المجلة العراقية للعلوم الإحصائية (29) 2019
عدد خاص بالمؤتمر الطلابي الاول
ص ص [47-60]

بناء نظام قاعدة معرفة (KBOSR) لتحسين إعادة استخدام البرمجيات

علي زين العابدين عبدالرزاق* د.صفوان عمر حسون**

alizeinswe2017@yahoo.com Dr.safwan1971@uomosul.edu.iq

المستخلص

نتيجة للتطور الحاصل في هندسة البرمجيات وخصوصاً فيما يتعلق بتحسين أنظمة البرمجيات ضمن أقصر وقت ممكن لتحسين عملية التطوير وتقليل الكلفة قدر الإمكان ، ظهر مقياس إعادة استخدام البرمجيات وهو أحد أهم مقاييس هندسة البرمجيات الذي يساعد على تقليل الجهد والوقت في عملية التطوير ، إعادة استخدام البرمجيات هو استخدام مكونات البرمجيات المبنية سابقاً في بناء أنظمة البرمجيات الجديدة وتحديثها ويعتمد مقياس إعادة استخدام البرمجيات على العديد من العوامل التي تؤثر عليه ، في هذا البحث تم الاعتماد على عاملي التلاحم والارتباط في حساب مقياس إعادة استخدام البرمجيات .

تم الاعتماد على عاملي التلاحم والارتباط في حساب مقياس إعادة الاستخدام للأصناف ، إذ يقيس عامل التلاحم مدى الاعتمادية الداخلية للدوال والمتغيرات ضمن الصنف الواحد ، في حين عامل الارتباط يهتم بحساب مدى الترابط والاعتمادية بين الأصناف المختلفة ضمن الحزمة الواحدة ، اذ تم بناء نظام قاعدة معرفة لإيجاد حالة التلاحم وحالة الارتباط للأصناف بالاعتماد على قيم التلاحم والارتباط التي تم حسابها ، و كذلك حساب حالة إعادة الاستخدام للأصناف لمعرفة مدى صلاحية الصنف لإعادة استخدامه ، إذ تم استخدام ماكنة استنتاج من نوع التسلسل الأمامي حيث يتم الوصول للهدف المطلوب من خلال تدقيق جزء الشرط للقاعدة .

تم اقتراح نظام قاعدة معرفة (KBOSR) يحتوي على (61) قانون لتقديم وعرض أسباب عدم صلاحية الأصناف لإعادة الاستخدام من خلال عرض مواضع الضعف في الشفرة ،

* باحث / قسم البرمجيات / كلية علوم الحاسوب والرياضيات / جامعة الموصل
** استاذ مساعد / قسم البرمجيات / كلية علوم الحاسوب والرياضيات / جامعة الموصل

تاريخ استلام البحث 2019/4/29 تأريخ القبول 4/7/ 2019

وتقديم النصائح والإرشادات والتعديلات الواجب تطبيقها على الشفرة لجعل الصنف قابل لإعادة الاستخدام وعرض هذه النصائح والإرشادات على واجهات المستخدم بطريقة واضحة و سهلة الفهم ، فمن خلال البحث والاطلاع على البحوث السابقة ، يعد نظام KBOSR أول نظام قاعدة معرفة لتحديد حالة إعادة الاستخدام للأصناف وتحسينها .

تم تطبيق النظام واختباره على ثلاثة عشر صنفاً موزعين على حزمتي جافا ، الحزمة الأولى تحتوي على خمس أصناف والحزمة الثانية تحتوي على ثمانية أصناف ، وأثبتت نتائج الاختبار قوة النظام وصحته في تحديد حالة إعادة الاستخدام للأصناف ، و دقة المقترحات والنصائح المقدمة بالاعتماد على بعض نقاط الضعف لشفرات الأصناف .

This is an open access article under the CC BY 4.0 license http://creativecommons.org/licenses/by/4.0/)

Design a knowledge base system (KBOSR) for enhancing software reusability

ABSTRUCT

As a result of the development of software engineering , especially with regard to improving the software systems within a short time to enhance the development process and reduce cost as much as possible . Software reusability is one of the most important software engineering metrics which aim to reduce the effort and time in the development process . It is the process of using pre-existing software components for implementing and updating new software systems , thereby reducing effort , time and cost in addition to increasing the quality and productivity of software systems . Many factors affect the reusability of a software , of these factors , coupling and cohesion are used in this search for the aforementioned purpose .

Reusability of classes is determined based on coupling and cohesion results . Cohesion measures the intra-dependability of functions and variables inside the class , while coupling measures the inter-dependability among classes in the same package . The proposed expert system also used for computing coupling , cohesion and reusability status , and produce suggestions to enhance the reusability status of unreusable classes by offering intelligence advices which depending on the weakness of the tested codes .

Proposed knowledge Base System (KBOSR) contains (61) rules to provide and demonstrate why the items are not reusable by displaying code weaknesses , and to provide advice, guidance and modifications to the code for making the product reusable and displaying these guidelines and instructions on user interfaces in a clear and easy-to-understand way, KBOSR is the first knowledge base system to determine the status of reuse and improve it .

The knowledge base was tested using 13 classes were distributed in two packages , first package contains 5 classes , second package contains 8 classes , coupling and cohesion values of classes are computed . Coupling , cohesion and reusability status of these classes are determined , then , by proposed expert system . Finally , suggestions of improvement of unresable classes are displayed . The results of testing process have proved the strength and validity of the proposed knowledge base in determining reusability status of the software components , and the accuracy of the advices that were suggested , these advices are practically applied on the codes of interest and the modified codes proved to be highly reusable .

1- المقدمة :

نتيجة للتطور الحاصل في هندسة البرمجيات وخصوصاً فيما يتعلق بتحسين أنظمة البرمجيات ضمن أقصر وقت ممكن لتحسين عملية التطوير وتقليل الكلفة قدر الإمكان ظهر مقياس إعادة استخدام البرمجيات وهو أحد أهم مقاييس هندسة البرمجيات الذي يساعد على تقليل الجهد والوقت في عملية التطوير عن طريق استخدام مكونات البرمجيات المبنية سابقاً في بناء أنظمة البرمجيات الجديدة وتحديثها ، إذ يساعد على تطوير النظام بأقل جهد وأسرع وقت وأقل كلفة وبأفضل جودة ممكنة

(Ashwin & Tomar ,2012)

يعتمد مقياس إعادة استخدام البرمجيات على العديد من العوامل التي تؤثر عليه ، في هذا العمل تم الاعتماد على عاملي التلاحم والارتباط في حساب مقياس إعادة استخدام البرمجيات .

تم في هذا العمل بناء نظام قاعدة معرفة مقترح لتحديد حالة إعادة الاستخدام للمكونات والأصناف وكذلك تقديم النصائح الذكية من خلال 61 قانون مثبت في نظام قاعدة المعرفة لتحسين هذه المكونات والأصناف وجعلها صالحة لإعادة الاستخدام .

2- مشكلة البحث

تكمن مشكلة البحث في إيجاد طريقة للتأكد من مدى صلاحية إعادة استخدام الأصناف التي تم تطويرها وبنائها سابقاً ، حيث تم بناء نظام قاعدة معرفة لتحديد حالة إعادة استخدام الأصناف وتقديم النصائح الذكية لجعل الأصناف صالحة لإعادة الاستخدام ، وذلك بالاعتماد على عاملي التلاحم والارتباط في قياس إعادة الاستخدام للأصناف .

3- أهداف البحث

الهدف من هذا البحث هو بناء نظام قاعدة معرفة لتحديد أمكانية إعادة الاستخدام للبرمجيات وتحسينها بالاعتماد على عاملي التلاحم والارتباط .

وتتلخص أهداف البحث بما يأتي :-

1. تقديم مقترحات ذكية تم بنائها بشكل 61 قانون في قاعدة المعرفة المقترحة لتحسين حالة إعادة الاستخدام للأصناف وجعلها قابلة لإعادة الاستخدام .
2. عرض هذه المقترحات على شاشة العرض بطريقة سهلة ومفهومة للمستخدم بالاعتماد على الواجهات الرسومية لبرنامج Matlab.

4- أعادة استخدام البرمجيات :

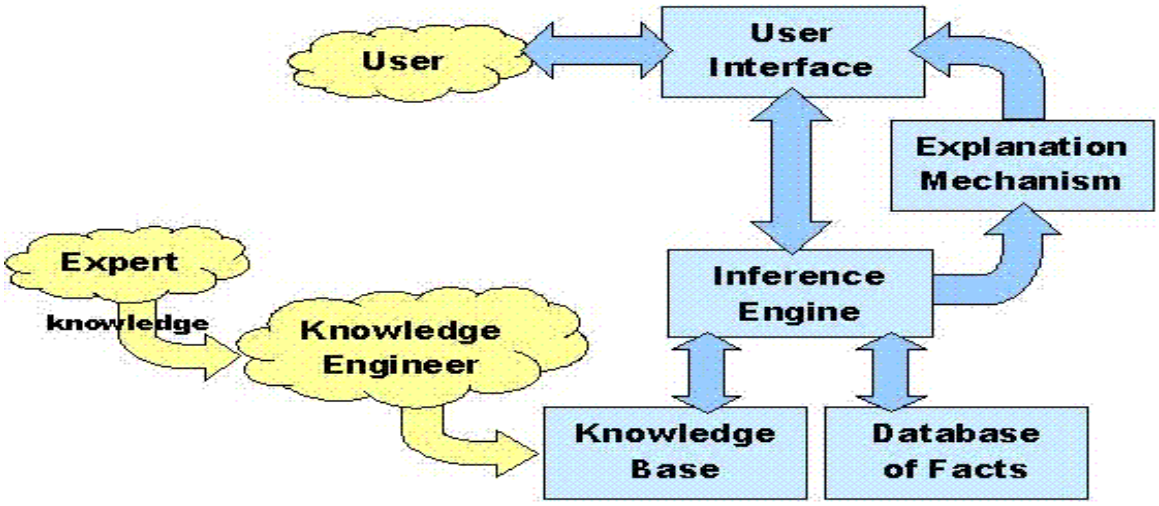
مقياس إعادة استخدام البرمجيات هو أحد أهم المقاييس التي تحسن من إنتاجية وجودة البرمجيات ، وهو إعادة استخدام برنامج أو جزء من برنامج تم تطويره سابقاً لنفس غرض البرنامج قيد التطوير في البرنامج أو التطبيق قيد التطوير وذلك من أجل التقليل من إهدار الوقت في كتابة شفرة (code) تم تطويرها سابقاً ، واختزال الجهد المطلوب لتطوير هذه الشفرة . (Aditi & Harleen ,2015)

هناك العديد من العوامل التي تؤثر على مقياس إعادة استخدام البرمجيات مثل التعقيد (Complexity) والجودة (Quality) و التلاحم (Cohesion) والارتباط (Coupling) وغيرها ، في هذا البحث تم اعتماد عاملي التلاحم والارتباط . (Neha ,2014) (Neha & Surender & Manjot Sadana ,2013)

إذ يشير التلاحم بين الأصناف (Classes) الى ترابط الوظائف أو الدوال ضمن الصنف الواحد وهو يؤثر بشكل طردي على مقياس إعادة الاستخدام (كلما تزداد قيمة التلاحم للصنف فإن قيمة إعادة الاستخدام لهذا الصنف تزداد) ، أما الارتباط بين الأصناف فيشير إلى مدى الاعتمادية المتبادلة بين هذه الأصناف، ويؤثر بشكل عكسي على مقياس إعادة الاستخدام (كلما تزداد قيمة الارتباط للصنف فإن قيمة إعادة الاستخدام لهذا الصنف تقل). (Ahmed & Amr & Hesham , 2016) (Yogesh & Yogyata , 2012)

الانظمة الخبيرة :

النظام الخبير هو برنامج حاسوبي يمكنه حل مشكلة معقدة في مجال معين تم تصميمه باستخدام مفهوم الذكاء الاصطناعي ، اذ يعتبر النظام الخبير تطبيق ينظم المعرفة ضمن قواعد و حقائق و إجراءات لحل المشاكل بطريقة مشابهة للتفكير البشري وبمستوى قريب من الخبير البشري ، الشكل التالي يوضح الهيكلية العامة للنظام الخبير: (Keith Darlington ,2007)



شكل (1) : الهيكلية العامة للنظام الخبير

5- نظام قاعدة المعرفة المقترح (KBOSR) :

تم تطبيق نظام قاعدة المعرفة المقترح (KBOSR) على شفرات جافا من أجل تحديد أمكانية أن تكون هذه الشفرات قابلة لإعادة الاستخدام وكالتالي :-

- يتم في بادئ الأمر تطبيق الماسح والمحلل (Scanner and Parser) على الشفرة وحساب قيمة التلاحم والارتباط لكل صنف (Class) من الاصناف الموجودة في الشفرة و حساب مقدار التلاحم والارتباط لكل صنف وتحديد حالة التلاحم والارتباط لكل صنف وبالتالي تحديد إمكانية إعادة استخدام هذه الأصناف .(Ali and Safwan,2017)
- إقتراح وتقديم (61) قانون تم تثبيته في نظام قاعدة المعرفة المقترح للتحسين على الشفرة وجعلها صالحة لإعادة الاستخدام وعرض هذه المقترحات للمستخدم باستخدام الواجهات الرسومية لبرنامج الماتلاب .

يوضح الشكل (2) خطوات نظام قاعدة المعرفة المقترح (KBOSR) .

الشكل (2) يوضح خطوات نظام قاعدة المعرفة المقترح (KBOSR)

6- التطبيق العملي للنظام المقترح :

تم تطبيق و اختبار نظام قاعدة المعرفة المقترح على 13 صنفاً موزعين على حزمتي جافا ، الأولى تحتوي على 5 أصناف والحزمة الثانية تحتوي على 8 أصناف ، بعد تطبيق عملية المسح والتحليل على الشفرة وحساب قيم عاملي التلاحم والارتباط للأصناف ، وتحديد حالات التلاحم والارتباط وإعادة الاستخدام للأصناف من خلال نظام قاعدة المعرفة المقترح ، تم عرض مقترحات التحسين على الأصناف غير المؤهلة لإعادة الاستخدام .

أثبتت نتائج الاختبار قوة النظام المقترح وصحته في تحديد حالة إعادة الاستخدام للأصناف ، و دقة المقترحات والنصائح المقدمة بالاعتماد على بعض نقاط الضعف لشفرات الأصناف ، حيث تم تطبيق المقترحات المقدمة من نظام قاعدة المعرفة المقترح عملياً على الشفرات التي تم اختبارها وعند إعادة اختبار الشفرات التي تم التعديل عليها أصبحت حالة الاستخدام لهذه الشفرات عالية .

7- نتائج التطبيق العملي للنظام المقترح :

تم تطبيق النظام المقترح على شفرتي جافا وتم عرض قيمة التلاحم والارتباط لكل صنف وحالة التلاحم والارتباط للاصناف ومن ثم عرض مقترحات التحسين على الاصناف لجعلها صالحة لإعادة الاستخدام .

الجدول (1) يوضح قيم التلاحم والارتباط للاصناف في الشفرة الاولى ، والجدول 2 يوضح حالة التلاحم والارتباط واعادة الاستخدام لهذه الاصناف ، والجدول 3 يوضح مقترحات التحسين لهذه الاصناف .

الجدول (1) : يوضح قيم التلاحم والارتباط لأصناف الشفرة الاولى .

أسم الصنف	قيمة التلاحم للصنف	قيمة الارتباط للصنف
Class1	1.000000	2.500000
LinkedList	0.142857	0.125000
Class2	1.333333	0.000000
LLIterator	0.166667	1.433333
Second	2.000000	0.500000

الجدول (2) : يوضح حالة التلاحم والارتباط واعادة الاستخدام للأصناف في الشفرة الاولى .

أسم الصنف	حالة التلاحم للصنف	حالة الارتباط للصنف	حالة إعادة الاستخدام للصنف
Class1	High Cohesion	High Coupling	Bad Reusability
LinkedList	Medium Cohesion	Low Coupling	Good Reusability
Class2	High Cohesion	Low Coupling	High Reusability
LLIterator	Medium Cohesion	Medium Coupling	Bad Reusability
Second	High Cohesion	Low Coupling	High Reusability

الجدول (3) : يوضح فرص التحسين المقترحة على الاصناف في الشفرة الاولى لجعلها صالحة لاعادة الاستخدام.

أسم الصنف	حالة إعادة الاستخدام	مقترحات التحسين
Class1	Bad Reusability	Bad reuse because high coupling , To improve reuseability , We should make this class not inheritance and decrease number of shared variables (in this class 1 shared variables from 2) and decrease number of shared functions (in this class 1 shared functions from 1)
LinkedList	Good Reusability	To improve reusability , We should increase number of Shared Attributes inside class (in this class 1 shared attributes from 7)
Class2	High	No Suggestions because this class is high reusability

	Reusability	
LLIterator	Bad Reusability	To improve reusability , We should decrease number of Shared Variables (in this class 1 shared variable from 3) And make this class not inheritance
Second	High Reusability	No Suggestions because this class is high reusability

الجدول 4 يوضح قيم التلاحم والارتباط للاصناف في الشفرة الثانية ، والجدول 5 يوضح حالة التلاحم والارتباط واعادة الاستخدام لهذه الاصناف ، والجدول 6 يوضح مقترحات التحسين لهذه الاصناف .

الجدول (4) : يوضح قيم التلاحم والارتباط لأصناف الشفرة الثانية .

أسم الصنف	قيمة التلاحم للصنف	قيمة الارتباط للصنف
MyStack	1.800000	0.000000
Temp1	-0.333333	1.166667
Demo	1.250000	1.000000
Bicycle	1.500000	0.000000
MountainBike	1.200000	1.000000
RecursionExampleDirectory	1.038462	1.000000

Directory	0.000000	0.000000
File	1.000000	0.000000

الجدول (5) : يوضح حالة التلاحم والارتباط واعادة الاستخدام للأصناف في الشفرة الثانية .

أسم الصنف	حالة التلاحم للصنف	حالة الارتباط للصنف	حالة إعادة الاستخدام للصنف
MyStack	High Cohesion	Low Coupling	High Reusability
Temp1	Low Cohesion	Medium Coupling	Bad Reusability
Demo	High Cohesion	Medium Coupling	Good Reusability
Bicycle	High Cohesion	low Coupling	High Reusability
MountainBike	High Cohesion	Medium Coupling	Good Reusability
RecursionExampleDirectory	High Cohesion	Medium Coupling	Good Reusability
Directory	Low Cohesion	Low coupling	Bad Reusability
File	High Cohesion	Low Coupling	High Reusability

الجدول (6) : يوضح فرص التحسين المقترحة على الاصناف في الشفرة الثانية لجعلها صالحة لاعادة الاستخدام.

أسم الصنف	حالة إعادة الاستخدام	مقترحات التحسين
MyStack	High Reusability	No Suggestions because this class is high reusability

Temp1	Bad Reusability	Bad reuse because low cohesion , To improve reuseability , We should increase number of shared Attributes (in this class 0 shared Attributes from 1) and increase number of shared function inside the same class(in this class 1 shared functions from 6) and decrease number of Public variables (in this class 1 Public variables from 3)
Demo	Good Reusability	To improve reusability , We should make this class not inheritance
Bicycle	High Reusability	No Suggestions because this class is high reusability
MountainBike	Good Reusability	To improve reusability , We should make this class not inheritance
RecursionExampleDirectory	Good Reusability	To improve reusability , We should make this class not inheritance
Directory	Bad Reusability	Bad reuse because low cohesion , To improve reuseability , We should increase number of shared Attributes (in this class 0 shared Attributes from 2) and increase number of shared function inside the same class(in this class 0 shared functions from 2) and decrease number of Public variables (in this class 2 Public variables from 2)

File	High Reusability	No Suggestions because this class is high reusability

8- الاستنتاجات :

1. إعادة استخدام البرمجيات من أهم المقاييس في الوقت الحاضر ، إذ تختصر من كمية الوقت المطلوب في بناء نظام جديد وكذلك تقلل من الجهد المطلوب ، ومن ثم تقلل من الكلفة .

2. عند تطبيق نظام قاعدة المعرفة المقترح على أي شفرة جافا مكتوبة بطريقة قياسية ، يمكّننا النظام المقترح من التأكد من قياسية هذه الشفرة ، ومدى صلاحيتها لإعادة الاستخدام بأقل تعديلات ممكنة عليها .

3. عند تطبيق نظام قاعدة المعرفة المقترح على شفرة جافا غير صالحة لإعادة الاستخدام ، يتم عرض أسباب عدم صلاحية هذه الشفرة في إعادة الاستخدام ، وكذلك تقديم بعض المقترحات و النصائح لتحسين الشفرة و التي تجعلها صالحة لإعادة الاستخدام .

10-المصادر :

1. Aditi Dubey , Harleen Kaur , , December (2015) ,"**Reusability Types and Reuse Metrics: A Survey**", International Journal of Computer Applications (0975 – 8887) Volume 131 – No.2.
2. Ahmed Abd Elhalim Ibrahim, Amr Kamel, Hesham Hassan , (2016) ,"**Object Oriented Metrics and Quality Attributes: A Survey2016**", Proceedings of the 10th International Conference on Informatics and Systems,ISBN:978-1-4503-4062-5.

3. Ali Zeinulabdeen Alfakhry , Safwan Omar Hasoon PhD , (July 2018)," **Expert System for Software Reusability "** , International Journal of Computer Applications , Volume 181 – No.1 .
4. Ashwin B. , Tomar Siddhant , (2012) ,"**A Study of Sofware Reuse and Models**" , International Journal of Computer Applications® (IJCA).
5. Keith Darlington , Feb-12 (2007), "**A brief historical review of explanation in expert system applications**" , IASTED International Multi-Conference: artificial intelligence and applications PAGES 604-609.
6. Neha Sadana , July (2013), "**AN EMPIRICAL REVIEW ON FACTORS AFFECTING REUSABILITY OF PROGRAMS IN SOFTWARE ENGINEERING**", International Journal of Computing and Corporate Research ISSN (Online) : 2249-054X Volume 3 Issue 4 Manuscript ID : 2249054XV3I4072013-05.
7. Neha Sadana, Surender Dhaiya, Manjot Singh Ahuja , February (2014) ,"**A Metric for Assessing Reusability of Software Components** " , International Journal of Computer Application Issue 4, Volume 1.
8. Yogesh Kumar , Yogyata Jain , (2012)," **Research Aspects of Expert System** " ,International Journal of Computing & Business Research ISSN: 2229-6166 .