ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 14, No. 4, 2019, *pp*.272-278

Ordered Metric Methods of Classes Dependency Graph Based on Structure Entropy

Aihua Gu [1+], Jinxia Xu [1], Lu Li [1], Shujun Li [1], Qifeng Xun [1], Jian Dong [1] and Zonglin Guo [2]

[1] *School of Information Engineering, Yancheng Teachers University, Yancheng 224002, China*
[2] *Department of Medical Instrumen,Yancheng Third People's Hospital, Yancheng 224001, China*
(Received August 01 2019, accepted September 26 2019)

Abstract. In this paper, structure entropy which is based on classes dependency graph of a software system is proposed to measure the complexity of the system. In order to research the structure entropy metrics of classes dependency graph in some large software systems, this paper takes the first step to mathematically prove that structure entropy does not have the property of cohesion. Thus, a structure entropy can be used as a single metric of complexity. And then, a program is written based on relevant matrix algorithm with the construction of classes dependency graph. The corresponding metrics of structure entropy of three pieces of open source software are calculated and figured out based on classes dependency graph. The calculation shows that most classes in the three pieces of the open source software are of randomness. Meanwhile, values of structure entropy features complex network Scale-free. Therefore, the different values of the structure entropy of open source software classes dependency graph influences the evaluation of software quality. Furthermore, some complex network statistical characteristics are found out in this paper, which will facilitate the further research on structure entropy as a metric of software complexity for sophisticated networks.

Keywords: object-oriented class, software quality, structure entropy, complex networks, software metrics

1. Introduction

A butterfly flapping its wings in tropical rain forests along Amazon River caused a tornado in an America state. This is the sentence from American meteorologist Lorentz who described the chaotic oscillator in 60s of 20th century approachably and vividly. Chaotic state can be regarded as a kind of Orderly Disorder. To combine shallow level disorder with internal regularity has called the attention of researchers. From the perspective of systematism, order is the structure form of things, that is, relationship between components of systems. Order is that whole structure of systems show regularity over time. For systematical projects, ordered structure benefits function-controllable and manageable, and facilitates system components to swap and coordinate. Therefore, Subscription services or functions can be implemented.

Software system structure has the characteristics of complex network scale-free. Heterogeneity of scale-free network represents a certain function expressing metric distribution indices of nodes. However, it may be not appropriate to use metric distribution index directly. If there are many edges in a network but very small metric distribution index, the metric of most nodes is also very large, that is, Heterogeneity becomes very little. Conversely, if there are few edges in a network but very large metric distribution index, then the metric of each node is very small. However, here heterogeneity is still very little. The inseparably link of network structure entropy and system complexity can indicate the heterogeneity degree of the network and represent the homogeneity of network nodes metric values. The heterogeneity of star network is unexpectedly the largest. Because there is only one node with very large metric in star network, in common situations, the metric values of other nodes are generally the same, so the network heterogeneity is not so strong [1].

How to represent and measure the ordered of such structure of software systems becomes the fundamental of research on complexity in science and systematic engineering relevant domains. In this paper, structure entropy is defined in terms of node degree of classes dependency graphs of software systems. Structure ordered degree of software systems is also analyzed. It provides reliable basis for optimizing overall configuration of classes in software systems.

+ Corresponding author: Aihua Gu
E-mail address: guaihua1978@163.com

A few researchers have evaluated the ordered of configuration structure and complexity basing on Shannon's information entropy defining system structure entropy, which offers solid basis for structure optimization and improvement. With the development of complex network researches, the structure entropy has been used to explore evolution rules of Internet topology structure and an order parameter which influences network topology structure evolution. These researches result that degrees of nodes in two topo structures in domain level and router level follows power law distribution [2].

In the field of software engineering, some researchers have analyzed UML graphs, and proposed entropy-based structure complexity metric method. Judging whether the system is uncertain relying on entropy can find out the system structure complexity. Zhou and his team applied entropy distance to analyze the complexity of UML graphs, when he found the relation between classes of a software system and assigned weight to edges according to the importance of classes' relation[3]. Xu and his team expressed UML graphs with weighted classes and classes dependency graphs, and judge the complexity of UML graphs according to the entropies distances[4]. Yi and his team who applied the methods proposed by Xu and Zhou verified the efficiency of the methods depending on the analysis on 27 systems[5]. Ma and his team theoretically verified the efficiency of evaluation coefficient in the metric structure order by simulating entropy decreasing process mathematically[1]. They quantitatively analyzed the order of software structures in terms of structure ordered metric computing algorism on ten different software systems and two network models. They revealed that there exists relationship between linking metric evaluation coefficient and structure entropy. With evaluation coefficient increasing, structure entropy tends decreasing. That means structure ordered degree is creasing, robustness is strengthening, systems' complexities are increasing during the period of entropy decreasing. The conclusion above offered effective proofs for further researching on different software system structure complexity. It also signifies that structure entropy metric method is feasible and effective.

Most current structure entropy analysis software systems focus on the problem of structure entropy of classes dependency graphs. This chapter analyzes the influence that structure entropy of class dependency graphs acts on software system complexity, which is directly composed of class methods and properties of three pieces of open source software.

2. Preliminaries

2.1. Structure entropy

The development of the concept of entropy of random variables and processes by Claude Shannon provided the beginning of information theory.

The entropy is the most influential concept to arise from statistical mechanics. Entropy measures the disorder in a system [6]. The theory of entropy is widely used in various fields. And the structure entropy can be used to describe the structure relation of a system.

If a system X has n sub-systems $X_1, X_2, \cdots X_n$, the sub-system X_i is related to X_j,then $g_{ij} = 1$,otherwise $g_{ij} = 0$. Let $N(i) = \sum_{j=1}^{n} g_{ij}$; linking intensity $\rho(i) = N(i)/\sum_{j=1}^{n} N(j)$ [7].

The structure entropy of system X:$H = -\sum_{i=1}^{n} \rho(i)\, ln\, \rho\,(i)$,where $\sum_{i=1}^{n} \rho(i) = 1$.

2.2. Class cohesion metric properties

According to Briand [8], class cohesion metrics carry these features: (1) non-negativity and normalization [0, Max], allowing easy comparison between various classes; (2) null and maximum values (no cohesive interactions, 0; all possible interactions within the class present, maximum); (3) monotonicity (even if the module is added cohesive interactions, the cohesion of the module cannot be decreased); (4) cohesive modules (merging two unrelated modules into one cannot enhance the cohesion of the module. Thus, for 2 classes, e.g. $c1$ and $c2$, the cohesion of the merged class c' must adhere to the following conditions: $cohesion(c') \leq max\{\, cohesion(c1), cohesion(c2)\}$.

3. The ordered metric methods of Classes dependency graph based on structure entropy

3.1. Dependence relations in class and dependence complex network

In the object-oriented software system, a software system is composed of a series of related classes. One class (c) describes a class in the software system, $c = (A, M)$, where $A = \{A_1, A_2, \cdots, A_a\}$is a set of attributes, and describes a attributes contained in the class c. $M = \{M_1, M_2, \cdots, M_m\}$is a set of methods, and describes m

methods comprised by the class c. Thus, dependence relations for a given class comprise: dependence relations between attributes and methods, among attributes, and among methods. Based on such dependence relations, a class dependence network $< N, E >$ can be constructed, where N and E are sets of nodes and edges, respectively. $N = N_A \cup N_M$, where N_A and N_M are sets of nodes representing the set of attributes A and methods M, respectively[9].

A dependence network between attributes

$< N_A, E_A >$, where N_A is a set of attributes, E_A is the dependence relations between the attributes. If $A_i, A_j \in N_A$, A_i and A_j are referenced by the same and common method directly, then $< A_i, A_j > \in E_A$.

A dependence network between methods

$< N_M, E_M >$, where N_M is a set of methods, E_M is the dependence relations among the methods. If $M_i, M_j \in N_M$, and there is a dependence relation between M_i and M_j (M_i references M_j directly, or $M_j$references M_i directly), or M_i and M_j reference a common attribute directly, then $< M_i, M_j > \in E_M$.

A dependence network between methods and attributes

$< N, E_{MA} >$, where $N = N_A \cup N_M$, E_{MA} is the dependence relations between the methods and attributes directly. If $A_i \in N_A, M_j \in N_M$, and there is a dependence relation between A_i and M_j (M_j references a common attribute A_i directly), then $< A_i, M_j > \in E_{MA}$.

A class dependence network is composed of three dependence networks, which are respectively between attributes, between methods, and between attributes and methods.

3.2. The Definition of Classes dependency graph Structure Entropy

Classes as the essential element in object orient software systems play important role in complexity. Structure entropy is not satisfied with cohesion mathematical characteristics in class cohesion metric. However, structure entropy in deed influences the complexity. Here, structure entropy of classes dependency graph is defined as:

$$H = -\sum_{i=1}^{n}\left(\frac{d(i)}{\sum_{k=1}^{n} d(k)} \times ln\frac{d(i)}{\sum_{k=1}^{n} d(k)}\right) \tag{1}$$

Here

$$\sum_{i=1}^{n}\frac{d(i)}{\sum_{k=1}^{n} d(k)} = 1$$

$d(i)$ is the degree of nodes in classes dependency graph, n is the number of these nodes. Suppose $\frac{d(1)}{\sum_{k=1}^{n} d(k)} = \frac{d(2)}{\sum_{k=1}^{n} d(k)} = \cdots = \frac{d(n)}{\sum_{k=1}^{n} d(k)} = \frac{1}{n}$, that is, each node in the network is equal, than the value of structure entropy is the largest up to $ln\, n$. Structure entropy of every network features as:

$$0 \le H / ln\, n \le 1 \tag{2}$$

Structure entropy quoted in class cohesion degree analysis in this paper does not play the role as an index of cohesion, since the structure entropy index does not satisfied with the monotonicity of cohesion mathematically. Assuming each node degree of the network is the same, then, the value of structure entropy is the largest one $ln\, n$. Supposing adding an edge in random in the classes dependency graph, then, the network is not uniformed, and the value of structure entropy is less than $ln\, n$.

Assuming there are 4 nodes in the class dependency graph, the degree of each node is 2, as following figure 1 showing:

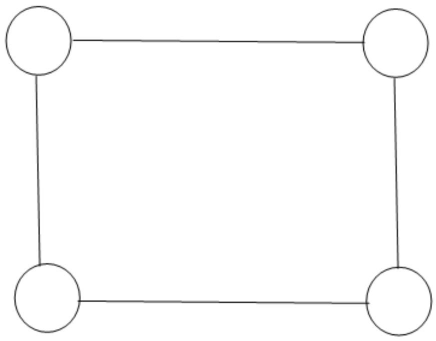


Fig. 1: Classes Dependency Graph with 4 Node and 4 Edge

Then, value of structure entropy is $ln\, 4 = 1.38629$
When adding one edge, Fig. 2 shows:

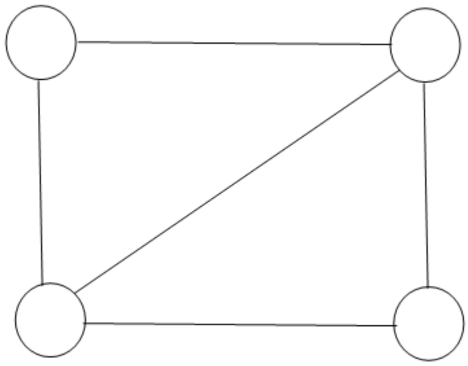


Fig. 2: Classes Dependency graph with 4 Node and 5 Edge

Then, the value of structure entropy is:

$$H = -\sum_{i=1}^{n}\left(\frac{d(i)}{\sum_{k=1}^{n} d(k)} \times ln\frac{d(i)}{\sum_{k=1}^{n} d(k)}\right) = -\left(2 \times \frac{3}{10} \times ln\frac{3}{10} + 2 \times \frac{2}{10} \times ln\frac{2}{10}\right) = 1.3662 < ln\, 4 \quad (3)$$

Therefore, structure entropy index does not satisfy mathematical cohesion metric, but due to that the entropy is the metric of uncertainty of matters, so it influences the system complexity in certain degree. Structure entropy can measure system internal randomness. In general, the more micro statuses behave in a macro status, the larger the system uncertainty is. While, more various randomness the micro statuses run inside the system, the larger the value of entropy will be; on contrary, the less the micro statuses exist inside the system, the more system unicity will be, that is, more ordered and less value of entropy.

3.3. The Algorithm of Classed Dependency Graph Structure Entropy

Calculate the structure entropy of classes dependency graph

h←0 // initialize the sum of structure entropy of each node to Zero

H←0 // initialize the sum of structure entropy of classes dependency graph to Zero

For *i* From 1 To Methods.length+ Attributes.length

If $D(i) \neq 0$

$$h \leftarrow h + \frac{D(i)}{\sum D(i)} \times \frac{Log D(i)}{Log \sum D(i)}$$

End For

$$H \leftarrow \frac{h}{Log(\text{ Methods.length+ Attributes.length})}$$

3.4. Application in Three Java Open Source Software Systems

DOI https://doi.org/2024-JICS-22403 | Generated on 2025-04-15 10:20:19

Three Java open source software systems (JOSSS) were selected in various domains: Art of Illusion [10], JabRef [11] and GanttProject [12]. Art of Illusion is a 3D rendering, modeling, and animation studio system. JabRef is a graphical application for managing bibliographical databases. GanttProject is a project scheduling application featuring resource calendaring, management, and importing or exporting (MS Project, PDF, HTML, spreadsheets). The restrictions placed on the choice of these systems were that they (1) are implemented using Java, (2) are relatively large in terms of number of class, (3) are from different domains, and (4) have available source code and fault repositories. The three JOSSS were employed by Gu Aihua, Al Dallal and Briand[9,13-20] for validating the cohesion metric.

Three pieces of open source software Illusion, Jabref and GantProject are calculated with the above algorithm. Table 1 shows the statistical values of system cohesion degrees of the three in diverse phrases (25% quartile, average, middle, 75% quartile, maximum and standard deviation). And Fig. 3 indicates that the discretions of Illusion, Jabref and GantProject are not large in the phrases of quartiles. Among the values, the difference between quartile, average and Stddev is not much sharp, the average value is about 0.9. Node degrees of most classes dependency graph are relatively average, that is, most classes dependency graphs of the three open source applications reveals relative randomness. Only from the perspective of classes orderliness, the complexities of most classes in the three are large.

Table 1: Statistical Characteristics of Structure Entropy of Classes Dependency Graph in the Open Source Programs

System	Min	Max	25%	Med	75%	Mean	Std.dev
Illusion	0	1	0.9	0.94	0.98	0.92	0.07
JabRef	0	1	0.86	0.91	0.96	0.89	0.1
GanntProject	0	1	0.86	0.91	0.95	0.89	0.09

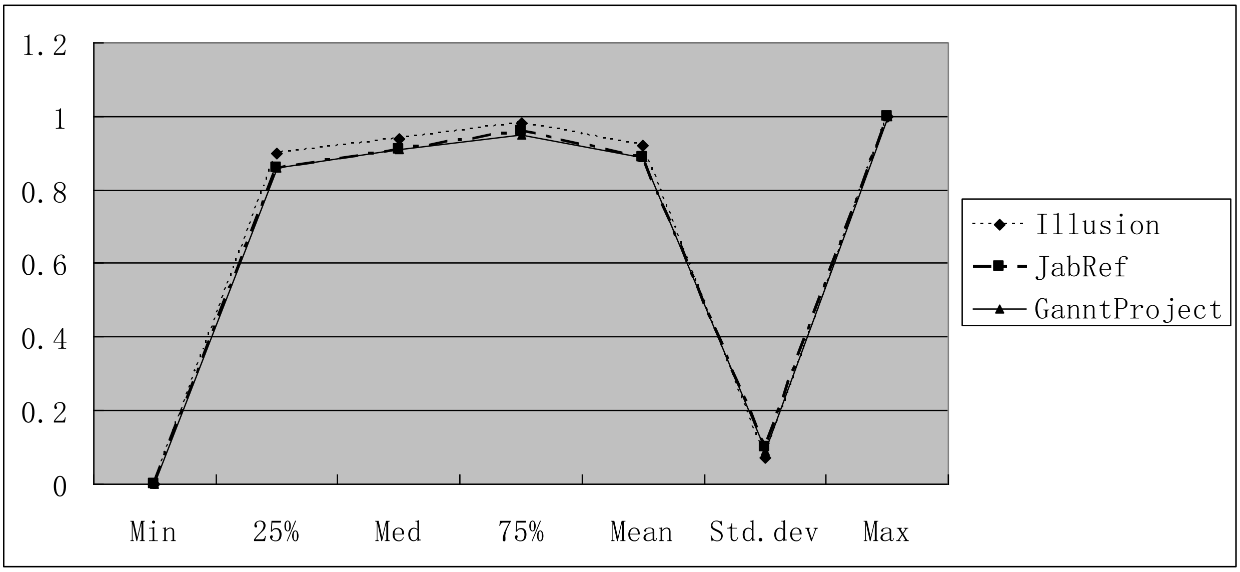
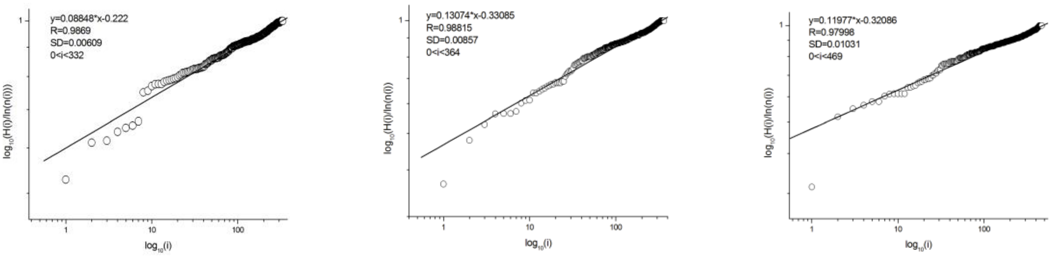


Fig. 3: Quartile, Average and Stddev Point Graph of Structure Entropy in the Three Open Source Programs

The calculation of the structure entropy values of the three programs with the algorithm shows that the structure entropy values of classes dependency graphs obey power law distribution. The conclusion is shown in Fig. 4.



（a） Illusion （b）Jabref （c）Ganntproject

Fig. 4: Classes Dependency Graphs Structure Entropy Power Law Distribution

Table 2: Pearson correlation coefficient, Standard Deviation and Power Law and its Error of the Three Open Source Programs Structure Entropies

	B	error of *B*	*R*	*SD*
Illusion	0.08848	7.97416E-4	0.9869	0.00609
Jabref	0.13074	0.00107	0.98815	0.00857
Ganntproject	0.11977	0.00113	0.98815	0.01031

Figure 4 shows the average cluster coefficient distribution of classes dependency graphs in the three open source programs. As table 2 denotes, when 0.95 is regarded as the minimum reliability value, R values of Illusion, Jabref and GanntProject are 0.9869, 0.98815 and 0.98815 respectively; while, SD values are 0.00609, 0.00857 and 0.01031 with the power exponents of B 0.08848, 0.13074, 0.11977 and power exponent errors of 7.97416E-4、0.00107 and 0.00113 respectively. The Pearson correlation coefficient (R) describes the qualitynof a given linear fit. As shown in Table 5, when 0.95 is considered the minimum reliable value [21], all distributions in Fig. 4 can be described as power-law. Result is that structure entropies of classes dependency graphs in the three open source systems follow the power law distribution indicating scale-free characteristic of complex networks.

4. Conclusions and perspectives

In this paper, structure entropy is defined in terms of node degree of classes dependency graphs of software systems. The quantization and the analysis on the order of the software system structure supply the reliable assurance for integral structure of optimization of software system classes. This paper proves mathematically that structure entropy index does not satisfy with the mathematic cohesion of classes. Therefore, structure entropy can act as a single index to measure the complexity. Three pieces of open source software Illusion, Jabref and GantProject are processed with the program written by the author basing on the definition of structure entropy. And the calculation shows the values of quartile, average and Stddev of the three pieces of the open source software. The result indicates that the discretions of Illusion, Jabref and GantProject are not large in the phrases of quartiles. Among the values, the difference between quartile, average and Stddev is not much sharp, the average value is about 0.9. Node degrees of most classes dependency graph are relatively average, that is, most classes dependency graphs of the three reveals relative randomness. Only from the perspective of classes orderliness, the complexities of most classes in the three are large. The structure entropy values of classes dependency graphs show power law distribution and feature Scale-free in complex networks, which offer reference for the further study on structure entropies of classes dependency graphs in the perspective of complex networks.

5. Acknowledgements

The current study was funded by the National Natural Science Foundation of China under grant No. 61602400. This work was also supported by the National Spark Program of Ministry of Science and Technology of China under grant No. 2015GA690259； Guiding Project of Agricultural Science and

Technology of Yancheng city under grant No. YKN2014021. The project was jointly supported by Natural Science Project of the Higher Education Institutions of Jiangsu Province of China under Grant No. 18KJD520005.

References

[1] Ma Y T, He K Q, and Du D H. A qualitative method for measuring the structural complexity of software systems based on complex networks[C]. In Proceedings of 12th Asia-Pacific Software Engineering Conference, Taiwan, 2005,257-263.

[2] Faloutsos M, Faloutsos , Faloutsos C. 1999. On power-law relationships of the Internet topology [J]. ACM SIGCOMM Compueter Communication Review, 29(4):251-262.

[3] Zhou Y, Xu B. 2003. Measuring structure complexity of UML class diagrams [J]. Journal of Electronics (China), 20(3):227-231.

[4] Xu B W, Kang D Z, Lu J J. 2004. A structural complexity measure for UML class diagrams [C]. In Proceedings of 4th International Conference on Computational Science, Springer-Verlag LNCS 3036:421-424.

[5] Yi T, Wu F. 2004. Empirical analysis of entropy distance metric for UML class diagrams [J]. ACM SIGSOFT Software Engineering Notes, 29(5):1-6.

[6] Ahlers G (1980). “Critical phenomena at low temperatures”. Reviews of Modern Physics, No.52, pp.489–503.

[7] LAI Yuan, JI Fu-xin, BI Chang-jian (2001). Evaluation of the Command Control System’s Organizational Structure (CCSOS) Based on Structure Entropy Model. Systems Engineering, Vol. 19, No. 4, pp.27-31 (in Chinese).

[8] Briand, LC.; Morasca, S.; Basili, VR.: Property-based software engineering measurement, IEEE Transactions of Software Engineering, 22, 68–86 (1996)

[9] Gu A , Zhou X , Li Z , et al. Measuring Object-Oriented Class Cohesion Based on Complex Networks[J]. Arabian Journal for Science & Engineering, 2017, 42(8):3551-3561.

[10] Illusion(Jun.2012), http://sourceforge.net/projects/aoi/.

[11] JabRef(Jun.2012), http://sourceforge.net/projects/jabref/.

[12] GanttProject(Jun.2012), http://sourceforge.net/projects/ganttproject/.

[13] Al Dallal J, Fault prediction and the discriminative powers of connectivity-based object-oriented class cohesion metrics(2012). Information and Software Technology, 54, 396-416.

[14] Al Dallal J, Briand LC(2012). A precise method-method interaction-based cohesion metric for object-oriented classes. ACM Transactions on Software Engineering and Methodology (TOSEM).

[15] Al Dallal J, Briand LC(2010). An object-oriented high-level design-based class cohesion metric, Information and Software Technology, 52, 1346–1361.

[16] Al Dallal J(2011). Measuring the discriminative power of object-oriented class cohesion metrics. IEEE Transactions on Software Engineering, 37, 778-804.

[17] Al Dallal J(2011). Transitive-based object-oriented lack-of-cohesion metric. Procedia Computer Science, 3, 1581–1587.

[18] Al Dallal J. Theoretical analysis for the impact of including special methods in lack-of-cohesion computation(2012). 2012 First World Conference on Innovation and Computer Sciences(pp.167 – 171).

[19] Al Dallal J(2013). Incorporating transitive relations in low-level design-based class cohesion measurement. Software: Practice and Experience, 43, 685–704.

[20] Al Dallal J(2012). The impact of accounting for special methods in the measurement of object-oriented class cohesion on refactoring and fault prediction activities. The Journal of Systems and Software, 85, 1042–1057.

[21] Savić, M.; Ivanović, M.; Radovanović, M.:. Characteristics of class collaboration networks In Large Java Software Projects. Information Technology and Control Journal, 40, 48-58 (2011).