Complexity Measurement of Large-Scale Software System Based on Complex Network

Li Dali
College of Computer Science & Technology, Jilin University, Changchun, China, 130012
Email: lidali001@163.com

Abstract—With the increase of software system complexity, the traditional measurements can not meet the requirements, for the reason that the developers need control the software quality effectively and guarantee the normal operation of software system. Hence how to measure the complexity of large-scale software system has been a challenge problem. In order to solve this problem, the developers have to obtain a good method to measure the complexity of software system first. Only through this work, the software quality and the software structure could be controlled and optimized. Note that the complex network theory has offered a new theoretical understanding and a new perspective to solve this kind of complexity problem, this work discusses the complexity phenomenon in large-scale software system. Based on this, some complexity measurements of large-scale software system are put forward from static structure and dynamic structure perspectives. Furthermore, we find some potential complexity characteristics in large-scale software networks through the numerical simulations. The proposed measurement methods have a guiding significance on the development for today's large-scale software system. In addition, this paper presents a new technique for the structural complexity measurements of large-scale software system.

Index Terms—Software Complexity; Complex Network Theory; Complexity Measurement; Large-Scale Software System; Whole Perspective

I. INTRODUCTION

The measurement study is an indispensable element in the area of engineering technology. The measurement technology is gradually integrated into the software engineering field, and has become an important part of a good software engineering. It is because that whatever can not be measured, it can not be controlled. The measurement is helpful to understand and control the various activities in the process of software development, as well as improve the process and products. With the development of software engineering, the quantitative point of view for the process of software development, products and information has been paid attention to by more and more scholars, so the research on software metrics has pushed on in recent years [1] [2] [3] [4] [5]. The traditional object-oriented measurements, such as C&K method [6], Mood method [7] and so on, focus mainly on the class. They measure the internal complexity of the class, the complexity of the relationship between class and class, such as inheritance complexity, the coupling complexity and so on. These measurements are focused solely on statistics number or size relating to the object class. In terms of large-scale software, too much emphasis is on fine-grained statistics, and no attention has been paid on the measurement of the coarse-grained modular construction. In addition, the large-scale software measurement is an important guarantee for the science of software design and development. The software development is often in a runaway state for the reason that a reasonable measurement approach is lack. The measurement of software complexity has close relation with the reliability, the maintainability and other characteristics of software.

On the other hand, the formulation of complex network theory in recent years gives a new perspective for the measurement of large-scale software systems. It has been shown that many networks, such as social networks, ecological networks, Internet, and software systems have obvious complex characteristics, including small-world feature, scale-free feature and so on. Through the approach that social systems, ecological systems in actual life are modeled as complex networks, some scholars have studied their structural characteristics and their safety problem and found many significant results of these networks [8] [9] [10] [11] [12]. These studies have an important influence on the social, biology and physics development.

Studies have shown that large-scale software systems also include complex characteristics [13] [14] [15] [16] [17]. There are a lot of units with different sizes in large scale software systems. Once these software units and the interactions between them are modeled, we can find that many complex systems have obvious complex network phenomenon. If we apply the nature and the law of complex network to software system and investigate the structural features of the software system, we can better understand and control software system. The basic work of software measurement began at a very early age [18]. And then the calculation of the number of linearly independent paths is proposed to measure the complexity of the program. In this study, the proposed McCabe ring complex degree is a more successful measurement method on structured program. It is worth noting that it is the first time that the internal network topology features of software systems are considered. For the nature evaluation of software measurement, there are seven
standards for the quality assurance and the object-oriented software life cycle. In the later study, the object-oriented ideas become the mainstream of the software design and the software development. Some measurement methods which are suitable to the object-oriented software appear. For instance, the C&K measurement based on inheritance tree is very representative approach [6] [19] [20]. These methods are mainly focused on the perspectives which are the internal complexity of the class, the complexity of inheritance between classes and the complexity of the coupling between classes. However, these methods have the following problems. First, although these methods pay attention to the coupling between the object classes which only is a kind of number statistics, the interaction between the classes has not reflected. Second, the study is still at the stage of measurement on internal attributes of software systems, the definition on the relationship between the external and internal properties is lack. Third, these studies focus on the microscopic statistics. Although the complexity of the software systems is usually the overall performance, little work is on the complexity measurement based on the global and overall perspective. In addition, software quality and software development are troubled with the software complexity, especially for large-scale software systems [21] [22].

Considering these existing problems, we analyze the complexity of the large-scale software systems and finds that they have the obvious complex network feature. In this work, the complex network theory is introduced to analyze the complexity characteristics of large-scale software systems first. Combining with the internal topology features of object-oriented software, we further propose a measurement model for the complexity of large-scale software systems from a whole level which is based on the current research findings of complex network theory. And then some measurement approaches are presented so as to explore the complexity of large-scale software systems from static structure perspective and dynamic structure perspective, respectively. At the same time, we attempt to integrate the complexity measurement of the traditional software. Finally, a large-scale software system as a case study is proposed to apply our measurements. As a result, we find that the degree of this software network follows power-law distribution, as well as some other significant results. It can be seen that this work gives a perspective for meaningful future research on the complexity of the large-scale software systems.

II. COMPLEXITY MEASUREMENT OF LARGE-SCALE SOFTWARE SYSTEM

A. Complex Characteristics

The complex networks use graph theory method and statistical physics method to capture and describe the evolution mechanism and the overall behavior of various system [5-6]. For the description of complex networks, the nodes represent different individuals in the system, and the links represent the specific relations between individuals. Let $G = (V, E)$ be a network graph on $N$ vertices where $V = \{v_1, v_2, \ldots, v_N\}$ and $E = \{e_1, e_2, \ldots, e_m\}$. $N = |V|$ and it is the total number of all nodes. Each edge $e_i$ in the set $E$ is corresponding to a pair of nodes in the set $V$. The networks could be divided into the directed and weighted network, the undirected and weighted network, directed and unweighted network, and undirected and unweighted network according the real situation.

For the directed complex network, the degree of a node refers to the number of edges which are connected with the node. The in-degree of a node is the number of edges which point to this node from the other nodes to this day, while the out-degree of a node represents the number of edges which point to other nodes from this node. The node degree distribution is represented by the function $p(k)$ which refers the probability that a node has $k$ edges. The results have shown that the node degree distribution mainly has two forms: one is the exponential distribution, and the other is a power-law distribution.

The clustering coefficient is used to measure the clustering degree of complex networks, that is, the connectivity degree between nodes within complex networks or the degree of local transmission between network nodes. The average path length refers to the average shortest distance among all the nodes in complex networks. The shortest distance is the minimum number of the edges from one node to another node. Many studies have shown that the real complex networks have the large clustering coefficients and the small average paths.

A large-scale software system divides a complex problem into multiple parts, and then is finished by multiple developers working together. During the development process, the complex functionality is decomposed into a large number of reusable components according to the functional needs. If these components are regarded as nodes, the relationships between the components represent the edges between the nodes. In this way, a networked model of the large-scale complex software system could be created.

The key of the software complexity is the complexity of the structure because the changes of the structure complexity can cause the change of the external quality in the software system. The traditional software measurement method is difficult to measure the change of the external quality. We need to measure the software structure from the granularity of the system. The complex system theory emphasizes that we should control system from the whole perspective, rather than focus on the local. The recent study on the complexity of the software structure is mainly on the static research of the software structure, rather than on the dynamic study of the software structure.

B. A Networked Method for Large-Scale Software System

At present, the research of large-scale software systems using complex network theory and the networked method is mainly aimed at the open source system using the reverse engineering method which begins from the
process of the sequence code, that is, from the code to the class diagram of the system, and then to the network graph. Thus, we analyze the structure of the large-scale software systems and find their complex network features. Finally, we could get the overall properties which would not be obtained by a single component accumulation in the large-scale software systems. As a simplified example, we give the following software system.

Here, we abstract the relations between classes in the simplified software system and obtain Figure 1.

![Figure 1. The representation of over-simplified relations in a software system](image)

Furthermore, we get the network representation of a simplified software system through applying the networked method, as shown in Figure 2.

![Figure 2. The network representation of a simplified software system](image)

C. The Proposed Complexity Measurements

The empirical analysis has shown that the topology structure of complex software system is not random, but has small-world property and scale-free property. At present, the development of large-scale software system is mainly based on the component software technology and the service-oriented architecture, and each component is designed to operate by the object-oriented programming methods. Thus, it would be closer to the objective world, and the scale-free property is more obvious. The reusability property of the component in software systems makes small-world feature more prominent. There is a hierarchical feature between the components because of the autonomy and the heterogeneity of the components. Those components interact with each other in the form of services and then the coupling of them seems to be looser. Therefore, the operating platform of the large-scale software systems is not only networked, but also the topology features of the large-scale software systems have the obvious complex network topology characteristics.

For the static measurement on large-scale software systems, the classes, the objects and the methods are regarded as the nodes. Their mutual relations are abstracted as the edges in the networked research. The measured parameters based on the complex network theory are given below.

1) The global topological structure of large-scale software system

It is assumed that $d_{\min}(m,n)$ is the length of the shortest path between node $m$ and node $n$. For the network graph $G$ of a large-scale software system, its average distance is defined as follows:

$$d = \frac{1}{N(N-1)} \sum_{m \neq n} d_{\min}(m,n)$$

where $N$ represents the number of the nodes and $V$ is the set of the nodes. The value of $d$ is used to measure the global topological structure of large-scale software system.

2) The network clustering coefficient of large-scale software system

The network clustering coefficient is used to measure the cluster level of the nodes. The clustering level of a node $i$ is defined as the proportion of the number of the existing edges on the neighboring nodes of node $i$ and the number of the completely connection edges between neighboring nodes. That is, the clustering coefficient of the node $i$ is expressed as the following.

$$C_i = \frac{2E_i}{n_i(n_i - 1)}$$

where $n_i$ is the number of edges which connect with node $i$, and $E_i$ is the number of the existing edges between the $n_i$ nodes.

Then, the clustering coefficient of the architecture structure of the large-scale software network is the average value of the clustering coefficients of all nodes in the network, i.e.,

$$C_G = \frac{1}{N} \sum_{i=1}^{N} C_i$$

where $C_i = \frac{2E_i}{n_i(n_i - 1)}$. Obviously, $C_G \in [0,1]$. When $C_G = 0$, it implies that all the nodes are isolated nodes. When $C_G = 1$, it represents that the entire network is connected, and any two nodes are connected directly.

The empirical analysis shows that most of the nodes in the large scale software systems tend to gather together. Despite the clustering coefficient $C_G$ is far less than 1, but are far more than $N_l$ in the completely random network. In the actual software system, we also found the same phenomena. The smaller average distance and the
larger clustering coefficient become the features of a lot of large-scale software systems.

3) The community modularity of large-scale software system

The definition of the community modularity calculation formula is based on the two facts. First, the edges between the nodes within the community should be as much as possible. Secondly, the edges which connect the nodes between the communities should be as little as possible. The modularity is defined as the following.

\[ T = \sum_{l=1}^{k_i} \left[ \frac{l_i}{L} - \frac{k_i}{2L} \right] \]  

(4)

where \( L \) is the number of all edges in a network, and \( l_i \) is the number of the edges in the community \( Q \). The total degree of all nodes in the community \( Q \) is denoted by \( k_i \).

According to the definition of the modularity in Eq. (4), the modularity is the difference between the proportion of the edges within the community and the expectation ratio of these corresponding edges. The greater the number of edges within the communities and the fewer the number of edges between the communities, the value of \( T \) is larger. Thus, the features of the communities are more significant. For the large-scale software systems, the scope of the interaction between nodes is often the intrinsic semantic collaborative results of the nodes. For instance, the packages, components, and so on, are to synthesize a set number of classes into the higher-level units, and form a high cohesion and low coupling set of the classes. Therefore, for the given packages or components in the large-scale software system, the larger value of \( T \) indicates that the framing module is better.

4) The betweenness in large-scale software system

The betweenness centrality is divided into the betweenness of the node and the betweenness of the edge. The betweenness of the node refers to the number of the shortest paths which pass through the node in all of the shortest paths on the network. It reflects the influence capacity of the node. We denote the set of the shortest paths between node \( i \) and node \( j \) by \( S(i,j) \). The betweenness of the node \( m \) is defined as the following.

\[ B_m = \sum_{i,j \in S(i,j)} \frac{\xi_{ij}^m}{|S(i,j)|} \]  

(5)

In the above expression, if \( l \) passes through the node \( m \), then \( \xi_{ij}^m = 1 \). Otherwise, if \( l \) does not pass through the node \( m \), then \( \xi_{ij}^m = 0 \). The definition on the betweenness of the edge is similar to that on the betweenness of the node. According to the betweenness of the node and the betweenness of the edge, we can analyze the influence on the entire network when some relation between classes or objects is deleted. Thus, we could provide the guidance for the reconstruction and the optimization of the large scale software system. However, a traditional software measurement method could not reach this purpose.

5) The correlation in large-scale software system

The correlation between the degree and the clustering coefficient is used to describe difference among various network structures, which includes two aspects: the correlation between the nodes with the different degrees, and the correlation between the connectivity of the degree of the node and its clustering coefficient.

Table 1 presents the basic parameters and the corresponding roles of these measurements in a large-scale software system.

<table>
<thead>
<tr>
<th>The basic parameter</th>
<th>Symbol</th>
<th>The role of the measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>The average distance</td>
<td>( d )</td>
<td>Used to measure the efficiency of the information transmission, the cost of the communication and the response ability of the system between the software entities in the large-scale software system.</td>
</tr>
<tr>
<td>The degree distribution</td>
<td>( p_i )</td>
<td>Used to evaluate the complexity degree of the large-scale software system, and analyze the relationship between the network structure and the system function.</td>
</tr>
<tr>
<td>The average clustering coefficient</td>
<td>( C )</td>
<td>Used to measure the degree of cohesion of the different granularity software entities in the large-scale software system.</td>
</tr>
<tr>
<td>The betweenness</td>
<td>( B )</td>
<td>Used to analyze the impact caused by the failure of any software entity or the failure of a relationship between the software entities on the whole large-scale software system.</td>
</tr>
<tr>
<td>The correlation</td>
<td>( R(k, k_i) )</td>
<td>Used to analyze the collaboration relation between the different software entities, and then find the entity which has some problem.</td>
</tr>
<tr>
<td></td>
<td>( R(k, C) )</td>
<td>Used to analyze the hierarchy and the modularization degree in the large-scale software system.</td>
</tr>
</tbody>
</table>

III. NUMERICAL SIMULATIONS

A. A Case Study

With the development of the computer technology and the network technology, a government construction sector exploits the spatial information system. This corresponding software system serves the various functional departments respectively and provides the support and the regulation guidance for the economic construction of the region. The system includes five subsystems which are the agriculture, the water, the land, the environment, the public safety, respectively. In order realize the networked approach in section 2, considering the corresponding software system is large-scale, this article selects the detection and evaluation system for the agricultural information system in the water conservancy subsystem. This large-scale software system is mainly to achieve the survey in the various functional departments of the region, and further establish a remote sensing monitoring model of rapid identification. The
model could realize time monitoring on the operation and the effectiveness in the different types of functional departments. For example, whether there is disaster or not, and whether funds or substances would be shortage, and so on. In this way, the large-scale software system provides the rapid intuitive and reliable information for the dynamic monitoring and the evaluation of the entire system.

The geographical information system database
The remote sensing image
The use types of the functional departments
The fast extraction model on the running of the functional departments
The estimates of invalid operation range
The invalid operation scope and the degree of the functional departments use type
The economic factors
The data extraction of classes
The complexity calculation of the large-scale software system
The evaluation on the loss based the complexity calculation of the large-scale software system

Figure 3. The process representation of a large-scale software system structure

We transform the process representation in Figure 1 into the formal architecture graphics expression, as shown in Figure 4.

Figure 4. The formal architecture graphics representation of this software system

We put a class of the large-scale software systems as a unit and decompose them into the class unit collection. Through the interaction between the units of these classes, the various functions of the large-scale software system can be realized. If the units in the software system are regarded as the nodes, and the interactions between them as the edges, then the software structure presents a network topology based on a complex interconnection. Thus, the networked graph of this software structure is realized, which is shown in Figure 5.

B. Simulation Results and Complexity Analysis

The basic statistics indicators of the complexity of an undirected and unweighted network include the degrees, the degree distribution, the average path length, the clustering coefficient and other indicators. According to the proposed measurements of the complexity in this paper, we use Matlab 2010 software to program and obtain the statistic calculated results, which are shown in Table 2. Table 2 show that the statistically characteristic parameters of the complexity in the large-scale software system. On this basis, the complex features of the large-scale software system would be further explored.

TABLE II. THE STATISTICALLY CHARACTERISTIC PARAMETERS OF THE COMPLEXITY IN THE LARGE-SCALE SOFTWARE SYSTEM

<table>
<thead>
<tr>
<th>The large-scale software system</th>
<th>$N$</th>
<th>$L$</th>
<th>$C$</th>
<th>$B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d$</td>
<td>36</td>
<td>71</td>
<td>0.1312</td>
<td>3.9267</td>
</tr>
<tr>
<td>$R$</td>
<td>3.6615</td>
<td>9.2561</td>
<td>4.9311</td>
<td>1.3568</td>
</tr>
</tbody>
</table>

Table 2 shows that the average shortest distance of the network is 3.6615. The average shortest distance of a network reflects the global characteristic of the network. The value shows that to the average separation degree of the network node pairs, and is negatively correlated with the convenient level of the information dissemination within the network. In addition, the value of the clustering coefficient is 0.1312. This implies that the connection probability of an edge between any two nodes in the large-scale network is 13.12%. That is to say, there is a relatively high clustering coefficient. The greater network clustering coefficient shows that the exchange of the energy and information among all enterprises in the
network is more convenient, and the collectivization level of the network is higher.

Considering the directions of the interactions among the units, most nodes are with a small in-degree or a small out-degree. In addition, the nodes in the source position or in the end position have only the out-degree or the in-degree. We regard that the indicator based on the network density can be used to judge the status of nodes in a particular location within the large-scale software system. For example, if the in-degree of a node is 0 the node can be regarded as the source node in the information relationship chain. On the other hand, if the out-degree of a node is 0, the node can be regarded as an end node in the information relationship chain. In this paper, in order to verify the complexity feature of the large-scale software network, we examine the degree distribution of this network under the double logarithmic coordinates. Thus, when there is a linear relationship between \( \log k \) and \( \log p(k) \), they would be satisfied with the relation as the following.

\[
\log p(k) = -\lambda \log k + \mu \tag{6}
\]

i.e.,

\[
p(k) \sim k^{-\lambda} \tag{7}
\]

This implies that the software network meets the power-law distribution. This paper adopts the programming software to compute the degree distribution of this network, further calculates the corresponding value, and finally determines whether the network has a scale-free complex feature. The simulation results are shown in Figures 6, 7 and 8.

Figures 6-8 show the degree distribution of this large-scale software network under the double logarithmic coordinates. Figure 6 and Figure 7 represent the in-degree distribution and the out-degree distribution of the software system in Figure 5. The results show that in the out-degree distribution and in-degree distribution of this network, there exist linear relationship between the \( \log k \) and \( \log p(k) \). They are both satisfied with the equation \( \log p(k) = -\lambda \log k + \mu \). Therefore, for the large-scale software system, the in-degree distribution and the out-degree distribution of this networked structure meet the power-law distribution which is the characteristic of scale-free network.

Figure 8 shows the cumulative degree distribution of this large-scale software network under the double logarithmic coordinates. In Figure 8, we find that the cumulative degree distribution of this network approximates a straight line. We adopt the least squares method to obtain the fitting curve expression as the following.

\[
p(k) = 0.3316k^{-0.9107} \tag{8}
\]

Therefore, according to Eq. (8), it can be seen that this large-scale software network is a scale-free network through the analysis on the complexity of the large-scale software network. This indicates that there are a small number of nodes existing in the software system. These nodes are called the network hub nodes which refer to the nodes with high degree. The nodes have the dominant position in the network and could affect the other nodes with small degree relatively. Meanwhile, the scale-free network has a feature of the preferential attachment. Hence the new nodes within the large-scale software system tend to connect with the hub nodes. This phenomenon is also known as the phenomenon of “rich get richer”.

IV. CONCLUSION

The complexity of the large-scale software systems is closely related to the development cost, the software quality and the production efficiency. Hence the measurement on the complexity of the large-scale
software systems is a challenging computer science problem. Especially, the complexity of the modern computer software systems is becoming more and more complex. The complex software is more difficult to guarantee the quality of its products, the production efficiency and the development costs. The software development is easy to be in a state of out of control, and even lead to the avalanche effect. It is because that the large-scale software systems are constituted by countless details. Thus, a small mistake can lead to the disastrous consequences. Therefore, how to recognize, measure, manage and control the complexity of the software has become an extremely important and challenging issue in the field of software engineering. In addition, the measurement and evaluation for the structural complexity of the software require the rational description and efficient quantization for the software structure and information.

Although the complex network research has made considerable progress in recent years, and its theories, techniques, methods and results provides a new tool for the global structure and the complex behavior of software systems. However, the traditional measurement methods focus on a single module, and ignore the structural integrity of the large-scale software systems. That is to say, those methods focus on the microscopic statistics, and don’t consider the global and overall structure of the software measurement. Actually, the large-scale software system as a kind of complex system, the topology structure is bound to affect its functionality, performance and quality of the whole system.

In this article, we employ the complex network theory to analyze the statistical properties of large-scale software systems from a whole perspective. The nodes in large-scale software systems represent the components, and the edges between the nodes denote the relationships of the information exchanges. In this way, the complex network theory is supposed to be effective in describing the real large-scale software system. After describing the case of certain large-scale software system, we establish the topology of the large-scale software system by employing the Ucinet (version 6.0) software. The statistical characteristics of the network in the software are further examined. We study the phenomenon of complex networks in large-scale software system, and proposed metrics from several different respects. We explore the phenomenon of complex networks in the large-scale software system, and propose metrics from three different levels. An index system including the node degree, the degree distribution, the clustering coefficient, the average shortest distance length and the modularity is developed to measure the complex properties of the large-scale software systems. The results show that the average shortest distance length and the clustering coefficient in the abstracted software network are both higher than those of random network. In addition, we find that the degree of this software network follows power-law distribution. Therefore, we can draw a conclusion that the large-scale software network has the scale-free and small world characteristics. In summary, a perspective for meaningful future research on the complexity of the large-scale software systems was given in this paper, including the aspects such as network modeling and the complexity features based on a dynamical evolving mechanism of software systems.

REFERENCES


