A Structural Complexity Measure for UML Class Diagrams*

Baowen Xu[1,2], Dazhou Kang[1], and Jianjiang Lu[1,2,3]

[1] Department of Computer Science and Engineering, Southeast University, Nanjing, 210096, China
[2] Jiangsu Institute of Software Quality, Nanjing, 210096, China
[3] PLA University of Science and Technology, Nanjing, 210007, China
bwxu@seu.edu.cn

Abstract. UML class diagrams constitute a key artifact in the conceptual modeling phase and their quality can have a significant impact on the quality of the system. The structural complexity measure is one of the most important measures to evaluate the quality of a UML class diagram. This paper uses weighted class dependence graphs to represent a given class diagrams, and then presents a structure complexity measure for the UML class diagrams based on entropy distance. It considers complexity of both classes and relationships between the classes, and presents rules for transforming complexity value of classes and different kinds of relations into a weighted class dependence graphs. This method of measure has many good properties; therefore it can measure the structure complexity of class diagrams objectively.

1 Introduction

One of the principal goals of software engineering is to assure the quality of object oriented software from the early phases of the life-cycle, such as conceptual modeling phase. UML [1] class diagrams constitute a key artifact in this phase. The structural complexity measure is one of the most important metrics to evaluate the quality of a UML class diagram [2].

Chidamber and Kemerer proposed a set of design metrics defined at class level [3]. Lorenz and Kidd proposed a group of metrics deal with the static characteristics of software design [4]. Brito, Abreu and Melo proposed a set of metrics at system level [5]. Marhchesi proposed a set of metrics to measure UML class diagrams at the analysis phase, but did not take into account some UML measurable elements [2]. Genero proposed new metrics to cover the necessity of measuring these relationships [6]. Manso and Genero used 8 metrics for measuring the structural complexity and the size of UML class diagrams and their maintainability [7]. But they did not give a single complexity measuring integrate all these metrics.

* This work was supported in part by the Young Scientist's Fund of NSFC (60373066, 60303024), National Grand Fundamental Research 973 Program of China (2002CB312000), National Re-search Foundation for the Doctoral Program of Higher Education of China.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3036, pp. 421–424, 2004.

2 Weighted Class Dependence Graphs

2.1 Complexity Measure for Classes and Relations

Classes and relations are the basic elements of class diagrams. We use suitable metrics proposed by others to measure the complexity of classes which satisfied that: only one metric for measuring each class; the value is above zero and denotes the complexity of the class; it take both the class structure and inheritance into account. The cohesion measure sometimes can be used here.

There are mainly three kinds of relations in UML class diagrams: associations, generalizations, and dependencies. Complexity of different kinds of relations can be compared. Different kinds of relations influence the dependency between classes in different degrees. We fractionize relations in UML class graph into totally 10 kinds and their different influences on the dependency between classes can be weighted. It forms Table 1.

Table 1. Dependency weight value of relations

No.	Relation	Weight
1	Common Dependency	H1
2	Common association	H2
3	Qualified association	H3
4	Association class	H4
5	Aggregation association	H5
6	Composition association	H6
7	Generalization (parent class is concrete)	H7
8	Binding	H8
9	Generalization (parent class is abstract)	H9
10	Realize	H10

Comparing the complexity between these kinds of relations, it has H1 ≤ H2 ≤ H3 ≤ H4 ≤ H6 and H1 < H5 < H6 < H7 < H8 < H9 < H10.

Each ends of a relation should link to a certain class. So relations of the same kind may have different complexity because they related to different classes. This can be calculated in the Weighted Class Dependence Graphs (WCDG) which denote the given class diagram abstractly.

2.2 Weighted Class Dependence Graphs

Definition 1. *D* denotes a given class diagram. WCDG is defined as *G(D) = (N, E),* where *N = V(D), E = R(D),* i.e. the nodes and edges. $V(D) = \{c \mid c$ is a class in $D\}$, $R(D) = \{(n_1, n_2, W(n_1, n_2)) \mid n_1, n_2 \in V(D) \wedge$ (there are relations in T from n_1 to n_2 or $n_1 = n_2)\}$. When $(n_1 \neq n_2), W(n_1,n_2) = \sum W_i$, W_i is the dependence weighted value of each relation from n_1 to n_2; when $(n_1 = n_2)$, $W(n_1, n_2)$ should be added H1 * C(n_1), C(n_1) is the complexity of the class n_1 denotes.

Every node in the WCDG is corresponding to a class in the class diagram; relations are transformed to edges between the nodes. Firstly we form the structure of the WCDG from the class diagram. Then calculate the dependence weight value of all the relations in the class diagram.

Let the complexity measure of class A and B are C(A) and C(B), the weight value of dependencies is H. The dependence weighted value W of a relation from B to A can be calculated as follows: when the relation has no destination multiplicity , such as dependency including Generalization, Binding and Realize, W = H*C(A); when the relation has a destination multiplicity of n, such as association including Aggregation and Composition, W = H*(2–1/n)*C(A), if n is *, let 1/n = 0; when it is a qualified association, W = H3*(2–1/n)*C(B)+b, b denotes the complexity of the qualifier; association class is transformed to a new node has relations with A and B.

Now we can calculate the weight value of the edges and nodes by the dependence weight value of relations. The WDCG can also be expressed in a matrix, in which W[i][j] = W(n_i, n_j). It will predigest the calculation of complexity.

3 A Structure Complexity Measure Based on Entropy Distance

X and Y are discrete stochastic variables: $A_x = \{ x_i \mid 1 \le i \le m\}$, $A_y = \{ y_j \mid 1 \le j \le n\}$.

The entropy of their joint distribution is:

$$H(X,Y) = -\sum_{x_i \in A_x \wedge y_j \in A_y} p(x_i, y_j) \log p(x_i, y_j) \tag{1}$$

The entropy of X when Y happened:

$$H(X \mid Y) = \sum_{x_i \in A_x \wedge y_j \in A_y} P(x_i, y_j) \log \frac{1}{P(x_i \mid y_j)} \tag{2}$$

The mutual-information of X and Y:

$$I(X,Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X) \tag{3}$$

Let D be a given class diagram; G(D) is the WCDG corresponding to D; N(D) is the set of all nodes in the WCDG.

We can use the entropy distance to measure the complexity of G(D). Use stochastic variables X and Y to denote the output and input edges weight of each node. Let $A_x = A_y = N(D)$, for each $x_i \in A_x$, each $y_j \in A_y$, we have

$$p(x_i) = \frac{\sum_{n2 \in N(D)} W(x_i, n2)}{\sum_{n1 \in N(D)} \sum_{n2 \in N(D)} W(n1, n2)}, \quad p(y_j) = \frac{\sum_{n1 \in N(D)} W(n1, y_j)}{\sum_{n1 \in N(D)} \sum_{n2 \in N(D)} W(n1, n2)} \tag{4}$$

$$p_{X,Y}(x_i, y_j) = \frac{W(x_i, y_j)}{\sum_{n1 \in N(D)} \sum_{n2 \in N(D)} W(n1, n2)}, \quad p_{X,Y}(x_i \mid y_j) = \frac{W(x_i, y_j)}{\sum_{n1 \in N(D)} W(n1, y_j)} \tag{5}$$

Definition 2. The complexity of D is defined to be the entropy distance of X and Y:

$$\text{Complexity(D)}= DH(X,Y) = H(X,Y) - I(X,Y) \tag{6}$$

Specially, when D=ϕ , Complexity(D)=0. For any class diagram D, it has that:

$$0 \le \mathit{Complexity}(\mathrm{D}) \le 2\log|\mathrm{V(D)}| \tag{7}$$

where V(D) is the set of classes in class diagram D.

We use Weyuker's properties [8] for complexity measures to evaluate our measure method. It contains 9 properties. We can prove that our measure satisfies 7 of the total 9 properties. We think our method can measure the structure complexity of class diagrams objectively.

4 Conclusions

We uses weighted class dependence graphs to represent a given class diagrams, and then present a structure complexity measure for UML class diagrams based on entropy distance. It considers complexity of both classes and relationships between the classes. This method of measure has many good properties; therefore it can measure the structure complexity of class diagrams objectively.

The UML class diagrams can only represent static model of the software. When dealing with dynamic knowledge, UML dynamic diagrams and state diagrams should be used. How to measure the quality of these diagrams is the future work.

References

1. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Maual. Addison-Wesley, Reading, MA, USA (1999)
2. Marchesi, M.: OOA metrics for the United Modeling Languages. Proceedings of 2nd Euromicro Conference on Software Maintenance and Reengineering. Palazzo degli Affari, Italy (1998) 67-73
3. Chidamber, S., Kemerer, C.: A Metrics Suite for Object Oriented Design. IEEE Transactions on Software Engineering, 20(6). (1994) 476-493
4. Lorenz., M., Kidd, J.: Object-Oriented Software Metrics: A Practical Guide. Prentice Hall, Englewood Cliffs, New Jersey (1994)
5. Brito, E., Abreu, F., Melo, W.: Evaluating the Impact of Object-Oriented Design on Software Quality. Proceedings of 3rd International Metric Symposium (1996) 90-99
6. Genero, M., Piattini, M.: Empirical validation of measures for class diagram structural complexity through controlled experiments. Proceedings of 5th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering. Budapest, Hungary (2001) 87-95
7. Manso1, M.E., Genero, M., Piattini, M.: No-Redundant Metrics for UML Class Diagram Structural Complexity. CAiSE 2003 The 15th Conference On Advanced Information Systems Engineering, LNCS 2681, 127-142
8. Weyuker, E.J.: Evaluating Software Complexity Measures. IEEE Transaction on Software Engineering. (1988) 1357-1365