a Package . A large part of previous works related to oriented software metrics has focused on the issue of characterizing the class design, either looking internal complexity or relationship between a given class and other classes[16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26].

In the literature, there is also a body of work that focus on object oriented metrics from the standpoint of their correlation with software changeability [16][27], or from the standpoint of their ability to predicate softwair maintenability [16][28]. Other reasearchers argue that the measures resulted by the cohesion and coupling metrics of the previous works are open to interpretation [16] [28].
In general, there are few metrics in the the literature devoted to packages.

Our cohesion and coupling metrics we provide in this work are similar to the metrics provided by Ducasse [16].

8. Conclusion and discussion

In this paper we explore the issue of redistributing classes of a package to other packages. We use an approach based on Oriented Graph with the technique based on the definition of the shortest path, to determine the packages that receive the redistributed classes, and we have evaluate the quality of a remodularized software architecture by metrics for measuring coupling and cohesion of a package The results have an improvement at the level of remodularized architectures.

The results obtained at the level of the cohesion for the remodularized architecture 1 provides an optimum value 1 compared to the original architecture 1. The results of the coupling have an improvement at the level of remodularized architecture 1 compared to the original architecture 1.

The results of the cohesion and coupling metrics were conclusive for both approaches, namely:
- Remodularization of architecture 1 by the FCA formal concept method [30] [13] [14].

- Remodularization of architecture 1 by this method based on the directed graph [28] [31].

The formal method of FCA concept is not popularized among scientists as opposed to the use of the labeled directed graph.

It is for this reason that our directed graph approach is more effective in its simplicity and popularity.

The directed graph approach easily leads to the desired result because it is easily understood by a large scientific majority as opposed to the FCA method [30] known by a scientific minority results that also conclude that.

References

[1] H. Abdeen, “Visualizing, Assessing and Re-Modularizing Object-Oriented Architectural Elements”, PhD thesis, Lille University, 2009.
[2] H. Abdeen, S. Ducass, H.A. Sahraoui, and I. Alloui, “Automatic package coupling and cycle minimization ”, In International Woking Confence on Reverse Engineeing (WCRE), . IEEE Compter Society Press, Washington, DC, USA, pp. 103-112, 2009.
[3] F.B. Abreu, G. Pereira, and P. Sousa, “A coupling-guided cluster analysis approach to reengineer the modularity of object-oriented systems ”, In Proceeding of the confeence on Software Maintenance and Reengineering. CSMR „OO, IEEE Compter Society Press, , DC, USA, pp. 13, , 2000.
[4] G. Arévalo, S. Ducass, and O. Nierstrasz, “ Lessons leaned in appling fomal concept analysis to reverse engineering”, In Proceeding of the Third international conference on Fomal Concept Analysis, ICFCA‟05, Spinge-Velag, , Berlin. Heidelberg, pp. 95-112, 2005.
[5] M. Bauer and M. Trifu. Architecture-aware adaptive clustering of oo s ystems. In Poceedings of the Eighth Euromicro Working Conference on Software Maintenance and Reengineering (CSMR „O4), CSMR „O4, IEEE Compter Society, Washington, DC, USA, pp. 3, 2004. [6] B. Ganter and R. Wille, “Formal

Concept Analysis. Mathematical Fondations ", Spinge, 1999.
[7] B.S. Mitchell and S. Mancoridis, "Compaing the decompositions produced by software clustering algorithms using similarity measurements ", In ICSM, pp. 744-753, 2001.
[8] M.O‟Keeffe and M. I. Cinneide,"Seach-based refactoring fo software maintenance", J. Syst. Softw, pp. 81(4): 502-216, April 2008.
[9] O. Seng, J. Stammel and D. Burkhart, "Search- based determination of refactorings for improving the class structure of object-oriented systems", In Mike Cattolico, edito. GECCO, ACM, pp. 1909-1916, 2006.
[10] G.Snelting, "Software reengineering based on concept lattices", In CSMR, pp 3-10, 2000.
[11] T. Tilley, R. Cole, P. Becker and P.W. Eklund, "A survey of formal concept analysis support for software engineering activities", In Int. Conf. Fomal Concept Analysis (ICFCA), pp. 50-271, 2005.
[12] P. Tonella, "Concept analysis for module restructuring", IEEE Trans, Software Eng, pp. 27 (4): 351-363, 2001.
[13] L. M. Hakik, M. Huchard, R. El Harti and A.D. Seriai, "Exploration de la redistribution des classes d'un package par des techniques d'Analyse Formelle de Concepts", The first conference in software engineering (CIEL 2012), France, 2012.
[14] L. M. Hakik, , R. El Harti , "Exploring the Redistribution Classes of a Package with an Approach Based on Formal Concept Analysis", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, www.ijert.org, December 2013.
[15] A. Anwar Harti , "Formalisation par une approche IDM de la composition de modeles dans le profil VUML", Thesis. Toulouse University. 2009.
[16] S. Dcasse, N. Anquetil, M.U. Bhatti and A.C. Hora. , "Software metrics for package remodularisation", Research report, November 2011.
[17] S. R. Chidamber and C. F .Kemer, "A metrics suit for object oriented design", IEEETSE, pp. 20: 476-493, 1994. [18] F.B. Abreu and R. Carapuca, "Candidate metrics for objected-oriented software within a taxonomy framework. ", Journal of Sys, Sof, pp. 26: 87-96, 1994.
[19] W. Li and S. Henry, "Objected-oriented metrics that predict maintainability", Journal of Sys, Sof, pp. 23:111-112, 1993.
[20] W. Li, " Another metric suit for object oriented programming", Journal of Sys, Sof, pp. 44:155-162, 1998. [21] B.H. Selers, " Object-Oriented Metrics: Measures of Complexity", Prentice-Hall, 1996.
[22] J.M. Bieman and B.K. Kang, "Cohesion and reuse in an object-oriented system", In ACM Symposium on Software Reusability.,April 1995.
[23] J.M. Bieman and B.K. Kang, " Measuring design-level cohesion", IEEETSE, pp. 24(2) :111-124, February 1998.
[24] L.C. Briand, S. Morasca and V. R. Basili, "Defining and validation measures for object-based high-level design", IEEE TSE, pp. 722-743, 1999.
[25] L.C. Briand, J.W Daly and J. Wust, "A Unified Framework for Cohesion Measurement in Objected-Oriented Systems", Empirical Software Engineering, An International Journal, pp. 3(1):65-117, 1998.
[26] L.C. Briand, J.W Daly and J. Wust, "A Unified Framework for Coupling Measurement in Objected-Oriented Systems", IEEETSE, pp. 25(1):91-121, 1999. [27] H. Kabaili, R.K. Keller, F. Lustman, "Cohesion as changeability indicator in object- oriented systems", In Fifth Europ. Conf, on Sof. Maintenance and Reengineering. CSMR 01, IEEE Computer Society, Washington, DC, USA, pp. 39-46, 2001. [28] R.K. Bandi, V.K. Vaishnavi and D.E. Tuk, "Predicting maintenance performance using object- oriented design complexity metrics", IEEETSE, pp. 29: 77-87, 2003.
[27] http://liris.cnrs.fr/csolnon/polyGraphes.pdf.
[28] L.M. Hakik. R. El harti .Technique Of Redistribution Classes Of A Package With An Approach Based On Oriented Graph And Evaluation Quality of A Remodularized Software Architecture. International Journal of Innovative Research in Science, Engineering and Technology. pp. 8133-8140 Vol. 3, Issue 1, January 2014.
[29] L.M. Hakik. R. El harti. Application of the Relational Concept Analysis (RCA) for Remodularization of a Software Architecture and Comparison with Others Techniques Based on FCA and Oriented Graph IJCSNS International Journal of Computer Science and Network Security, VOL.14 No.3, March 2014.
[30] L.M. Hakik, R. El Harti. Measuring Coupling and Cohesion to Evaluate the Quality of a Remodularized Software Architecture Result of an Approach Based on Formal Concept Analysis. IJCSNS International Journal of Computer Science and Network Security .Vol. 14 No. 1 pp. 11-16. Journal ISSN : 1738-7906. January 2014.
[31] L.M. Hakik, R. El harti. Technique Of Redistribution Classes Of A Package With An Approach Based On Oriented Graph And Evaluation Quality of A Remodularized Software Architecture. International Journal of Innovative Research in Science, Engineering and Technology.ISSN:2319-8753. Vol. 3, Issue 1, January 2014.

Lala Madiha Hakik received the Master in Software Engineering- Research-, University of Montpellier 2, France.2009 and Phd in software engineering from Hassan 1st University, Faculty of Science and Techniques of Settat, Morocco, 2014. She is a Research Professor in *Higher Institute of Engineering and Business ISGA*, Casablanca, Morocco