دراسة تجريبية للمقارنة بين مقياسLack of Cohesion in : Methods (Lcom5) ومقياس Cohesion on Method and : Attribute (COMA).

أ. يونس أبوبكر عبدالصمد. محاضر – كلية التقنية الهندسية – مسلاته

الملخص

تقدم هذه الورقة البحثية دراسة تجريبية حول مقارنة مقياس التماسك (LCOM)، ومقياس (COMA)، حيث إن مقياس (LCOM5) يستند في عمله على حساب عدد أزواج الطرق، التي لا تشترك السِّمات في أصنافها؛ أما بالنسبة لمقياس (COMA) والذي يعتمد في عمله على استعمال الخاصِّيَّة، واستعمال الطريقة المستدعاة، وكذلك على التفاعل المباشر بين الخاصِّيَّة، والطريقة، والطرق مع بعضها. ويُستعمَل هذا المقياس في مستوى التصميم البرمجي، والغرض من الدراسة: هو تحديد عيوب مقياس (Lcom5) ومعالجتها، والتي تتمثَّل في عدم قدرة المقياس على حساب تماسك الطرق في الأصناف (CLASS)، واعتمدتُ في المنهجية على تصميم أداة باستخدام (C#) تقوم بعملية حساب تماسك البرنامج الشيئي بـــ (LCOM5, و COMA) حيث نستنتج من هذا القياس أن (LCOM) لا يُفرِّق _ بشكل صحيح _ التماسك بين الفئات أو الأصناف، بينما يحسب (COMA) ذلك، وهذا يُحقِّق الفرضيَّة بأن مقياس (LCOM5) ليس له القدرة على حساب التماسك الحقيقي في الأصناف؛ بينما مقياس (COMA) يقوم بذلك.

الكلما ت المفتاحية: (التماسك، المقاييس البرمجيَّة، مقاييس التمسُّك).

Abstract

This paper presents an experimental study on the comparison between the LCOM5 cohesion metric and the COMA metric, The LCOM5 metric is based on the calculation of a number Pairs of roads with no traits in common.

Software metrics can help address the most critical issues in software development and can provide support for planning, predicting, monitoring, controlling, and evaluating the quality of both software products and processes . Quite a number of object oriented cohesion metrics have been proposed; we identified lapses in the definition of some of the object-oriented cohesion metrics. In this paper we propose a new metric for measuring cohesion at the design level. The purpose of the study is to identify and address the defects of the LCOM5 metric , The methodology relied on designing a tool using C # that calculates the consistency of the object-oriented program with LCOM5 and COMA.

we can see that , LCOM5 is solely based on attribute referencing and thus will not show an increase in the cohesion of a class where there is method invocation in addition to attribute referencing, Unlike COMA, LCOM5 could not differentiate between the cohesion of this class . But COMA was able to differentiate between the cohesion of these classes.

المقدمة:

"أيُّ شيء لا يمكن قياسه؛ لا يمكن السيطرة عليه". (Dr. Linda, 1998) القياس والمقاييس لعبت دوراً فعَّالاً في الممارسة العملية عبر تاريخ البشرية، وكذلك دوراً في جميع الأنشطة البشرية الحيويَّة.

لا يمكن التحكم في أيِّ عملية إنتاجية، أو التطوير يكون مفيداً فيها؛ في حالة عدم وجود مقاييس فعَّالة.

يعتمد المقياس Lack of Cohesion Measure5 (LCOM5) (**Amendeep, Paneet,2013**). على قياس التماسك في الأصناف. من أهم العيوب في هذا المقياس هو عدم قدرته على قياس تماسك الطرق داخل الأصناف، **وهي تُمثِّل بيان المشكلة.** بينما يعتمد المقياس Cohesion on Method and Attribute (COMA) التماسك على

الطريقة والسِّمة، ويستند على الإستراتيجية الثانية (أي المقاييس التي تستند إلى علاقات USES) **(Alhadi,2013)**. على الرغم من أنه لا يمكن التقاط جميع الاستخدامات (التفاعلات) بشكل شامل في نهاية مرحلة تصميم البرنامج، يمكن استخدام المعلومات المتاحة في هذه المرحلة، لتحديد مقياس يمكن استخدامه على مستوى التصميم. في حساب هذا المقياس نقوم _ ببساطة _ بتتبع جميع الطرق التي تُستخدَم للوصول إلى كل سمة في الصنف، وجميع الطرق التي تُستخدَم (استدعاء) كل طريقة في الصنف، و يعتمد المقياس على (LCOM5) ويعالج بعض أوجه القصور فيه.

حدود الدراسة:

تتمثَّل حدود الدراسة في شيئين، هما:

1. أوجه القصور في مقياس (LCOM5) و قيام مقياس (COMA) بمعالجة هذا القصور.
2. تصميم أداة تقوم بالمقارنة والتحقق من المقياسين.

الدراسات السابقة:

- دراسة لليندا بدري، ومراد بدري، "مقترح لتماسك صنف جديد: دراسة تجريبية"، قسم الرياضيات والحاسوب جامعة كيبيك، كندا، مجلة تكنولوجيا الأشياء، سنة النشر: 2004.
- دراسة لستيف كونسيل، لستيفن سويفت، "تفسير وفائدة لثلاثة مقاييس تماسك للتصميم في البرمجة الشيئية"، ACM المعاملات في هندسة ومنهجية البرمجيات، أبريل، 2006، https://doi.org/10.1145/1131421.1131422.

- دراسة لجهاد الدلال، "مقياس عدم التماسك القائم على الطرق في الأصناف"، قسم المعلومات، جامعة الكويت. متاح على الإنترنت، https://doi.org/10.1016/j.procs.2011.01.053.
- دراسة لهاري غانيش، "مقياس تماسك اتصالي للبرمجة الشيئية"، قسم علوم الحاسب، كلية سمو الراجح، بودوكوتاي، الهند، المجلة الدولية لتطبيقات الكمبيوتر، سنة النشر : 2016م.

منهجية الدراسة: اعتمدت في هذه الدراسة على المنهج الكَمِّي من خلال:

1. منهجية التحليل: وتَتمثَّل في دراسةِ لمقاييس التماسك السابقة وآلية عملها.
2. مصادر البيانات: الدراسات السابقة في هذا المجال و الدراسة التجريبية.
3. أخذ عَيِّنات عشوائية لبعض برامج التوجُّه الكينوني، وقياس التماسك بها؛ باستخدام أداة تمَّ تصميمها، باسخدام لغة (C#) السي شارب، وتحليل نتائجها إحصائياً.

الإطار النظري.

1. الاقتران والتماسك:

في هندسة البرمجيات، الاقتران هو الدرجة التي تعتمد فيها وحدات البرنامج على بعضها البعض، حيث تعتمد كل وحدة برنامج على كل وحدة من الوحدات الأخرى، وعادةً ما يتعارض الاقتران مع التماسك. في البرمجة الشيئية يكون الاقتران هو مدى قوة ارتباط فئة واحدة، للحصول على معرفة، أو اعتماد على فئات أخرى. لا تعتمد الفئة ذات الاقتران المنخفض على العديد من الفئات الأخرى، من ناحية أخرى، تعتمد الطبقة ذات

الاقتران العالي (أو القوي) على العديد من الفئات الأخرى، قد تكون هذه الفئات غير مرغوب فيها.(Letha,2004).

التماسك هو سمة برمجيَّة داخلية، تُصوِّر مدى جودة الاتصال بمكوِّنات وحدة البرنامج. ويمكن تحديد ذلك من خلال معرفة المدى الذي تتطلبه المكونات الفردية للوحدة لتنفيذها مهمة.

في التوجُّه الكينوني: التماسك هو الدرجة التي ترتبط بها الطرق داخل الفئة بعضها البعض، والعمل معاً لتوفير سلوك محدود.

2. العلاقة بين الاقتران والتماسك:

هناك ارتباط وثيق بين الاقتران والتماسك، حيث يؤدِّي التماسك المنخفض عادةً إلى ارتفاع الاقتران، والعكس بالعكس. {4، 5، 6}.

1.2 **الاقتران المنخفض:** الاقتران هو مقياس لمدى قوة الفئة، واعتمادها على فئات أخرى. لا تعتمد الفئة ذات الاقتران المنخفض على العديد من الفئات الأخرى، ومن ناحية أخرى؛ فإن الفئة ذات الاقتران المرتفع (أو القوي) تعتمد على العديد من الفئات الأخرى، وقد تكون هذه الفئات غير مرغوب فيها؛ بسبب ما يلي من الأسباب:

- التغييرات في الفئات ذات الصلة تفرض تغييرات محلية.
- الصعوبة في الفهم عند العزل، أو الفصل بينهما.
- يصعب إعادة استخدامها؛ لأن استخدامها يتطلب وجود فئات أخرى تعتمد عليها. وبالتالي فإن الاقتران المنخفض، يدعم تصميم الطبقات الأكثر استقلالية؛ مما يُقلِّل من تأثير التغيير.

الحالة القصوى للاقتران المنخفض غير مرغوب فيها، أي عندما لا يكون هناك اقتران بين الفئات على الإطلاق، أو عندما تكون منخفضة للغاية. إذا تمَّ أخذ اقتران

منخفض إلى فائض، فإنه ينتج تصميمًا رديئًا؛ لأنه يؤدِّي إلى عدد قليل غير مرتبط، ومُعقَّد. **(Letha ,2004)**.

2.2 التماسك العالي: التماسك في البرمجة الشيئية: هو مقياس لمدى قوة الاتصال داخل الفئة الواحدة. الفئة ذات التماسك المنخفض، من أبرز عيوبها ما يأتي:

- من الصعب إعادة استخدامها.
- من الصعب الحفاظ عليها.

بشكل عام، فإن العلاقة بين الاقتران والتماسك، هي أن الاقتران يجب أن يكون منخفضًا؛ بينما يبقى التماسك مرتفعًا. **(Hitz, Montazeri ,1996)**.

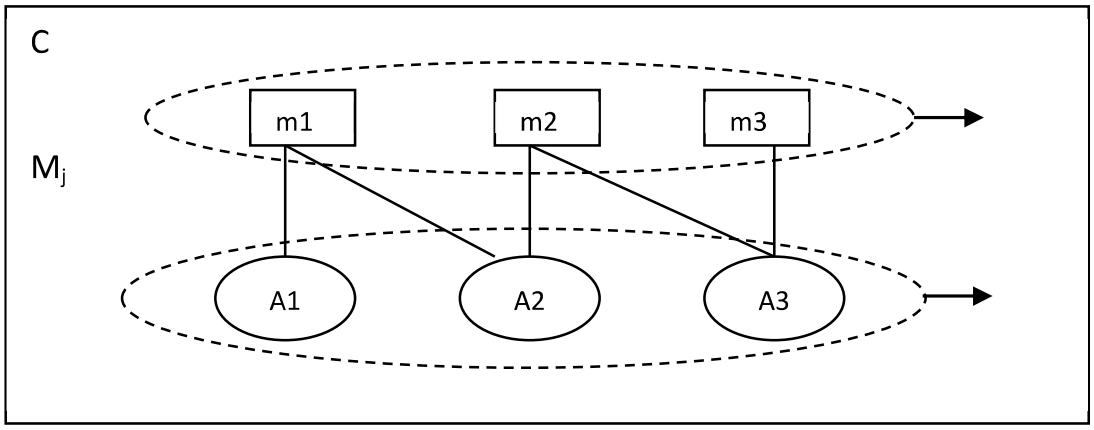
3. مقاييس تماسك البرمجة: يقدم هذا الجزء تعريفاً لمقياس من مقاييس التماسك الطبقي وهو (LCOM5).

3.1 مقياس (LCOM5): استند عمله على مجموعة المقاييس لـ (Chidamber) و (Kemere) (Dr. Linda, 1998) وهو مُقَيَّم من وجهة نظر رياضية، وتمَّ تحديد صيغة جديدة لمقياس (LCOM) وعرفت القيم بين 0 و 1 كنسب مئوية من القيمة المثالية، ومقياس (LCOM5) تمَّ تعريفه على النحو التالي:

ضع في اعتبارك مجموعة من الطرق Method {i = 1} {Mi،.. ...، m) للفئة (C) التي تصل إلى مجموعة من الخواصِ Attributes {j = 1} {Aj،.. ...، a). دع عدد الطرق التي تصل إلى الخاصِيَّة (Aj) تكون (Aj) وإجمالي عدد الخواصِ في {Aj} هو:

$$LCOM5 = \frac{1}{a.m}\sum_{I=1}^{a}(A_i)$$

هذا التعريف مُوضَّح في الشكل رقم: (1).



الشكل رقم: (1) الفئة (C) بثلاثة طرق وثلاثة خواصٍ.

من المثال، لدينا: m = 3 و a = 3 ، فإن

$$LCOM5 = \frac{1}{3 \times 3}(1 + 2 + 2) = \frac{5}{9}$$

المبحث الثاني: مقياس التماسك الجديد.

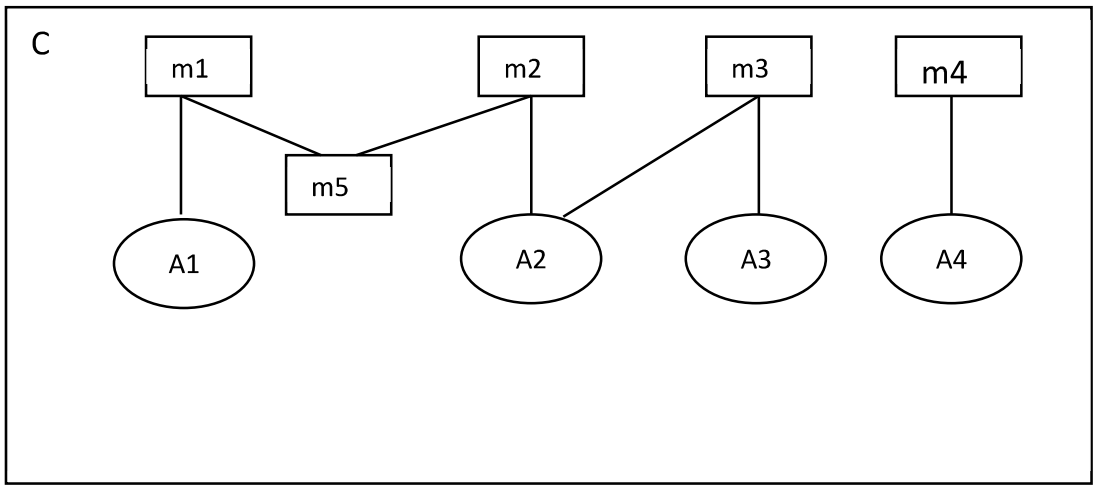
تمَّ اقتراح مقياس جديد لقياس التماسك، يمكن استخدامه على مستوى التصميم، و في هذا المقياس نتتبَّع_ ببساطة _ جميع الطرق التي تستخدم للوصول لكل الخواصِ في الفئة، وجميع الطرق التي (تستدعي) بعضها في الفئة.

ويعتمد في عمله على مقياس (LCOM5)، ويعالج بعض أوجه القصور فيه.

1. التماسك المعتمد على الطريقة والخاصِّية (COMA).

أغلب مقاييس التماسك الحالية تحاول قياس التماسك من ناحية الأخذ في الحسبان التفاعل بين الطرق والخواصِ فقط.

أما المقياس الجديد (COMA) فقد استند في عمله على استخدام الخاصِّية، واستخدام الطريقة المستدعاة داخل الفئة، وهو يعتمد على التفاعل المباشر بين الطرق



الشكل رقم: (2) الاتصالات المباشرة [2].

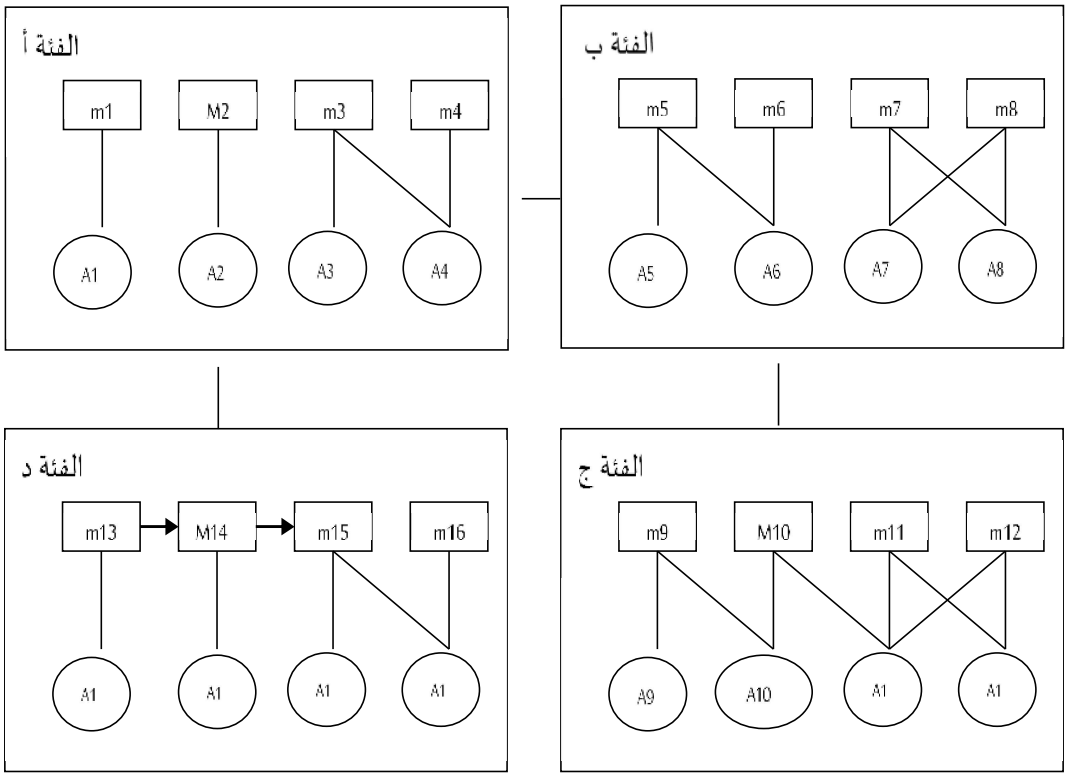
من الشكل رقم: (2) نلاحظ أن (m2) و (m3) مرتبطان بشكل مباشر؛ لأن كليهما يصل نفس الخاصِّية (A2)؛ أما بالنسبة للاتصال بين (m1) و (m2) عبر (m5) نحتاج إلى النظر في استخدامات الطريقة في الفئة.

2. النتائج:

لتقييم تماسك الصنف باستخدام (LCOM5) و (COMA)، ومقارنة نتائجهما؛ يكون وفق مايلي:

2-1 الحساب الرياضي:

يتم حساب التماسك للفئات (أ، ب، ج، د) من الشكل رقم: (6) باستخدام (COMA) و (LCOM5)، و النتائج التي تمَّ الحصول عليها باستخدام الحساب الرياضي كما في الجدول رقم: (1).



الشكل رقم: (3) يُوضِّح تماسك أربعة أصناف مختلفة.

1- يتَّضح من الجدول رقم: (1) أن كلاً من (LCOM5) و (COMA) يُميِّزان _ بشكل صحيح _ التماسك بين الفئات، حيث إن قيمة التماسك تزداد مع زيادة التقاطع بين المكوِّنات، وهذا واضح للفئات (أ، ب، ج) ولكن ليس للفئة (د)، حيث توجد أربع طرق، وأربع خواصّ، كما في الفئة (أ).

2- الفرق الوحيد بين الفئتين (أ، د) هو أن (m13) تستدعي (m14) وأن (m14) تستدعي (m15)، حيث يتمُّ حسابها من قبل (COMA) بينما لا يمكن لـ (LCOM5) التفريق بين تماسك الفئة (أ) والفئة (د)؛ لأنه لا يستطيع حساب التماسك بين الطرق.

الجدول رقم: (1) النتائج التي تمَّ الحصول عليها بعد تقييم التصميم في الشكل رقم: (7).

اسم الصنف	LCOM5	COMA
الفئة أ	0.312	0.156
الفئة ب	0.437	0.218
الفئة ج	0.5	0.25
الفئة د	0.312	0.239

2-2 استخدام أداة البرمجة:

تمَّ تصميم برنامج، تمَّت برمجته باستخدام لغة بـ (C#) لحساب التماسك، حيث يقوم بحساب تماسك الفئة لـ (LCOM5) و (COMA) ثم يقارن النتائج التي تمَّ الحصول عليها.

خطوات البرنامج مُلخَّصةً في التالي:

1. قراءة برنامج مكتوب بلغة (++ C) أو (C#) أو أيّ لغة برمجة شيئية.
2. البحث عن الفئات في البرنامج وتخزينها في ملف.
3. تحليل الأصناف الموجودة في الخطوة (2) للعثور على كل فئة وحساب عدد الطرق، وعدد الخواصّ.
4. حساب التماسك بتطبيق صيغ (LCOM5) و (COMA) المُوضَّحة مُسبَقاً.
5. عرض النتائج التي تمَّ الحصول عليها في الجدول رقم: (1).
6. نتائج المُدخَلات والمُخرَجات في البرنامج معطاة بالجدول رقم: (2).

الجدول رقم: (2) مُدخَلات ومُخرَجات برنامج (C#).

Class Name	No of attributes	No of methods	Sum of Ai	Sum of Mi	LCOM5	COMA
Class AAA	4	3	8	1	0.6666667	0.4166667
Class BBB	5	3	9	2	0.6	0.4666667
Class CCC	3	3	9	6	1	1
Class DDD	3	3	8	1	0.0888889	0.5277778
Class EEE	4	3	9	2	0.75	0.5416667
Class FFF	4	3	9	6	0.75	0.875

النتائج :

تمَّ الحصول عليها، باستخدام الأداة البرمجيَّة: كما هي مُوضَّحة بالجدول رقم: (2) وهي ست فئات مختلفة، مع عدد مختلف من استخدام الخاصِّية، وطريقة الاستدعاء. يتمُّ حساب تماسك كل فئة من الفئات الست، باستخدام (LCOM5) و (COMA). حيث يتمُّ ملاحظة أن آخر عمودين في الجدول رقم: (2) أن (LCOM5) لا يُفرِّق _ بشكل صحيح _ التماسك بين الفئات، بينما يحسب (COMA) ذلك.

الخاتمة:

نقص مقاييس تماسك البرامج في البرمجة الشيئية؛ أدَّى إلى اكتشاف (COMA) وهو مقياس برمجي جديد مُصَمَّم ليتم استخدامه في مرحلة تصميم البرنامج، ضمن دورة حياة البرمجيات. الهدف من مقياس (COMA) هو قياس مقدار التماسك للأصناف؛ بالأخذ بين اعتبار الطرق والخواصِّ داخل الصنف.

يشير التماسك العالي إلى ارتفاع ناتج قسمة تكامل وحدات البرامج وهو أفضل عامل؛ لقابلية الصيانة والتعديل، وإمكانية الفهم من البرامج، علاوةً على ذلك؛ فإن

التماسك العالي أيضًا يُقلِّل من تعقيد البرنامج الكلي، وعلاوةً على ذلك؛ فإن مقياس (COMA) يساعد المُطوِّرين على تقييم برامجهم، لضبط جزء الترميز ،الذي يؤدِّي _ بالضرورة _ إلى تقليل من التكاليف التشغيلية والوقت. تقييم مقياس (COMA) لقد أثبت أنه مقياس مُؤهَّل لِيُؤخَذ بعين الاعتبار. أوجه القصور الموجودة في مقياس (LCOM5)ومن ثَمَّ فإن المقياس سيعمل على نطاق واسع في صناعات البرمجيَّات، و لبناء منتجات ذات جودة.

المراجع:

1- Dr. Linda H. Rosenberg, " Applying and interpreting Object Oriented Metrics", Mr. Larry Hyatt Software Assurance Technology Center, NASA SATC, 1998.

2- Amendeep Kaur, Paneet Jai Kaur," Class Cohesion Metrics in Object Oriented Systems", International Journal of Software Web Science,IJSWS, 12-365,2013,78-82.

3- Chidamber Shyam R., Kamerer Cris F., "A metric Suite for Object Oriented Design", IEEE Transactions on Software Engineering, Vol. 20, Number 6, Pages 476-493, June 1994.

4- W. Stevens,G. myers, L. Constantine, "Structured Design", IBM Systems Journal, 13(2), 115-139, 1974.

5- Letha H. Etzkom et al., "A comparison of Cohesion metrics for Object Oriented Systems", Information and Software Technology, Vol. 46, 2004, 677-687.

6- Ion Ivan, Zsolt Mark, Paul Pocatilu, Marius Popa, Doru Urgureanu, " Cohesion –Software Quality Characteristics ", in Studies and researchs of Economic Calculus and Cybernetics, Bucharest, 2003.

7- Hitz M., Montazeri B., "Measuring Coupling and Cohesion in Object Oriented Systems", in proc. Int. Symposium on Applied Corporate Computing, Monterrey, Mexico, October 1995.

8- Hitz M., Montazeri B., " Chidamber and Kemerer's Metrics Suite: A measurement Theory Perspective" , IEEE Transactions on Software Engineering, Vol. 22 no. 4, April 1996.

9- Bieman J. M., Kang B., " Cohesion and Reuse in an Object Oriented System", in Proc, ACM Symp., Software Reusability(SSR'94), 259-262,1995.

10- Ronan Fitzpatrick, "software Quality ", Accessed on July 2009, http://satic.gsfc.nasa.gov/metrics/codemetrics/00/definitions/index/html.

11- Hnderson-sellers, B., "Object Oriented Metrics measures of Complexity ", 1996, Prentice-Hall.

12- Basili, V. R.,Briand, L. C., Melo, W. L., " A Validation of Object Oriented Design as Quality Indicators ", IEEE Trans. Software Eng., Vol. 22, no 10, pp751- 761, Oct.1996.

13- Alhadi T. A. et al., " Classification for Object-Oriented Cohesion metrics.",
International Journal of Advanced Research in Computer Science and Electronics Engineering ", 2012.