2016 IEEE 40th Annual Computer Software and Applications Conference

# A Methodology to Automate the Selection of Design Patterns

Shahid Hussain, Jacky Keung, Arif Ali Khan, Kwabena Ebo Bennin
Department of Computer Science
City University of Hong Kong
shussain7-c@my.cityu.edu.hk, Jacky.Keung@cityu.edu.hk, [aliakhan2-c, kebennin2-c]@ my.cityu.edu.hk

***Abstract*:**
**Background:** Over the last two decades, numerous software design patterns have been introduced and cataloged on the basis of developer's interest and skills. **Motivation:** In software design phase, inexperienced designers are mostly concerned on how to select an appropriate design pattern from the catalog of relevant patterns in order to solve a design problem. The existing automated design pattern selection methodologies are limited to the need of formal specification of design patterns or an appropriate sample size to make the learning process more effective. **Method:** To address this concern, we propose a three step methodology to automate the selection process of design pattern for a design problem. The steps of the methodology are text preprocessing, use of an unsupervised learning technique (that is Fuzzy c-Mean) as a core function to quantitatively determine the resemblance of different objects and selection of most appropriate pattern for a design problem. We evaluate our methodology with two samples that is Gang-of-Four (GoF) design pattern and spoiled pattern collection, and three object-oriented related design problems. Moreover, we used Fuzzy Silhouette test, Kappa (k) test, Cosine Similarity and argmax function to measure the effectiveness of our methodology. **Results:** In case of GoF pattern collection, we validated the reliability of Fuzzy c-Mean (FCM) results using a classification decision tree, and observed promising results compared to other automation techniques. **Conclusion:** From the comparison results, we observed 11%, 4% and 18% improvement in the performance of proposed technique as compared to supervised learning techniques of Support Vector Machine, Naïve Bayes and C4.5 respectively.

Keywords—Software Design Pattern; Gang-of-Four; Text Mining; Fuzzy c-means; Decision Tree; Classification; Selection.

## I. Introduction

The developers of object-oriented softwares adopted the idea of patterns from the mid-90s, however, Gamma et al [1] was the first to catalogue the 23 design patterns in order to overcome the object-oriented design problems and fulfill the needs of developers. These 23 design patterns are also known as GoF or Gang-of-Four (Gamma, Helms, Johnson and Vlisides) design patterns. A design pattern is a standardized form, which is organized through expert's knowledge and experience based on solutions available for a design problem, for example the GoF Composite design pattern, which presents the canonical solution for structural and compound problems. The examples of these problems comprises the objects, its nesting and building tree structures. In order to establish a trade-off between design and implementation of quality, consistency and reusability of developer software, design patterns can be considered as a catalyst [1,2]. Recently, in their study A. Ampatzoglou et al [3] summarized and reported the interest of the software engineering research community for GoF design patterns in both academia and industry. However, the selection and integration of design patterns for a design problem still remains the main concerns of a designer. To address these concerns, several studies [3,7,8,9] have facilitated the designer by introducing techniques for classification and selection of design patterns, while [11,12] studies help the designer to check the correct integration of a selected design pattern. In these studies, different approaches such as UML-based 15, 16, 17], linguistic-based [18, 19] and Ontology-based [23] are used to select and determine an appropriate pattern. However, our study is related to the selection process of design patterns. In his survey study, A. Birukou [4] discussed the two issues related to the selection of a right pattern through its description. The first issue is related to the availability of variety of pattern description which exists in books form such as Gang-of-four (GoF), Pattern Oriented Software Architecture (POSA) and Pattern Language Markup Language (PLML), and the second issue is related with accessing of patterns via the web due to unavailability of description on the internet. Similarly, In their study, Henninger and Correa [13] reported that 31% of software related patterns were not accessible via the web and are only available in book, reports or proceedings format. Moreover, from the last two decades, a lot of software design patterns with its variants (spoiled patterns) have been introduced and catalogued [6], which create difficulties for designers to select a right design pattern in order to solve a design problem without the support of any tool. This assumption is true specifically for novice or inexperienced designers. In order to address these issues, we proposed a three step methodology shown in Figure 1, to automate the classification and selection of an appropriate design pattern from the list of correlated patterns based on feature similarity with a design problem. These features are retrieved from the text

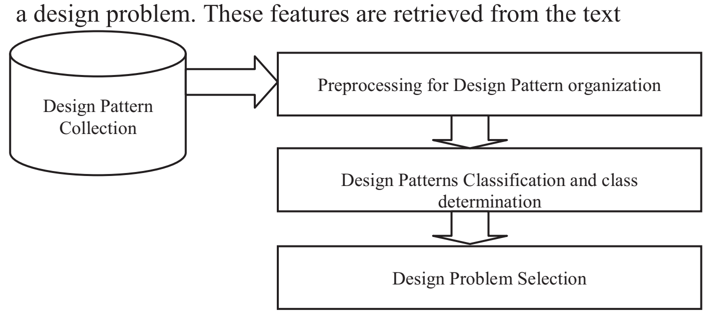


**Figure 1. Design Patterns Classification and Selection Process**

0730-3157/16 $31.00 © 2016 IEEE
DOI 10.1109/COMPSAC.2016.226

description of the design problem and problem definition of the design pattern collection. Three steps of the proposed methodology are text preprocessing for training unsupervised classifiers, grouping of design patterns besides design problem through Fuzzy c-Means (FCM), and selection of the most appropriate pattern for a design problem based on feature similarities computed through Cosine Similarity (CS) measure and argmax function.

## II. Related Work

We summarized the classification scheme into three groups, 1) problem based classification [1, 11, 20, 21], 2) solution based classification [14, 22], and 3) Text classification [5]. Moreover, we also summarized and grouped the design pattern formalization and selection approaches into UML based approaches [15, 16, 17], text based [5] and Ontology-based approaches [19, 23, 24]. In the studies of UML-base approaches, D-K Kim and C.E. Khawand present an approach to select the design pattern by specifying its problem domain. They reported that problem domain of design patterns can help to describe the problem context in which the pattern is defined. Moreover, the authors used Role-Based Modeling Language (RBML) as UML based pattern specification language. This language provides three types of specification in order to capture the functional perspectives of pattern properties such as Static Pattern Specifications (SPSs) to capture the structural properties, Interaction Pattern Specification (IPSs) to capture the interaction behaviors and State Machine Pattern Specification (SMPs) to capture the pattern's state-based behaviors. The level of reliability which is desired or problem specification depends on a number of problem models. Therefore, their approach does not focus on presenting highly reliable specification [15]. Similarly, N-L Hsuch et al [17] introduced a goal-driven and pattern-based approach in order to resolve design conflicts, to meet requirements and improve the design quality. The focus of their research is to overcome the object-oriented design challenges when design patterns are applied during the software development, non-functional requirements instead of only functional requirements. In their study, they used UML notation to describe and specify the design problems by incorporating the non-functional requirements instead of only functional requirements. In their study, they also discover and apply design patterns from four perspectives such as activity-view, requirement-view, problem-view and tradeoff-view and these are context dependent which is not true for spoiled patterns. In the studies of ontology-based approaches, E. Blomqvist [23] proposed and implemented a semi automatic ontology construction approach OntoCase which relies on ontology pattern for selection and ranking of patterns. The pattern selection process is based on the threshold of the ranking values retrieved from different approaches. However, there is a concern related to the flexibility of OntoCase. Similarly, for text classification, S.M.H Hasheminejad and S. Galileo used text classifiers to automate the design pattern selection process. They introduced two phase method to automate the selection of design pattern. The first phase of the proposed method is used in the selection of design patterns while the second phase is used to determine the right pattern. In order to evaluate their approach, they took samples from Douglass and GoF pattern catalogs [5]. They used text classifiers in order to select and rank the design patterns. Moreover the automation decision of classifiers depends on the training samples and also for each design pattern class there is a need of a separate classifier for the learning process. In case of linguistic-based approaches, S. Hasso and C.R Carlso [18, 19] introduced a new methodology to organize the design patterns and to make an easy identification process. Their methodology was based on complex algebraic structures to describe the design patterns and used reliable theories for classification. They examined 76 design patterns retrieved from different catalogs. Moreover, they also demonstrated their methodology through a case study comprising of 13 design patterns which were retrieved from the catalog of the Rising's Pattern Almanac. However, their reference domain was limited to linguistic bases and not to software engineering or any business-centers such as telecommunication, accounting, database or real-time systems. Similarly, S. Peasson and Y. Shen [10] proposes a decision based support which can assess the user related contextual factors such as sensitivity of data, location of data, sector, contractual restrictions and some other factors, and deduces a list of recommendations and controls There are a few common points among these classification schemes and approaches. The first, is the selection of more appropriate

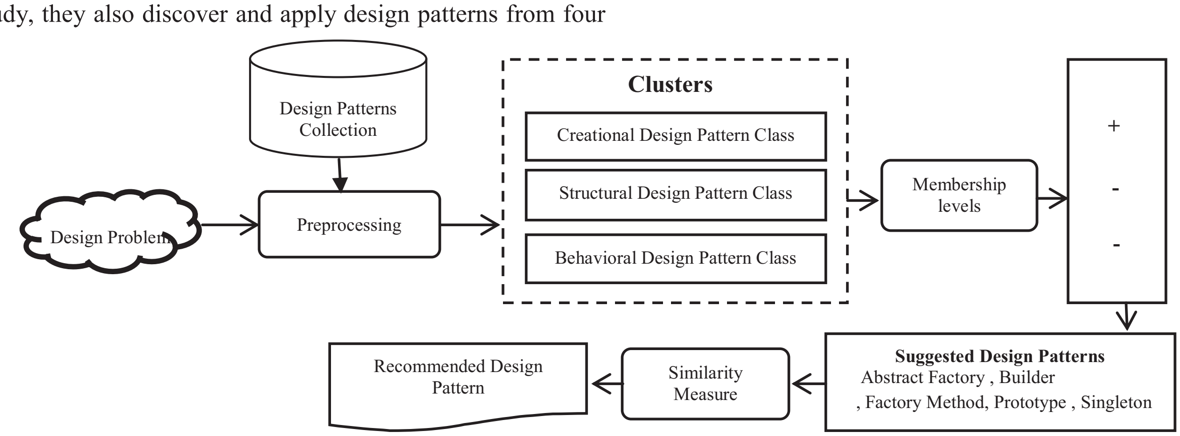


**Figure 2. Design Pattern Slection Process for a Design Problem**

design pattern either through precise formalization of the problem or analyzing the solutions of design patterns. The second, is the high cost of production of meta-model in UML-based approaches or cost of implementation and automation process for ontology-based or linguistic-based approaches. In case of existence of complex catalogs or a catalog with numerous variants in the form of spoiled patterns, an inexperienced designer will struggle most in selecting a right pattern for a design problem.

## III. PROPOSED METHODOLOGY

The aim of our study is to automate the classification and selection process of patterns for design problems. Our proposed methodology is based on an unsupervised learning technique as a core function, to learn the classification on the sample of the design pattern collection with a design problem, and recommend a right pattern to the inexperienced designer in order to solve the corresponding design problem. The detail of the steps of the proposed methodology (Figure 1) is given as.

### *A. Preprocessing*

The first step of our methodology is to perform preprocessing, since learning techniques either supervised or unsupervised can not be directly applied to text. The set of activities which are performed in the preprocessing process are shown in Figure 3. The first activity of preprocessing is related to the description of a design problem and problem definition of patterns besides its implications. The description of the design problem and the design pattern collection are considered as input documents for text preprocessing process (shown in Figure 2).

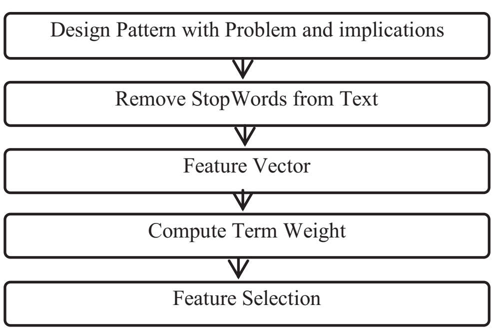


**Figure 3. Preprocessing activities**

The second activity is related to the removal of stop words (e.g. Preposition, articles, and etc.) and words stemming to build a basic form of words. Several text mining tools like JPreText (A simple Text Preprocessing Tool) [27] are available to perform these tasks. The third activity is related to the use of feature vector methods in order to present each document without using explicit semantic information. A feature vector is documented for the description of each design pattern and a design problem. The fourth activity is related to computing the frequency of any word in a document to improve the performance and remove noise from the documents [25]. There are many techniques related to reduce the dimensionality of data such as principal component analysis (PCA), linear discriminant analysis (LDA) and so on with different pros and cons, however, in the last activity, we use of Document Frequency (DF) feature selection method in order to reduce the dimensions of feature vectors [26]. Each feature vector is normalized in N features and includes highly weighted words of all documents. The existence of a word for a document is interpreted in binary mode that is -1 or 1.

### *B. Design Pattern Classification and Class Determination*

In the second step of our methodology, we perform two tasks that are 1) classification of design patterns and 2) determination of most appropriate design pattern class for a design problem. In order to perform these two tasks, we need to use an unsupervised learner to achieve the objective of our study. There are numerous unsupervised learning algorithms such as hierarchical, k-means, Bi-Section-k-means and Self Organizing Map (SOM) that can be used in the automation process. However, in our study, we used Fuzzy c-means (FCM) [29] clustering technique as a core function to learn for each design pattern and a design problem, and classified them into groups on the basis of their correlated features. The main reason behind the use of Fuzzy c-Means unsupervised learning technique is its functionality to assign class membership value to each object and group them accordingly. In software design phase, the class membership value for each design pattern can help the inexperienced designer in the classification of design patterns and determination of most appropriate design pattern class for a design problem. We used Fuzzy Silhouette Coefficient to judge the quality of clustering and optimal number of clusters.

TABLE I. INTERPRETATION OF SILHOUETTE COEFFICIENT

| Silhouette Coefficient | Interpretation |
|---|---|
| 0.70-1.00 | Cluster separation is excellent |
| 0.50-0.70 | Cluster separation is clear |
| 0.25-0.50 | Cluster separation is with some noise |
| 0.00<0.25 | Cluster separation is not possible |

The list of Fuzzy Silhouette coefficients with its interpretation is shown in Table I. The inputs of our methodology (shown in Figure 2) are a design pattern collection and a design problem. The unsupervised learner that is FCM takes design problem and design pattern collection as input documents in preprocessed form, and makes the learning process for classification of design patterns into groups, and determine the design pattern class for corresponding design problem. Moreover, we consider a GoF design pattern collection and a design problem as input for FCM with cluster size that is c=3 (Figure 2), since E. Gamma et al [1] have divided the design patterns into three classes creational, structural and behavioral. The positive or negative symbols corresponding to each design pattern class depend on the class membership value of a design problem for that class. Moreover, these positive and negative symbols also present the high or low association of design problem with design pattern classes. In Figure 2, the positive symbol for the creational design pattern class presents its high association with a design problem. At the end of this step, a list of suggested patterns from a design pattern class are retrieved. The classification results of our methodology with three design problems and sample of the GoF design patten collection is shown in Table II. The high class membership degree value of a design problem for a design pattern class presents its high association for that class. For example, the class membership value 0.71 of design problem 2 (DP-2) presents its high association with the creational design pattern class. Moreover, in Figure 2, the list of suggested design patterns for a design problem 2 are retrieved from the creational design pattern class, which are Abstract Factory, Builder, Factory Method, Prototype and Singleton. The description of design problems is given in section VI.

### C. Design Pattern Selection

In case of unsupervised learning process, there are different measures which can help to find the similarity between target objects and a query. In our study, the term query refers to a design problem while target objects refer to suggested design patterns. The similarity measures reflect the degree of closeness or separation of a query with its corresponding target objects. Generally, a similarity measure is used to map the distance or similarity between the symbolic description of two objects which are either described in numeric value or in vector form. The most common similarity measures are Euclidean distance, Jaccard Coefficient (also know as Tanimoto Coefficient), Peasrson Correlation Coefficient, however, we used Cosine Similarity (CS) measure to compute the similarity of text documents [30].

In our study, these documents are descriptions of **Pattern** and a **Problem** as shown in equation 1. The Cosine Similarity (CS) measure, which is described in equation 1 evaluates the similarity between a vector **Pattern** i and a query vector **Problem**, and computes their similarity on the basis of the inner product of the both vectors. Moreover, in equation 1, it is assumed that documents of CS measure such as Pattern and Problem are defined by a vector of term weights shown in the equation 2. Subsequently, in equation 2, each term w (d, $t_j$) presents a weight for each word j of the design patterns and problem.

$$CS_i = \sum_{j=1}^{N} w(\text{Pattern}_i, t_j) \times w(\text{Problem}, t_j) \qquad (1)$$

$$W(d) = (w(d,t_1), w(d,t_2), ..., w(d,t_n)) \qquad (2)$$

The size of the vector N is defined by the number of words of the complete design patterns group after removing stop words and word stemming. After measuring the similarity of the design problem and problem definition of design patterns, arguments of the maximum (that is arg max/argmax) is applied to select the most appropriate design pattern. Instead of global maximums, we used argmax because it takes inputs in order to return maximum output. The description of argmax for the selection of right design pattern is shown in equation 3. Where the term RP refers to a right design pattern and $CS_i$ refer to the cosine similarity of design pattern i for a design problem. The similarity values for all suggested design patterns for a design problem are considered as as input of argmax and it return the right design pattern RP with the highest similarity value.

$$DP = \arg\max_i CS_i \qquad (3)$$

Where the design pattern RP with highest similarity values are recommended for an inexperienced designer in order to solve a design problem.

## IV. Evaluation Procedure of Proposed Methodology

In section V, we have described our methodology and its implementation procedure. However, in this section we are discussing the evaluation of two important tasks of our methodology that is 1) classification of design patterns and determination of a pattern class for a design problem and 2) selection of right design pattern through FCM. In this evaluation process, we considered three design problems and two samples of design pattern collection. The first sample includes the 23 design patterns from the Gang-of-Four (GoF) collection, and second sample includes spoil patterns presented by Bouhours et al [6]. The description of real design problems [24] which we considered in the evaluation process are described as follows.

- **Design Problem (DP-1).** A socket attaches to a ratchet, provided that the size of the drive is the same. Typical drive sizes in the United States are 1/2" and 1/4". The rest is described in [24].
- **Design Problem (DP-2).** The manufacturing process of Japanese automobiles is required to stamp the equipments with metal sheet. The same machinery is used to stamp right hand doors, left hand doors, right front fenders, left front fenders, hoods, etc. The rest is described in [24].
- **Design Problem (DP-3).** Children's meals typically consist of a main item, a side item, a drink, and a toy (e.g., a hamburger, fries, Coke, and toy dinosaur). Note that there can be variation in the content of the children's meal, but the construction process is the same. Whether a customer order a hamburger, cheeseburger, or chicken, the process is the same. The rest is described in [24].

### A. Evaluation of Unsupervised learning Process

We evaluate the unsupervised learning process of FCM in order to verify the appropriate classification of design patterns into groups and determination of class for a design problem. We used two measures Fuzzy Silhouette and Kappa (k) coefficients to evaluate the classification performance of Fuzzy c-Mean (FCM) technique. In case of a sample of GoF design pattern collection,

TABLE II. CLASS MEMBERSHIP DEGREE VALUES FOR PROBLEMS

| Design Problem | Class Membership Degree Value | | |
|---|---|---|---|
| | Creational Design Pattern Class | Structural Design Pattern Class | Behavioral Design Pattern Class |
| DP-1 | 0.18 | 0.51 | 0.31 |
| DP-2 | 0.71 | 0.18 | 0.11 |
| DP-3 | 0.74 | 0.18 | 0.08 |

we found a fuzzy silhouette coefficient of 0.29, which shows that the data is clustered with some noise which might be due to duplication of values or null values [25]. Moreover, in case of spoiled pattern collection, we found a fuzzy silhouette coefficient of 0.28. The second measure, Kappa (k) is used to estimate the agreement in categorical data. Like other measures such as Precision, Recall, and F-measure, Kappa is also used to extract the information from the confusion matrix of an unsupervised learner and evaluate its effectiveness for classification [29]. We found k=0. 55 and k=0. 53 for a sample of GoF and spoiled design patterns respectively, which refers to moderate agreement. The results of Fuzzy Silhouette and Kappa (k) coefficients are in the support of applicability of FCM to support the classification and determination of design patterns.

TABLE III. COSINE SIMILARITY FOR DESIGN PROBLEM-1

| Structural Design Pattern | Cosine Similarity (CS) |
|---|---|
| Adapter | 0.69 |
| Bridge | 0.48 |
| Composite | 0.18 |
| Decorator | 0.35 |
| Façade | 0.61 |
| Flyweight | 0.11 |
| Proxy | 0.05 |

### B. Evaluation of Design Pattern Selection Process

In this section, we evaluate the selection process of our methodology in order to verify how a right design pattern is selected for a design problem. This is decided on the closeness or similarity between the description of a design problem and problem definition of design patterns.

TABLE IV. COSINE SIMILARITY FOR DESIGN PROBLEM-2

| Creational Design Pattern | Cosine Similarity (CS) |
|---|---|
| Abstract Factory | 0.76 |
| Builder | 0.25 |
| Factory Method | 0.41 |
| Prototype | 0.39 |
| Singleton | 0.58 |

The evaluation process for the selection of right design pattern is performed in two steps, 1) we use Cosine Similarity (CS) measure to compute the closeness between descriptions of the design problem and problem definition of suggested design patterns, and 2) we apply argmax function to recommend a right design pattern with highest similarity value between two documents. The cosine similarity between suggested patterns and design problems is shown in Table III, Table IV and Table V respectively.

TABLE V. COSINE SIMILARITY FOR DESIGN PROBLMEM-3

| Creational Design Pattern | Cosine Similarity (CS) |
|---|---|
| Abstract Factory | 0.49 |
| Builder | 0.74 |
| Factory Method | 0.34 |
| Prototype | 0.57 |
| Singleton | 0.70 |

## V. VALIDATION OF CLASSIFICATION RESULTS OF FCM

In this study, we used unsupervised learning technique that is Fuzzy c-mean (FCM) to automate the classification of design patterns of a collection. In order to validate the results of an automated classification process of our methodology, we construct and implement a decision tree which can be used for clustering of labeled data. Firstly, we labeled the FCM classification results and secondly, we implement a classification decision tree in order to validate the effectiveness of FCM results. In case of GoF design pattern group, we used three labels for creational, structural and behavioral class and implemented the decision tree. We used precision, F-measure and Receiver Operating Characteristics Area Under Curve (ROC-AUC) measures to evaluate the effectiveness of classification results. The values of precision, F-measure and ROC-AUC measures for class-wise classification performance are shown in Table VI, which are better than the evaluation results of supervised learners for GoF pattern collection [5]. For example, the F-measure value of supervised learners such as for Support Vector Machine (0.73), Naïve Bayes (0.80) and C4.5 (0.71). Moreover, we observed 82%, 83% and 87% values for the performance measure precision, F-measure and ROC-AUC respectively.

## VI. THREATS TO VALIDITY

In our study, we also find some threats. The first threat is relevant to the number of design pattern collection and design problems which were considered during the evaluation process of our proposed methodology. In order to evaluate the significance of results of proposed methodology, we will consider more design pattern collection (such as a Douglass design patterns group) and more examples of design problems. The second threat is about the use of a tool to extract the features from the description of the design problem and problem definition of design patterns because removing and stemming words strategies for text mining algorithms varied. A more appropriate extraction of features can help to enhance the significance of the proposed methodology in order to determine the appropriate design pattern.

## VII. CONCLUSION AND FUTURE WORK

The existence of numerous design patterns besides its variants in the form of spoiled pattern, reflect the concerns of an inexperienced designer about the selection of a right design pattern from the list of correlated patterns for a design problem. The existing automated design pattern selection methodologies are limited to the need of formal specification of design patterns or an appropriate sample size to make the learning process more effective. In order to address these issues, we proposed a three step methodology to automate the selection process of an appropriate design pattern. The proposed methodology was based on an unsupervised learning technique that is Fuzzy c-Means (FCM) as a core function and aimed at presenting an appropriate way of recommending the most suitable design pattern. We evaluated our methodology with three design problems and two samples that is Gang-of-Four (GoF) design pattern and spoiled pattern collection. We found the Fuzzy Silhouette (0.29) and Kappa(k) coefficient (0.55) for FCM in case of GoF design pattern collection to judge

TABLE VI. PERFORMANCE MEASURE RESULTS

| Design Pattern Class | Precision (%) | F-measure (%) | ROC-AUC (%) |
|---|---|---|---|
| Creational | 85 | 92 | 94 |
| Structural | 77 | 82 | 90 |
| Behavioral | 85 | 75 | 77 |

the quality. We validated the reliability of the results of FCM using a classification decision tree. The value of Precision (82%), F-measure (83%) and Receiver Operating Characteristics Area Under Curve (87%) performance measures shows that the core function of our methodology that is FCM outperforms other automated supervised classification functions such as the Precision (0.75) and F-measure(0.80) for Naïve Bayes with GoF pattern collection. We will consider more design problems and design pattern collection (such as Douglass and Booch pattern group) to verify and generalized the applicability of our methodology.

## VIII. ACKNOWLEDGEMENT

This research is supported by the City University of Hong Kong research funds (Project No. 7200354, 7004222, 7004474).

## IX. REFERENCES

[1] E. Gamma, R. Helm, R. Johnson and J. Vlissides, Design Patterns. Boston, MA: Addison-Wesley, 1995.

[2] H. Kampffmeyer and S. Zschaler, Finding the pattern you need: The design pattern intent ontology, Proceedings of the 10th International Conference on Model Driven Engineering Languages and Systems (MoDELS), lecture notes in computer science,vol 4735/2007, p. 211–225, 2007.

[3] A. Ampatzoglou et al, Research State of the art on GoF Design Patterns: A mapping study, Journal of System and Software, 86, p. 1945– 1964, 2013.

[4] A. Birukou, A Survey of Existing Approaches for Pattern Search and Selection, Proceeding of PLoP, 2010.

[5] S. M. H Hasheminejad and S. Jalili, Design Patterns selection: An automatic two-phase method, Journal of System and Software, 85, p. 408-424, 2012.

[6] C. Bouhours, H. Leblance and C. Percebois, Spoiled Patterns: How to extend the GoF, Software Quality Journal, 23, p. 661-694, 2015.

[7] H. Albin-Amiot, P. Cointe, Y. G. Gue and N. Jussien, Instantiating and Detecting Design Patterns: Putting bits and pieces together. Proceedings of the 16th Conference on Automated Software Engineering (ASE), p. 166–173, 2001.

[8] J. Dong, and Y. Zhao, Classification of Design Pattern Traits. Proceedings of the 19th International Conference on Software Engineering and Knowledge Engineering (SEKE), p. 473–477, 2007.

[9] J. K. H Mak, C. S. T Choy and D. P. K Lun, Precise Modeling of Design Patterns in UML. Proceedings of the 26th International Conference on Software Engineering (ICSE), p. 252–261, 2004.

[10] S. Pearson and Y. Shen, Context-Aware Privacy Design Pattern Selection, LNCS 6264, pp. 69–80, 2010.

[11] H. Mili and G. El-Boussaidi, Representing and Applying Design Patterns: What is the Problem, Proceedings of the 8th International Conference on Model Driven Engineering Languages and Systems (MoDELS), p. 186–200, 2005.

[12] M. O. Cinneide and P. Nixon, A Methodology for the Automated Introduction of Design Patterns. Proceedings of the 15th IEEE International Conference on Software Maintenance (ICSM), p. 463–473, 1999.

[13] S. Henninger and V. Correa, Software Pattern Communities: Current Practices and Challenges. Proceedings of the 14th Conference on PLoP. 2007.

[14] G. J. Kim and J.S Han, Clustering algorithm of Design Pattern using Object-Oriented Relationship, Computational Science and Its Applications ICCSA, p. 997‑1006, 2007.

[15] D. K. Kim and C.E. Khawand, An Approach to Precisely Specifying the Problem Domain of Design Patterns, Journal of Visual Languages and Computing, 18, p. 560‑591, 2007.

[16] D. K. Kim, W. Shen, Evaluating Pattern Conformance of UML Models: A divide and conquer approach and case studies. Software Quality Journal, 16 (3), p. 329‑359, 2008.

[17] N. L. Hsueh, J-Y. Kuo and C-C. Lin, Object-Oriented Design: A Goal-driven and Pattern-based approach. Journal for Software and Systems Modeling, 8 (1), p.1-18, 2007.

[18] S. Hasso, C. R. Carlson, Linguistics-based Software Design Patterns Classification, Proceedings of 37th Annual Hawaii International Conference on System Science (HICSS-37), 2004.

[19] S. Hasso and C. R. Carlson, A Theoretically-based Process for organizing Design Patterns. Proceedings of 12th Pattern Language of Patterns, 2005.

[20] B. P Douglass, Real-Time Design Patterns: Robust Scalable Architecture For Real-Time Systems. Addison-Wesley/Longman Publishing Co., Inc., Boston, MA, USA, 2002.

[21] K. E. Bennin et al, End User in Charge - Social Framework for Open Source Development. Proceeding of 2nd International Conference on Advances and Trends in Software Engineering, IARIA, 2016

[22] W. Zimmer, Relationships between Design Patterns. Journal of Pattern Languages of Program Design, 1, p. 345‑364, 1995.

[23] E.Blomqvist, Pattern Ranking for Semiautomatic Ontology Construction, Proceedings of SAC, 2008.

[24] A. Shalloway and R. Trott, Design Pattern Explained: A new Perpective on Object Oriented Design, Addison Wesley, 2001.

[25] A. Hotho, A. Nurnberger and G. Paab, A Brief Survey of Text Mining, Journal for Computational Linguistics and Language Technology, 20, p. 19‑62, 2005.

[26] H. Joho and M. Sanderson, Document frequency and term specificity, Proceeding of RIAO, 2007.

[27] M. Ricardo et al, PreText: A Simple Text Preprocessing Tool, http://sites.labic.icmc.usp.br/torch/msd2011/jpretext/

[28] J. C. Bezdek, R. Ehrkich and W. Full, FCM: The Fuzzy c-Means Clustering algorithm, Journal of Computers and Geosciences vol. 10, No 2-3, p. 191-203, 1984.

[29] J. Carletta, Assessing agreement on classification tasks: the kappa statistic, Computational Linguistics, v. 22, 1996.

[30] A. Huang, Similarity Measures for Text Document Clustering, Proceedings of NZCSRSC, 2008.