Analysis on evolution features of software network

Zheng Liu[1], Qian Zhang[2]

[1]College of Information Science and Engineering, Northeastern University, China

[2]Technology Strategy & Development Department, Neusoft Corporation, China

liuzheng@mail.neu.edu.cn

Keywords: preferential attachment mechanism, evolution features, BA model, software network

Abstract. For software structure is described by directed network more accurately, traditional preferential attachment mechanism such as in BA model is not suitable for software network during network evolution. According to the change of software static structure during evolution of software system, the evolution features of software network and its formation mechanism are studied in this paper, which will provide guides to modeling software network during software evolution.

Introduction

The evolution of software is a long-time dynamic process and the structure of the system most of which are inherited from former must be changed to meet the new application environment [1]. Along with the evolution features being revealed, the researchers try to study the modeling of software network by means of modeling in complex network in order to find the forming reason and the future trend of software network during evolution. Myers presented an evolution model of software network based on reconstruction. He believed that the increasement of software network followed the reuse mechanism, and the software system must have continuous evolvability besides realize complex function [2]. He Keqing, etc. studied the increasement of software system starting with design patterns in software engineering and presented a model for Orient-Object software based on pattern increasement, which is verified with complex network features [3]. Zheng Xiaolong, etc. believed that the probability of vertices attached is related with not only its scale value but also its existing time in the network. Always long existing time lead to low attached probability [4]. Tessone, etc. believes that the essential factor of the power law degree distribution in software network is the inhomogeneous degree distribution of new vertices [5].

But most of the models presents above are based on undirected network, and the differences of the software network structure on each level are not considered. So in this paper, the multi-granularity features of software network, and also the preferential attachment and selection mechanism during the network establishing are studied by analyzing many real software network samples. Furthermore, an evolution model for multi-granularity software network is presented and compared with real software network by simulation test.

Increase of software network

In order to study the evolution features of real software network, we choose six open source software systems and each of them contains eight evolution versions. The statistics of the network scale increasement during these software systems evolving is listed as Table 1.

Table 1 Increasement of software network scale during software evolving

Software	Versions	Number of vertices	Software	Versions	Number of vertices
Azureus	V2100 to V4700	750 to 3847	Firefox	V1.0 to V8.0	5621 to 7226
Koffice	V1.2.1 to V2.3.2	1688 to 6233	Tomcat	V3.0 to V7.0.20	203 to 1363
Eclipse	V2.0.1 to V3.7	6172 to 24734	VTK	V2.4 to V5.10	566 to 3441

It can be seen that along with the software evolving, the scale of software network is increasing. The increase of software network scale is mainly in the increase of number of vertices and edges [6]. It is also found that the relationship between the increase of vertices and that of edges is almost

direct proportion. So the increasing of software network is considered as new vertices joining in continuously and when new vertices join in, m edges join in at the mean time. Here m is the ratio of the number of new edges to new vertices that can be calculated by statistics of real software network increasing.

Preferential attachment mechanism

Limitations of preferential attachment mechanism in BA model. At present, there are much research work on modeling of complex network and software network, and among which the preferential attachment (BA model) proposed by Barabási and Albert is the most famous and widely used [7]. This modeling is described as a network is started with a small network and increased through new vertices joining in. The probability these new vertices will attach themselves to the existing vertices is direct proportional to their degree values. But the relationship between different classes in Orient-Object software system is always unidirectional, and the power law distribution in undirected network cannot represent the structure features of software network completely.

The relativity between in-degree and out-degree marked by $R(K^{in}, K^{out})$ of real software network and the network generated by BA model is studied separately. The comparison result is listed in Table 2. It can be found that there is positive relativity between in-degree and out-degree in BA generating network while it is opposite to real software network. Especially analyzing on the vertices with high degree over 10, it is found there is obvious negative relativity between in-degree and out-degree in real software network, which is agree with reference [8]. According to the statistics of degree relativity based on BA model, although more connections will prefer to attach to the vertices with high degree in undirected complex network model, it cannot reflect the whole structure features of software network. So the preferential attachment should be discussed on directed software network, and it is necessary to discuss the evolution mechanism of software network further.

Table 2 comparison of degree relativity in real software network and simulation network based on BA model

software	R in real network	R in BA model	R of high degree vertices in real network	R of high degree vertices in BA model
Azureus V4700	-0.036	0.803	-0.389	0.86
Eclipse 3.7	-0.022	0.916	-0.54	0.957
Firefox 8.0	-0.025	0.842	-0.448	0.908
Koffice 2.0.0	-0.0102	0.847	-0.456	0.93
Tomcat 7.0.20	-0.032	0.828	-0.38	0.899
VTK 5.10	-0.06	0.774	-0.588	0.925

Directions of connection. An important difference between software network and other complex network is the in-degree and out-degree of vertices is not positively correlated. Usually, vertices with higher in-degree have lower out-degree and vice versa. The course that new vertices join in is actually that they attach to existing vertices in network while software evolving, including incoming and outgoing directions. If a vertex get a new incoming connection, it means the corresponding class is reused once, and on the contrary, if a vertex get a new outgoing connection, the corresponding class depends on the new vertex. So the ratio of number of incoming connections to that of outgoing connections may represent the reuse level of original code during software evolution.

Vertices set $\Delta V(i)$ is defined to denote the set of new vertices in software network with version i+1 that evolving from version i, and there is

$$\Delta V(i) = V(i+1) - V(i+1) \cap V(i) \tag{1}$$

Further, $V^*(i) \subset V(i)$ is defined to the set of vertices in original network that get new connection from new vertices, and $\Delta E(i)$ is defined to denote new connections between vertices in $\Delta V(i)$ and those in $\Delta V^*(i)$. $e_{m,n}$ denotes the edge between vertex v_m and v_n, where $v_m \subset \Delta V(i), v_n \subset V(i)$. Any vertex $v_j \in V^*(i)$ get l_j^{in} incoming edges and l_j^{out} outgoing edges that are in $\Delta E(i)$. Then, $L^{in} = \{l_1^{in}, l_2^{in}, l_3^{in}, \dots, l_n^{in}\}, L^{out} = \{l_1^{out}, l_2^{out}, l_3^{out}, \dots, l_n^{out}\}$ are defined to denote the number of incoming connections and that of outgoing connections of each vertex in

$V^*(i)$ separately, which meet the relation:

$$\sum_j l_j^{in} + \sum_j l_j^{out} = |\Delta E(i)| \tag{2}$$

So, we can get the ratio of new incoming connections to all the new connections $p\left(L^{in}|\Delta E(i)\right)$, and

$$p\left(L^{in}|\Delta E(i)\right) = \frac{\sum_j L^{in}(j)}{|\Delta E(i)|} \tag{3}$$

According to the above discussion, the statistics of the ratio of new incoming connections to all the new connections $p\left(L^{in}|\Delta E(i)\right)$ is listed in Table 3 by comparing each pair of software networks with two adjacent versions.

Table 3 ratio of new incoming connections to all the new connections during software evolution

software	1 to 2	2 to 3	3 to 4	4 to 5	5 to 6	6 to 7	7 to 8
Azureus	78.87%	85.69%	73.45%	80.05%	69.83%	82.70%	75.28%
Blender	78.99%	80.99%	85.62%	87.67%	83.04%	80.06%	84.83%
Eclipse	86.38%	84.15%	83.5%	88.65%	84.87%	88.7%	83.94%
Firefox	81.16%	85.83%	81.23%	77.19%	85.23%	89.42%	82.56%
Tomcat	77.06%	69.91%	73.89%	89.05%	93.05%	81.33%	94.74%
VTK	86.07%	84.99%	82.27%	93.85%	88.89%	83.31%	92.73%

It can be seen from above statistics that most of new connections in evolution model are incoming direct, which explains that software reuse is used widely in practice of software engineering. Reuse in software system can actually improve the efficiency of design and development, and furthermore even be good for software evolution.

Probability of connecting. The probability that vertices get new connection is direct proportion to their degree value in BA model. In software network, the probability that vertices get an incoming connection or outgoing connection reflects the reusing level of software system. The relativity between vertices getting new directed connection and their degree values is studied by statistics of the in-degree value and out-degree value of each vertex and the number and direction of new connection they get.

First, as to each vertex $V(i)$, their set of incoming connection L^{in} and set of outgoing connection L^{out} are defined. Then the relativity between set of degree value, including of in-degree and out-degree, and set of new connection is by calculating their relativity coefficient *R*. The results showed that the relativity between set of in-degree value and set of new incoming connection is higher than that between set of degree value and set of new incoming connection. Similarly, the relativity between set of out-degree value and set of new outgoing connection is higher than that between set of degree value and set of new outgoing connection. So it can believe that the probability of a vertex getting an incoming connection p^{in} is direct proportion to its in-degree value k^{in}, and the probability of a vertex getting an outgoing connection p^{out} is direct proportion to its out-degree value k^{out}.

Conclusion

Preferential attachment mechanism is the main formation cause for free-scale networks during evolution and BA model is a typical evolution model using this mechanism. We compare a simulated software network built based on BA model with real software network and discuss the serviceability and limitations in modeling of software network. According to the limitations of BA model using in evolution of directed networks such as software network, we discuss the preferential attachment mechanism fit for directed network and find that most of new connections in evolution model are incoming direct and the probability of getting an incoming connection is direct proportion to the vertex's in-degree value, as the same with that of outgoing connection. These evolution features will provide theory basis to modeling software network.

Acknowledgements

This work is supported by Fundamental Research Foundation of the Ministry of Education of China

(Grant No.N110304003)

References

[1] Breivold H P, Crnkovic I, Larsson M. A systematic review of software architecture evolution research[J], Information and Software Technology, 2012, 54(1): 16-40.

[2] Myers C R. Software systems as complex networks: Structure, function, and evolvability of software collaboration graphs [J], Physical Review E, 2003, 68 (2): 046116.

[3] He K Q, Peng R, Liu J, et.al. Design Methodology of Networked Software Evolution Growth Based on Software Patterns [J], Journal of Systems Science and Complexity, 2006, 19(2): 157-181.

[4] Zheng X L, Zeng D, Li H Q,Wang F Y. Analyzing open-source software systems as complex networks [J], Physica A, 2008, 387(24): 6190-6200.

[5] Tessone C J, Geipel M M, Schweitzer F. Sustainable growth in complex networks [J], Eurpphysics letters, 2011, 96(5): 58005.

[6] Israeli A, Feitelson D G. The Linux kernel as a case study in software evolution [J], The Journal of Systems and Software, 2010, 83(3): 485-501.

[7] Barabási A-L, Albert R. Emergence of scaling in random networks [J], Science, 1999, 286: 509-512.

[8] Myers C R. Software systems as complex networks: Structure, function, and evolvability of software collaboration graphs [J], Physical Review E, 2003, 68 (2): 046116.