软件网络的多粒度拓扑特性分析及其应用

韩言妮[1] 李德毅[1),2)] 陈桂生[2)]

[1)](北京航空航天大学软件开发环境国家重点实验室 北京 100191)

[2)](中国电子工程系统研究所 北京 100141)

摘 要 随着软件与网络的融合,以网络为基础的软件系统在规模、用户数量、组成单元的交互关系方面都成数量级的增长,成为一类重要的复杂系统,超出了开发人员的理解和控制.文中首先分析软件工程 40 年来软件开发设计方法学的变迁历程,并给出软件网络的定性描述,提出网络时代软件工程的新观点.然后以 eCos 开源系统为分析载体,从不同粒度上对网络模型进行拓扑特性分析,结果表明该载体网络在不同粒度上具有自相似的结构特性.最后针对嵌入式系统的本质特性,从节点异质性、局部抱团性和多粒度的网络规模简约 3 个方面进行实证分析,用以指导嵌入式软件系统的可配置、可裁剪目标,从而达到资源的最小负载.

关键词 软件网络;中心性;社区结构;拓扑势;复杂网络

中图法分类号 TP311 **DOI 号**: 10.3724/SP.J.1016.2009.01711

Analysis on the Topological Properties of Software Network at Different Levels of Granularity and Its Application

HAN Yan-Ni[1] LI De-Yi[1),2)] CHEN Gui-Sheng[2)]

[1)](*State Key Laboratory of Software Development Environment, Beihang University, Beijing* 100191)

[2)](*Institute of Electronic System Engineering, Beijing* 100141)

Abstract With the integration of software and Internet, complexity of software systems based on the Internet is an ever-present barrier in system development and evolution. Its principal manifestation is the massive accumulation of low-level details and intricate relationships among them that quickly exceeds human understanding. This paper presents the qualitative description of software network in the context of complex network, then gives analysis of the change history of software development methodology and introduces the new perspective of software engineering in network age. To validate this idea, the authors study the eCos open source system and analyze their network structure at different levels of granularity. Results show that they exhibit the structure characteristics of self-similarity at different levels. Finally, in accordance with the essence of embedded system, the heterogeneity of nodes, community structure and network reduction are tested at different granularity. These results have some practical value including that it allows us to identify important components, instructs the software configuration and cut-off with less load cost, thereby assists the maintenance and reusability of the embedded systems.

Keywords software network; centrality; community structure; topology potential; complex network

收稿日期:2009-04-19;最终修改稿收到日期:2009-07-26.本课题得到国家"九七三"重点基础研究发展规划项目基金(2007CB310800)、国家自然科学基金(60496323)资助.**韩言妮**,女,1981 年生,博士研究生,主要研究方向为复杂网络、软件工程.E-mail: hyn@nlsde.buaa.edu.cn.**李德毅**,男,1944 年生,博士生导师,中国工程院院士,主要研究领域为复杂网络、数据挖掘、人工智能.**陈桂生**,男,1965 年,博士,研究员,主要研究领域为复杂网络、人工智能.

1 引 言

在人类五千年的文明中,60 多年前计算机的发明对人类社会的影响是显著的.计算机科学从电子学中脱颖而出,紧接着是软件工程从计算机体系结构中脱颖而出.为克服以手工作坊方式为主的软件生产引发的软件危机,1968 年在原西德召开的北大西洋公约组织(NATO)会议上,人们第一次提出了软件工程的概念[1].软件脱离硬件作为一个独立产业,必然要关注软件的开发效率、大规模定制、大批量生产和自动化检验.

软件工程领域中,经过多年的工程实践,人们已经意识到软件结构的重要地位,软件工程进行结构分析研究的根本目的就是为工程实践人员开发高质量的软件提供指导.然而随着软件系统规模和复杂程度的改变,大量堆积的底层元素和它们之间错综复杂的交互关系,已逐渐超出了软件开发人员的理解能力,致使系统难以理解和维护,软件开发经常处于失控状态,软件系统中一个极小的错误都可能带来灾难性的后果,甚至引发"雪崩效应"[2-3].同时,由于缺乏相应的方法,传统的软件开发人员很少从整体和全局的观点来审视软件的结构及其演化规律,导致对软件结构的本质缺乏清晰的认知,软件的动态演化和质量难以保证[4-5].在这种情况下,我们需要转变视角来应对我们遇到的问题,复杂网络的研究成果为探索大规模软件系统的结构特性提供了有力的支持,与传统软件方法中只侧重微观层面的设计不同,复杂网络从整体上把握系统,关注内部属性与外部整体特性之间的映射关系和系统整体涌现出的新特性[6],为控制软件复杂性与保证软件质量提供了强有力的工具.

自 2002 年开始,不同领域的研究者对各类软件系统进行研究,将软件系统看作是不同软件单元的自组织、可伸缩、动态演化的复杂系统,发现软件系统的内部结构并不是随机无序的,都表现出复杂系统非常明显的"小世界"和"无标度"特性[7-8],到目前为止,软件系统的拓扑结构分析方面取得了较好的研究进展.但在试验结果的基础上,对软件系统内部结构的复杂特性进行深层的解释却存在挑战,大多数研究者猜测各类现象可能与软件开发过程中的一些规则和决策有关,例如 Myers 认为软件结构中存在一些入度大的 hub 节点是由于软件开发鼓励重用导致的[4],而 Valverde 则认为 hub 节点与软件开发过程降低开发成本的目标(分布式开发、"高内聚、低耦合"特性和鼓励重用等)存在着关联[9].因此,到目前为止,研究者依据软件工程领域的知识,完整系统地对软件系统内部的复杂特性给出令人满意的解释,是一项有待于进一步深入研究的工作.本文选择 RedHat 公司开发的具有高度模块化和内核可配置特性的 eCos 为载体进行拓扑特性及深层挖掘分析,发现其内部结构的共性特征和形成机制,为灵活地配置裁剪提供指导.

本文第 2 节分析网络时代下软件形态和开发方法的改变,对软件网络进行严格定义并提出网络时代软件工程的新观点;第 3 节在不同粒度上抽象出 eCos 的网络模型,从全局和连接特性两个方面进行统计分析,并阐述不同指标在 eCos 系统中的具体含义;第 4 节在统计分析基础上,从节点异质性、局部抱团性和多粒度的网络规模简约 3 个方面对该载体网络进行实证分析与结果验证;第 5 节进行总结.

2 网络时代的软件工程观

回顾软件工程的发展过程,虽然尚未彻底解决"软件危机"的问题,但也极大地推动了软件开发的工程化以及软件产业的迅速发展.通常,人们把软件工程走过的道路,概括为从面向过程、面向对象、面向构件、直到面向网络服务 4 个阶段,如图 1 所示.

图 1 软件工程 40 多年的发展历程

2.1 软件开发设计方法学的变迁

从科学的角度来讲,虽然软件工程发展过程比较发散,但也出现了许多具有里程碑意义的进展.这里从软件开发设计方法的视角来审视软件技术不断演化与发展的过程.

20 世纪 60 年代软件发展的初期,计算机以单机工作为主,软件开发直接从编码(coding)开始,程序规模较小,很少考虑系统的结构.随着软件规模不断增大,软件开发效率低下,质量难以保证,软件危机产生.

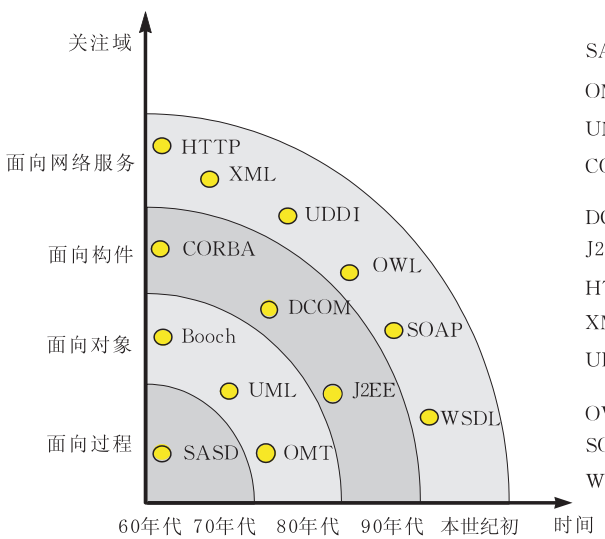
20 世纪 70 年代中后期,Yourdon 和 Constantine 提出了结构化分析设计方法(SASD)[10],将一个复杂的程序分解为函数或过程,强调计算流程中的顺序、循环和条件 3 个基本要素,关注程序开发过程和执行过程,考虑需求分析、结构设计、编码到软件测试整个流程,产生了软件生命周期的概念,形成一整套的软件开发工具,这种面向过程的方法学偏重在编码层面上.

20 世纪 80 年代,随着软件结构易变性以及软件复用性的需求上升,面向对象的程序设计方法逐渐成熟,软件的设计、编码以类和类的对象之间的关系为核心,封装、继承和多态成为 3 个重要特征[11],面向对象软件开发方法和设计工具的兴起和成熟,产生了许多不同的实施方案,比较著名的有 Coad-Yourdon 方法[12]、Booch 方法[13]、Rumbaugh 等的面向对象建模(OMT)方法[14]、Reenskaug 的面向对象的角色分析、合成和构造(OORASS)方法[15]、Jacobson 的面向软件工程对象的方法(OOSE)[16]等等.这就把编码层面的方法学上升到软件设计层面的方法学.

20 世纪末期,人们寻求比"类"的粒度更大、易于复用的构件,软件工程进入了面向构件的阶段,面向构件的方法将构件的接口与实现进行有效的分离,增加了构件交互和重用的能力,软件行业也形成了 3 个代表性的构件技术标准,分别是对象管理组织(OMG)提出的公共对象请求代理体系结构(CORBA)[17]、Sun 公司提出 J2EE(Java2 Platform Enterprise Edition)[18]和 Microsoft 公司提出的分布式构件对象模型(COM/DCOM)[19].面向对象的软件设计方法学上升到软件体系结构的方法学.

进入 20 世纪 90 年代后期,网络的发展改变了人们对软件的需求,软件产业从制造业转变为服务业,无论是软件自身形态,还是软件的工作环境,都是网络化了的.软件作为服务(Software as a Service,SaaS)已经成为一种趋势,软件工程进入了面向网络服务的时代,它以可扩展标记语言(XML)、简单对象访问协议(SOAP)、超文本传输协议(HTTP)、本体 Web 语言(OWL)和统一描述、发现和集成(UDDI)、网络服务描述语言(WSDL)和元模型框架(MFI)等为主要内容[20],软件的开发已经从体系结构的层面上升为了面向网络资源的开发方法学.

因此,在过去的 40 年中,科学家们对软件工程的关注域总是在"与时俱进",对软件的思维方式的不断变化导致了软件开发方法的转变,也催生了软件新技术的不断涌现,如图 2 所示.可以看出随着系统规模和复杂性的提高,个性化和多元化的需求比重越来越大,资源共享与聚合的趋势越来越强,软件的计算模式、应用模式以及软件形态本身都发生了转变,促使软件的生产方式更加强调规模化定制,来适应网络时代灵活、可信和即时的服务.



SASD	Structure Analysis, Structure Design	结构化设计方法
OMT	Object Modeling Technology	对象建模技术
UML	Unified Modeling Language	统一建模语言
CORBA	Common Object Request Broker Architecture	公共对象请求代理体系结构
DCOM	Distribute Component Object Model	分布式组件对象模型
J2EE	Java2 Platform Enterprise Edition	JAVA平台企业版
HTTP	Hypertext Transfer Protocol	超文本传输协议
XML	Extensible Markup Language	可扩展标记语言
UDDI	Universal Description, Discovery and Integration	统一描述、发现和集成
OWL	Ontology Web Language	本体Web语言
SOAP	Simple Object Access Protocol	简单对象访问协议
WSDL	Web Service Describe Language	网络服务描述语言

图 2　软件工程 40 年发展历程中关注域与时俱进

2.2　软件网络及其特征

软件网络是在网络环境下工作的复杂软件系统,软件单元之间的交互关系可以通过复杂网络的拓扑结构进行描述,更进一步,其拓扑结构和软件行为可以进行动态演化.大量研究表明软件系统的内部结构并不是随机的,而是具有小世界和无标度等复杂网络的基本特性[2,4,9].为了对软件网络有更深入的认识,下面给出软件网络的形式化描述.

软件网络 $G=(V,E)$ 是一个由复杂的软件系统抽象而出的网络模型. 其中,$V=\{v_1,v_2,\cdots,v_n\}$ 称为节点集合,表示软件系统中的软件单元,也可称为智能体(Intelligent Agent)或者主体(Active Agent),它们的大小若用粒度表示的话,对象、类、构件、子系统和系统等是逐步增大的. $E=\{(v_i,v_j)\mid i,j=1,2,\cdots,n\}$ 称为边集,表示每个软件单元间的相互作用,最常见的交互形式是消息传递、数据交换和程序调用. 消息传递是单向的,数据交换是双向的,程序调用的本质仍然是数据交换,因为数据和程序具有相同的表现形式.

从上述描述中可以看出软件网络是一个包含大量个体及个体间相互作用的复杂系统,是把软件单元间的某种关系抽象为个体(顶点)以及个体间相互作用(边)而形成的描述这种复杂关系的图. 但与图的研究有所不同,软件网络更侧重于从实际网络现象之上抽象出普适的网络拓扑特性,用于研究复杂软件系统的网络模型与动力学演化机制.

软件网络具有如下的基本特征:

(1) 复杂网络环境与网络拓扑性质和行为密不可分;

(2) 组成单元更自主、耦合更松散;

(3) 需求主导交互,相互协同,彼此适应,行为涌现;

(4) 可伸缩,可重组,可替代,可持续运行.

如果我们不从软件工程历史发展的角度看,仅仅把这些功能强大的可编程的软件单元放入互相连接的通信网络中的一个个节点上,节点完成各种各样的计算与处理功能,连接主要完成通信、数据交换和协同功能. 这样一来,网络通过软件促进了用户之间的资源聚合、信息共享和协同工作,这也许就形成了网络时代软件工程的新观点.

3 多粒度软件网络的特性分析

复杂网络研究中,不依赖于节点的具体位置和边的具体形态所能表现出来的性质就是网络的拓扑性质(topology property),相应的结构叫做网络的拓扑结构. 研究表明软件系统的复杂性主要取决于软件单元之间错综复杂的交互关系,软件系统一般呈现出体系结构的整体特征,而且不同层次可以进行不同粒度的抽象.

本文选取 RedHat 公司推出的嵌入式可配置实时操作系统 eCos(embedded Configurable operating system)为载体进行拓扑特性的分析. eCos 系统最大的特点是模块化和内核可配置,适合深度嵌入式应用,支持多种平台,主要分 package(包),包含多个 component(构件),而组件则会有多个 option(选项)或 interface(接口),它们之间的关系有 parent(父)、requires(依赖)、active_if(控制条件)等,如图 3 所示. 选用 eCos 系统作为我们的载体网络进行研究,是因为首先它是一款开放源代码的系统,节点及节点间交互关系易获取,其次 eCos 系统具有丰富的结构特性(可配置性、可裁减性、可移植性和实时性)和灵活的可配置能力,适合在不同的抽象层次实现系统组装,为不同粒度抽象软件网络提供了支持.

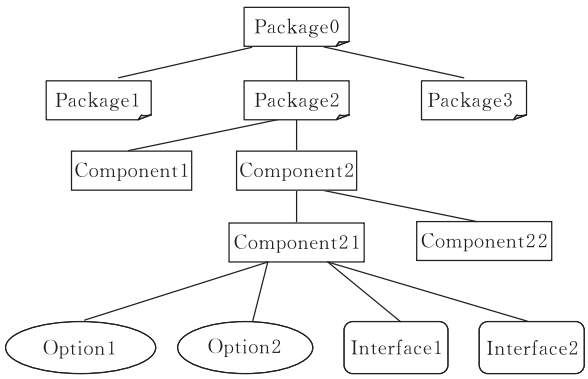
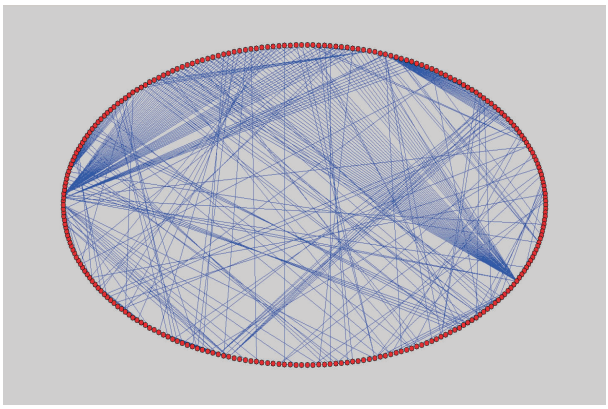


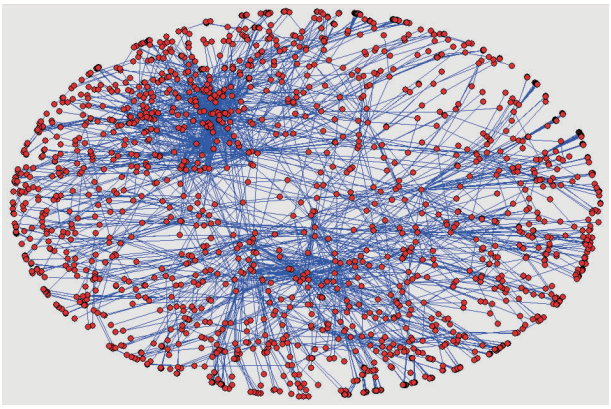
图 3 eCos 构件系统不同粒度的分类

我们开发了一个专门的网络抽取和分析工具,根据语法关键字对源代码进行解析和过滤. 本文选用的 eCos 源代码包含 284 个文件,存放在 packages 目录下,它把相同类型的包组织在一起,packages 目录下的每个子目录都是一个类,每个类又包含若干包,解析源代码中的语法关系(parent、requires 和 active_if)定义为节点间的边关系,从而从包层(package)和系统所有单元(包括 package、component、option 和 interface)两个粒度上提取网络模型,从不同的尺度上反映 eCos 系统内部的结构与交互关系,所有抽取原则不考虑连接方向. 解析 eCos 源码文件的所有 package 包,得到一个包含 326 个节点的软件网络,它由 58 个相互独立的连通子网络组成,其中最大的连通子图包含 266 个节点,如图 4(a)所示. 对所有单元进行解析,得到一个拥有 2344 个节点的软件网络,该网络由 80 个相互独立的连通子网络组成,其中最大连通子图由 2225 个节点组成,如图 4(b)所示.

首先对不同粒度上抽取的最大连通子图进行全局统计特性分析,如表 1 所示.



(a) eCos-Package层的最大连通子网络



(b) eCos-all层的最大连通子网络

图　4

表 1　不同粒度最大连通子图的全局统计参数

系统层次	规模	边数	平均最短路径长度	网络直径	平均聚集系数	幂率指数
eCos-Package	266	446	4.457285	10	0.347754	1.34924
eCos-all	2225	3352	6.200151	14	0.608391	1.90339

从表 1 中可以看出,随着抽取粒度的降低,网络规模增加大约 10 倍左右,平均路径长度和网络直径的增长不太大,但平均积聚系数变化较大,充分体现了复杂网络的“小世界”和“高内聚、低耦合”的普适特性[7,21],同时随着网络规模的增长,呈现出一定的无标度特性,但是幂率指数比较低.统计发现,幂率分布广泛存在于自然界与日常生活的众多领域,尽管网络类型不同,但它们的幂率指数 γ 大多在[2,3]范围之内[8-9].由于软件属性很大程度上取决于组成单元的属性及其之间的交互关系,那么对不同粒度上的网络模型而言,节点间的连接特性是否具有相似性呢?下面我们从不同粒度软件网络的节点度分布[22]、最短路径频度分布[22]、聚集系数-度相关性[22]和富人俱乐部特性[23]进行分析.

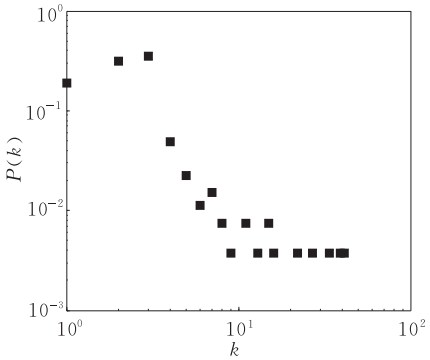
3.1　度分布

网络中节点度的分布情况可以用度分布函数 $P(k)$来描述,即网络中度为 k 的节点所占的比例,或任选一个节点的度为 k 的概率.$P(k)$的期望被称为网络的平均度(mean degree),如式(1)所示.

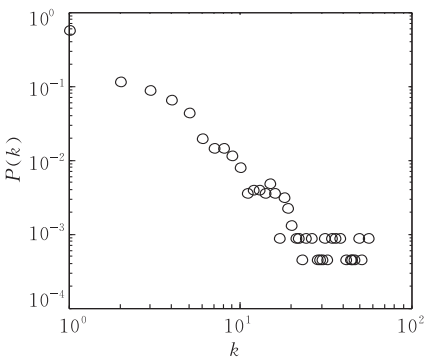
$$\langle k\rangle=\sum_{v\in V}\deg(V) \tag{1}$$

度分布函数反映了系统结构的统计特征,对于软件系统而言,度分布刻画了网络中每个节点的连通性(connectivity),并从系统角度反映了节点的重用度和复杂度,因此,度分布可以用来表征系统结构的不均匀性,定性分析结构信息的不确定性.

分析该 eCos 网络中不同粒度上最大连通子图的度分布情况(如图 5 所示),可以发现,虽然规模不同,但是它们大致都服从幂率分布,eCos-all 层的度分布曲线在双对数坐标下非常接近于直线,而 Package 层的度分布除截断开始的部分节点,也可以呈现比较明显的“无标度”和“重尾”特性.



(a) eCos-Package层最大连通子网络的度分布



(b) eCos-all层最大连通子网络的度分布

图　5

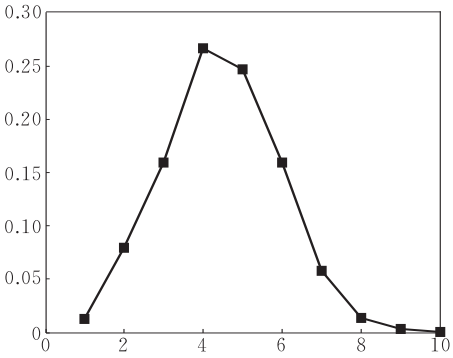
3.2 最短路径频度分布

网络中所有节点间最短路径长度的平均值(又称为网络平均距离),反映网络中节点的分离程度.任意两个节点 $v_i, v_j \in V$ 的最短路径长度为 d_{ij},平均路径长度如式(2)所示.

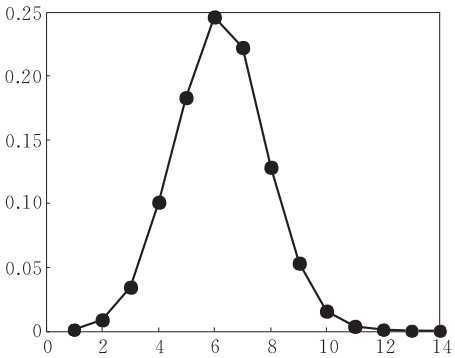
$$L=\frac{1}{n(n-1)}\sum_{v_i\neq v_j} d_{ij} \tag{2}$$

最短路径描述了网络中节点的分离程度(separation),即小世界特性.对于软件网络而言,最短路径反映了系统间消息传递的设计与优化,在节点间通信代价的评估与控制、系统响应能力方面有重要的意义.

分析该 eCos 网络中不同粒度上最大连通子图的最短路径频度分布情况(如图 6 所示),可以发现,从宏观上看,虽然网络规模扩大约 10 倍,但是网络的最短路径增长不大(Package 层为 10,all 层为 14),体现了比较明显的"小世界"特性,同时在线性坐标系下,两个粒度上的频度分布曲线非常相似,充分说明不同粒度下网络节点的连通性比较好.



(a) eCos-Package层网络的最短路径频度分布



(b) eCos-all层网络的最短路径频度分布

图 6

3.3 聚集系数-度相关性

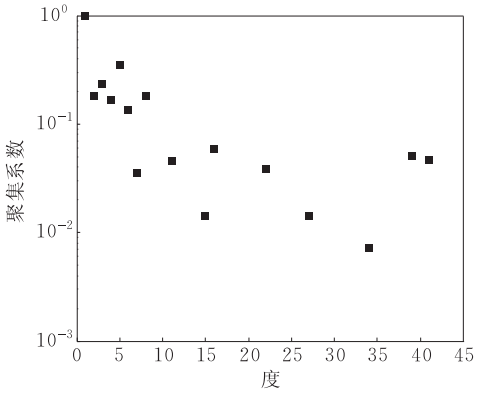
网络中任意节点 $v_i \in V$,令 $k_i=\deg(v_i)$,若 v_i 的 k_i 个邻居节点间的连接数为 E_i,则 v_i 的聚集系数如式(3)所示.

$$c_i=\frac{2E_i}{k_i(k_i-1)} \tag{3}$$

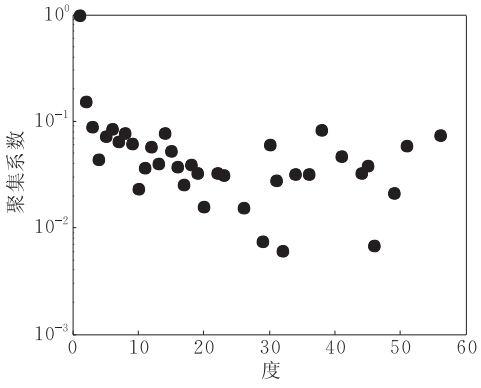
式中,$E_i=\frac{1}{2}\sum_{j\neq m} e_{ij}e_{im}e_{jm}$,若节点 v_i 的邻居节点 v_j,v_m 间存在连接,则有 $e_{ij}e_{im}e_{jm}=1$;否则 $e_{ij}e_{im}e_{jm}=0$.

该参数可以反映网络的内聚程度和传递性(transitivity).聚集系数越大,网络的传递性就越好.对于软件系统而言,聚集系数和度的相关性更多地代表了不同粒度的软件实体中组成单元的内聚程度以及系统组织分层的趋势.

分析该 eCos 网络中不同粒度上最大连通子图的聚集系数与度的散点分布情况(如图 7 所示),可以发现,在两个粒度上,平均聚集系数对度分布都接近于直线,Package 层更为明显,这个结果说明在两个粒度的网络结构中存在着某种层次模块的结构,与图 3 中所示的 eCos 系统的层次分类相互验证.



(a) eCos-Package层网络聚集系数与度的散点图



(b) eCos-all层网络的聚集系数与度的散点图

图 7

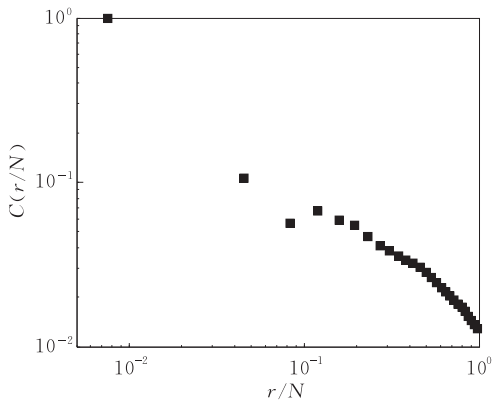
3.4　富人俱乐部特性

网络中少量的节点具有大量的边,这些节点称为“富节点”(rich nodes),它们倾向于彼此之间相互连接,构成“富人俱乐部”(rich-club)特性[23].该特性通过连通性来刻画,表示网络中前 r 个度最大的节点之间,实际存在的边数 L 与这 r 个节点之间总的可能存在的边数 $r(r-1)/2$ 的比值,如式(4)所示.

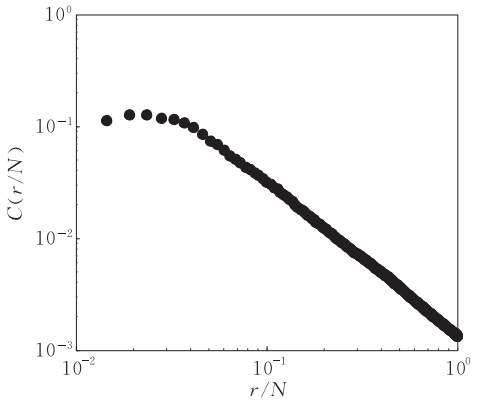
$$C\left(\frac{r}{N}\right)=\frac{2L}{r(r-1)} \tag{4}$$

该参数刻画了网络中与连接度高的节点更偏好于与度值高的节点相连还是与度值低的节点相连,当连通性为 1 时,那么前 r 个最富的节点可以组成一个完全连通的子图.对于软件系统而言,该特性有助于分析不同类型软件实体之间的协作关系,由于软件功能的层次性,复杂的软件实体更倾向于由相对简单的实体构成,体现软件的构造性原则.

分析该 eCos 网络中不同粒度上富人俱乐部特性,如图 8 所示.可以发现,网络中存在极少量的高度连接节点,绝大多数节点的度比较低,符合“二八定律”,分析具体的原因,在 eCos 配置的过程中,连接度高的节点更倾向于临近层中连接度低的节点进行交互.



(a) eCos-Package层网络的富人俱乐部特性分布



(b) eCos-all层网络的富人俱乐部特性分布

图　8

从上述的分析结果,可以看出该载体网络在不同的粒度上,统计特性分布曲线十分类似,进一步体现了软件网络不同层次或粒度上具有比较相近的结构特性.事实上,在生物领域中也有类似的研究,生物学家对各种生物的分子、细胞、生物体等不同粒度上的结构和功能进行研究,挖掘不同层次中结构特性的异同,对解释生物体中各个层次的结构和动力学的涌现行为和特征提供了支持[24-25].

4　软件网络结构特性的应用

软件系统的成功开发离不开大量可复用的构件,那么对于嵌入式构件系统而言,如何通过构件网络的拓扑结构特性更好的管理、维护这些系统,如何通过构件单元在网络中拓扑位置的差异,确定每个单元的重要程度或者寻找 20% 占据重要意义的构件单元,指导软件的可配置、可裁剪特性,达到资源的最小负载,如何从大规模的构件网络单元之间的交互关系,识别“微观”层面上构件内部紧密连接,而形成的高积聚局部社区以及局部社区之间通过松散耦合而形成的“宏观”社区结构,为提供可裁剪、可选择的系统内核和应用中间件的功能提供指导,如何在系统规模与复杂度剧增的情况下,通过构件单元之间多层次、多粒度中呈现的自相似特征,进行网络规模的简约与提升,更加准确地预测软件系统的整体行为.下面我们从复杂系统和复杂网络的角度,将构件单元的属性与其拓扑特性结合起来研究,提供了一个整体和全局的视角.

4.1　网络拓扑中心挖掘

“结构决定功能”是系统科学的基本观点[26].对于一个规模巨大的网络拓扑而言,我们很难快速地获得全局信息,但是由于网络中节点的异质性,使得每个节点在网络中的重要程度不同.现有量化复杂网络中心性的指标主要包括度数中心性[27]、介数中心性[28]、接近度中心性[29]、特征向量中心性[30](例如 Pagerank)等,但上述的各类指标都是侧重于单一的拓扑信息,挖掘的结果存在不确定性.为了更加合理地找到复杂网络的拓扑中心,本课题组基于数据场思想的启发,提出复杂网络的一种新的度量指标——拓扑势,并用于挖掘网络的拓扑中心,相关的

理论基础见文献[31-32],这里就不详细描述.基于拓扑势的度量指标对不同粒度上 eCos 网络进行挖掘,提取网络中前 10 位中心节点和它们的拓扑势值,结果如表 2 所示.

表 2 不同粒度网络中前 10 位中心节点及拓扑势值

eCos-Package		eCos-all	
函数名	拓扑势值	函数名	拓扑势值
CYGPKG_ERROR	32.757032	CYGPKG_ISOINFRA	143.35
CYGPKG_IO_SERIAL	31.273861	CYGPKG_ERROR	124.05
CYGPKG_IO_ETH_DRIVERS	24.726671	CYGPKG_IO_SERIAL	120.77
CYGPKG_IO_FLASH	22.239633	CYG_HAL_STARTUP	109.64
CYGPKG_HAL	18.211979	CYGBLD_GLOBAL_OPTIONS	106.36
CYGPKG_HAL_POWERPC	17.990894	CYGNUM_HAL_RTC_CONSTANTS	98.365
CYGPKG_IO	17.191863	CYGHWR_MEMORY_LAYOUT	97.517
CYGPKG_ISOINFRA	15.368653	CYGSEM_HAL_ROM_MONITOR	95.908
CYGPKG_HAL_SH	13.611014	CYGPKG_IO_SERIAL_DEVICES	94.781
CYGPKG_POSIX	11.672913	CYGNUM_HAL_VIRTUAL_VECTOR _COMM_CHANNELS	87.586

通过第 3 节的特性分析,我们可以看到在 Package 层和 all 层的节点交互关系与分布规律存在着极大的相似性,因此,这里选取 Package 层网络的拓扑结构进行详细的阐述,可视化 Package 网络拓扑结构,其中节点大小表示节点在网络中的中心程度,节点越大,越靠近网络的中心,将前 10 位重要节点标记为黄色,并标注相应的函数名,如图 9 所示.

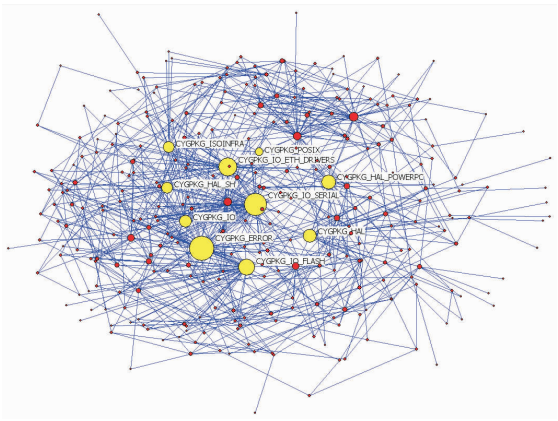


图 9 eCos-Package 层网络中心性的拓扑结构可视化

为了与其它的中心性度量指标进行比较,统计分析 Package 层网络的度数、介数、接近度和 Pagerank,进行归一化处理后,比较各种指标的变化趋势,如图 10 所示.可以发现,在介数指标下,存在大量的节点为 0 的情况,不能很好地体现出节点之间的差异性;在节点度数的指标下,存在大量节点度数相同的情况,判断结果失效;在接近度的指标下,仅仅考虑节点本身的特性,忽略了节点间边的关系,统计值变化差异性很大;在本实验中,拓扑势方法和著名的 Pagerank 存在着较相似的变化趋势,但后者的变化趋势在尾部位置比较平缓.因此,采用拓扑势的方法,综合考虑了节点和边的总体特性,可以比较准确地显示网络拓扑中节点间位置的差异和节点本身的连接特性.

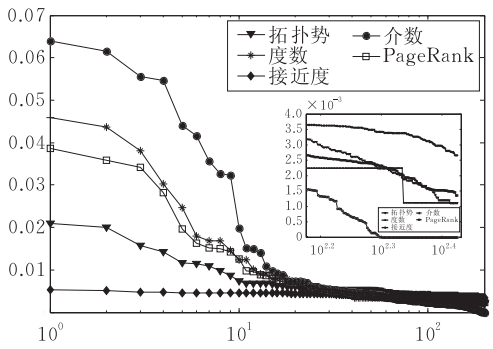


图 10 eCos-Package 层网络拓扑结构中心性指标对比图

4.2 社区结构的提取

软件工程的发展历程可以看出,从结构化程序设计到面向对象开发方法再到面向构件的开发方法,不同粒度的软件实体,系统的设计一直都遵循着"高内聚,低耦合"的原则[4,21],因此,为了使 eCos 构件系统具有高度灵活的可配置性,系统应具有良构的模块化结构,在一定的上下文中可以被视为独立的逻辑单元,实现构件内部功能比较单一,便于信息和功能的识别、封装和描述.而复杂网络中呈现出明显的社区结构,可以看作不同粒度的内聚模块组成的层次结构,因此对软件网络进行社区结构的发现与提取有助于从理论分析角度得到系统的理想结构,为软件开发和设计人员提高软件模块化程度提供指导.

目前网络模块性的基准计算方法是 Newman 等提出的模块度[21],如式(5)所示.

$$Q=\sum_{i=1}^{k}(e_{ii}-a_i^2)=Tre-\|e^2\| \tag{5}$$

对 Package 层网络计算最优划分结果,将该网

络划分为若干模块后,模块内部具有紧密的连接而模块之间存在少量边的连接,一般来说,具有社区结构的真实世界网络的模块值在 0.3～0.8 之间.计算 Package 层网络的模块度曲线,如图 11 所示,其中最大值 0.665 对应的社区划分个数是 16,即网络划分成 16 个社区时,理论上可以得到最优的划分结果.

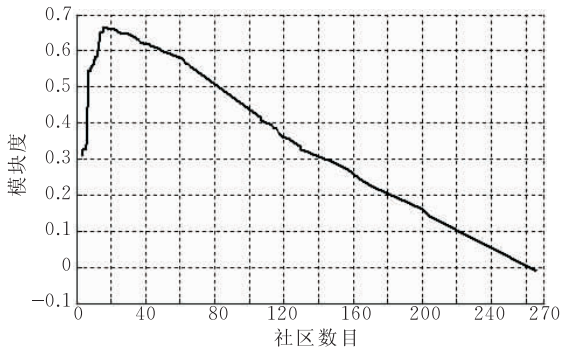


图 11　eCos-Package 层网络的模块度曲线

按照模块度的划分数目,可视化该网络的社区结果如图 12 所示,图中不同社区的节点通过不同的颜色和形状来区别,并对区域边界进行标注.

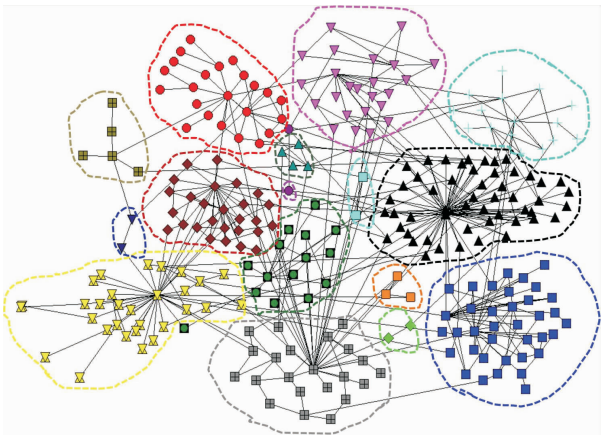


图 12　eCos-Package 层网络的社区划分结果

随机抽取 package 网络的社区 1 进行分析,发现该社区由 23 个 package 节点组成,局部可视化该网络如图 13 所示,在嵌入式构件系统中节点的名称可以反映其功能,从社区 1 的组成来看,除了 2 号(CYGPKG_INFRA)和 132 号(CYGPKG_DEVICES_WATCHDOG_ H8300_H8300H)外,都具有 CYGPKG_HAL 前缀,说明 CYGPKG_HAL_* 构件包中的 package 节点之间的引用关系较为密切,相比其它包中的引用关系具有较高的耦合性,进一步分析两个特殊的节点,发现 CYGPKG_HAL 节点通过与两个节点之间的连接,达到与其它社区内节点的交互.因此,根据社区划分的结果,可以帮助开发人员理解软件系统的结构与组成元素的相互关系,有效地指导软件包的设计与归类,更进一步为结构的优化提供有益的参考.

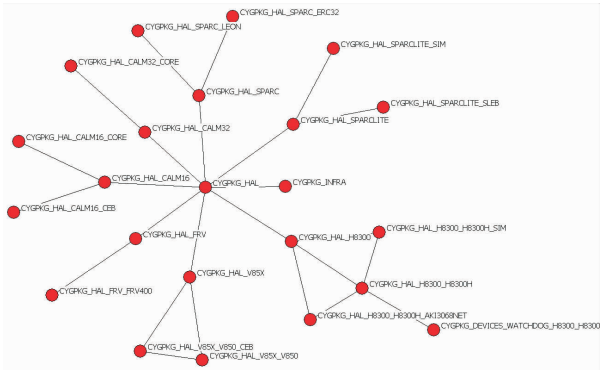


图 13　eCos-Package 层网络局部社区的拓扑图

4.3　多粒度网络规模的简约

现实世界中复杂网络规模巨大,节点的数目往往有几十万甚至几百万,节点间的相互联系和作用关系错综复杂,卡内基・梅隆大学的软件工程研究所报告中指出,未来的软件系统代码行规模将达到数亿,超大规模的软件系统(ultra-large-scale)对目前的开发实践提出了挑战[33].这里的多粒度可以理解为在保持复杂网络基本特性的基础上,实现网络中节点和边的数量规模的简约.在 4.2 节对网络社区的分析基础上,我们将每个社区抽象为一个虚拟节点,将社区之间的连接抽象为虚拟节点间的无权无向连接边,从而得到一个由虚拟社区节点组成的简约网络,这里,每个虚拟节点用社区中拓扑势值最大的节点作为代表,根据上述思想,递归地对 eCos-all 网络进行约简,形成自底向上多粒度上的网络视图,最终得到包含 16 个虚拟节点的简约网络,如图 14 所示.

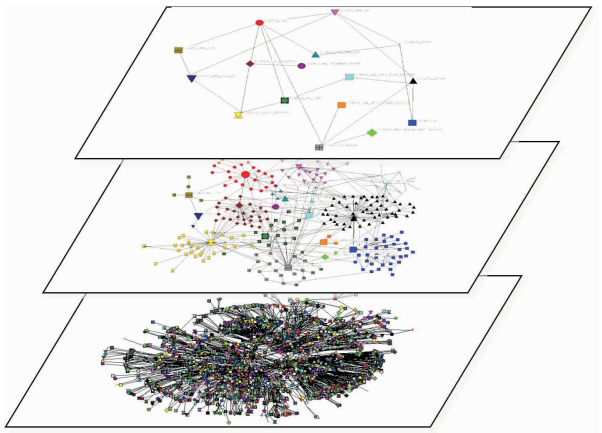


图 14　eCos 网络的多粒度规模简约视图

可以看出,在进行复杂网络拓扑映射时,粒度的

大小在一定程度上决定了拓扑的规模.一般而言,粒度越细,网络越微观,个性化程度就越多;反之,粒度越粗,网络越宏观,共性就越明显.因此,从不同刻面上对网络进行抽象,发现隐藏在拓扑结构中的共性和个性信息,根据目标硬件平台的实际需要,在不同的粒度上选择使用相应的构件或者移除不必要的构件,方便嵌入式应用软件的设计,提高复用的灵活性,实现完整的嵌入式系统.

5 结 论

从复杂系统和复杂网络的角度来重新审视软件,从整体和全局的角度来探索和发现复杂软件系统的结构特性、演化规律和由此产生的行为特征,有助于科学、全面地认识和理解软件的本质特征,为网络时代下的软件工程的发展提供了一种新的视角.本文通过对嵌入式构件系统从多粒度上进行抽象,统计它们的全局特性,挖掘网络拓扑中隐藏的知识信息,量化其分析结果,对提高软件质量、减少软件测试与维护费用有着重要的意义.

参 考 文 献

[1] Perlis A J. Software engineering: Report on a conference sponsored by the NATO Science Committee. Scientific Affairs Division NATO, 1968: 135-138

[2] Valverde S, Sole R V. Hierarchical small worlds in software architecture. Santa Fa Institute, Santa Fe, New Mexico, USA: Technical Report SFI/03-07-044, 2003

[3] He Ke-Qing, Li Bing. Software Network. Beijing: Science Press, 2008

[4] Myers C R. Software systems as complex networks: Structure, function, and evolvability of software collaboration graphs. Physical Review E, 2003, 68(4): 046116

[5] Mens T, Wermelinger M, Ducasse S. Challenges in software evolution//Proceedings of the IWPSE 05. Lisbon, 2005: 13-22

[6] Mitchell M, Newman M. Complex systems theory and evolution//Pagel M. Encyclopedia of Evolution. New York: Oxford University Press, 2002

[7] Watts D J, Strogatz S H. Collective dynamics of 'small-world' networks. Nature, 1998, 393(6684): 440-442

[8] Barabási A L, Albert R. Emergence of scaling in random networks. Science, 1999, 286(5439): 509-512

[9] Valverde S, Cancho R F, Solé R. Scale free networks from optimal design. Europhysics Letters, 2002, 60(4): 512-517

[10] Yourdon E, Constantine L L. Structured Design: Fundamentals of a Discipline of Computer Program and System Design. New Jersey: Prentice-Hall, 1979

[11] Yang Fu-Qing, Mei Hong, Lu Jian, Jin Zhi. Some discussion on the development of software technology. Acta Electronica Sinica, 2002, 30(12A): 1901-1906(in Chinese)
(杨芙清, 梅宏, 吕建, 金芝. 浅论软件技术发展. 电子学报, 2002, 30(12A): 1901-1906)

[12] Coad P, Yourdon E. Object Oriented Analysis. 2nd Edition. New Jersey: Yourdon Press, 1990

[13] Booch G. Object Oriented Design with Applications. California: Benjamin-Cummings, 1990

[14] Rumbaugh J R, Blaha M R, Lorensen W, Eddy F, Premerlani W. Object-Oriented Modeling and Design. New Jersey: Prentice Hall, 1990

[15] Reenskaug T, Andersen E, Berre A et al. OORASS: Seamless support for the creation and maintenance of object oriented systems. Journal of Object Oriented Programming, 1992, 5(6): 27-41

[16] Jacobson I. Object Oriented Software Engineering: A Use Case Driven Approach. New York: Addison-Wesley Professional, 1992

[17] OMG. Object Management Architecture Guide. 3rd Edition. USA: John Wiley & Sons, 1995

[18] SUN Microsystem. Java 2 platform enterprise edition specification. Version 1.3, 2001

[19] Microsoft Corporation. Distributed component object model protocol——DCOM/1.0, 1996

[20] Mei Hong, Cao Dong-Gang, Yang Fu-Qing. Development of software engineering: A research perspective. Journal of Computer Science and Technology, 2006, 21(5): 682-696

[21] Girvan M, Newman M E J. Community structure in social and biological networks. Proceedings of the National Academy of Sciences, 2002, 99(12): 7821-7826

[22] Barabási A L, Albert R. Statistical mechanics of complex networks. Reviews of Modern Physics, 2002, 74(1): 47-97

[23] Zhou S, Mondragon R J. The rich-club phenomenon in the Internet topology. IEEE Communication Letters, 2004, 8(3): 180-182

[24] Brown J H, Gillooly J F, Allen A P et al. Toward a metabolic theory of ecology. Ecology, 2004, 85(7): 1771-1789

[25] Oltvai Z N, Barabási A L. Life's complexity pyramid. Science, 2002, 298(5594): 763-764

[26] Xu Guo-Zhi et al. Systems Science. Shanghai: Shanghai Scientific & Technical Education Publishing House, 2000(in Chinese)
(许国志主编. 系统科学, 上海: 上海科技教育出版社, 2000)

[27] Bonacich P. Factoring and weighting approaches to status scores and clique identification. Journal of Mathematical Sociology, 1972, 2(1): 113-120

[28] Freeman L C. Centrality in social networks: Conceptual clarification. Social Networks, 1979, 1(3): 215-239

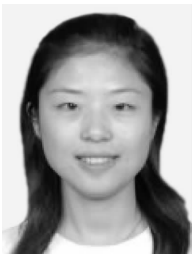
[29] Wasserman S, Faust K. Social Network Analysis: Methods and Applications. Cambridge: Cambridge University Press, 1994

[30] Bonacich P, Lloyd P. Eigenvector-like measures of centrality for asymmetric relations. Social Networks, 2001, 23(4): 191-201

[31] He Nan, Li De-Yi. Evaluate nodes importance in the network based on data field theory//Proceedings of the International Conference on Convergence Information Technology. Korea, 2007: 1225-1234

[32] Han Yan-Ni, Li De-Yi. A novel measurement of structure properties in complex networks//Proceedings of the Complex 2009. Shanghai, 2009: 1292-1297

[33] Northrop L, Feiler P, Gabriel R P et al. Ultra-large-scale systems: The software challenge of the future. Pittsburgh: Software Engineering Institute, 2006



HAN Yan-Ni, born in 1981, Ph. D. candidate. Her research interests include complex network, software engineering.

LI De-Yi, born in 1944, Ph. D. supervisor, academician of Chinese Academy of Engineering. His main research interests include complex network, data mining, and artificial intelligence.

CHEN Gui-Sheng, born in 1965, Ph. D., researcher. His main research interests include complex network, artificial intelligence.

Background

In recent years there has been a strong upsurge in the study of complex network in many disciplines. Software is composed of many interacting subsystems or units at many different levels of granularity and the interactions or collaborations of these subsystems can be represented in the form of an abstract complex network.

This paper aims to address the problem of abstracting topology from software systems at different granularities. All of these aspects are helpful to reveal the underlying properties besides the basic statistics parameters such as degree, clustering coefficient of complex systems, especially we hope to exploit the hub node, community discovery and network reduction in the embedded component network. Not only does this result inspire the configuration and portability of embedded component systems, but it also has practical value in building more stable systems and achieving the least resource load. Furthermore, it can contribute to the evaluation of development cost and scheduling.

This work is supported by the National Grand Fundamental Research 973 Program of China with the title "Requirement Engineering — Fundamental Research on the Software Engineering with Complexity System" under grant No. 2007CB310800 and the National Natural Science Foundation of China with the title "Networked Data Mining" under grant No. 60496323. The result in the paper belongs to the part of networked data mining of software network, which is to find important nodes among large network, community structure discovery and complex network reduction, etc. All of these aspects are helpful to reveal the underlying properties besides the basic statistics parameters such as degree, clustering coefficient of software systems, especially the complex network consisted of abundant software resources among the whole Internet. In addition, the evaluation of node importance bas been studied based on topology potential, which is able to obtain an accurate ranking of vital nodes compared with classic measures. And the results are published in the Journal of software engineering and the 2009 International Conference on Complex Sciences: Theory and Applications, etc.