

# 时态实体依赖关系与度量方法研究

傅 妤 婧 张 俊 王 毅 恒

(大连海事大学 辽宁 大连 116000)

**摘 要** 实体间存在各种各样的依赖关系,尤其是在软件开发过程中,软件实体间的依赖关系对软件的变更影响分析以及风险分析等都具有重大影响。依赖图作为最常用的依赖关系表示方法,其节点与边的定义与属性计算不尽相同,且大部分方法中并没有考虑到节点与边的时态属性。针对时态实体依赖图,文中系统地提出了时态实体依赖关系的形式化定义并分析了其特性,然后分析了时态实体依赖图的节点中心性、节点重要性、节点依赖度和边的重要性等 4 个度量指标,同时,针对 MAVEN 数据集分析了上述各个指标随时间变化的规律。

**关键词** 依赖关系,时态实体依赖图,度量

**中图分类号** TP301 **文献标识码** A

## Research on Temporal Entity Dependencies Relation and Measurement Method

FU Yu-jing ZHANG Jun WANG Yi-heng

(Dalian Maritime University, Dalian, Liaoning 116000, China)

**Abstract** All kinds of dependencies exist among entities. Especially in the process of software development, the dependencies between software entities has a big impact on the impact analysis of software and risk analysis. Dependency graph is the most commonly used dependency representation method, the definition of nodes and edges is different from attribute computation, while the temporal properties of nodes and edges are seldom taken into account in existing dependency graph methods. This paper presented formal definition and analysis of temporal characteristics of temporal dependencies, and also analyzed the importance of four measures including node center, node importance, node dependency and edge importance. Finally, test dataset was extracted from MAVEN data, and experimental results show regulation of indicators varying with the time.

**Keywords** Dependency relation, Temporal entity dependency graph, Measurement

依赖关系是实体间最重要的关系类型,无论是在软件开发过程中,还是在日常生活中,依赖关系随处可见并且发挥着举足轻重的作用。

在软件系统中,某一模块发生变更会导致许多相关模块也发生变更,甚至会波及到整个系统,这就是变更的波及效应<sup>[1]</sup>,而这种波及效应就是由实体间的依赖关系引起的。

实体依赖关系是实体变更影响分析以及风险分析的基础。在软件系统中,由于实体间存在依赖关系,使得任意一段代码的修改或删除都有可能引起其他相关联部分的变更;同样地,软件构件或者模型的增加与修改也都可能会导致相关联的构件、模型,甚至需求设计等实体的变更。

国内外对于实体依赖关系的研究很多,也存在许多实体依赖关系的表示方法,但是很多实体依赖关系的研究都忽略了时态属性对依赖关系的影响。

在时态实体依赖性分析的过程中,需要首先明确时态实体间的依赖关系类型,确定当前实体间存在的依赖关系是否合理,然后再针对实体之间不合理的依赖关系,分析时态实体的重要性与时态实体之间依赖关系的重要性与强度,以最小的修改代价对部分不合理的依赖关系进行修改和删减,使时

态实体依赖图中的时态实体依赖关系更加合理。

因此,本文在以往的实体依赖关系研究的基础上,结合时态属性以及节点与边属性的度量方法,首先提出了对时态实体依赖关系的分类与形式化,进而提出了时态实体依赖图的定义,并提出时态实体依赖图中节点中心性、节点重要性、节点依赖度与依赖重要性等概念与度量公式,并针对 MAVEN 数据集分析了上述各个指标随时间变化的规律。

本文第 1 节提出 6 种依赖关系的定义并且分别对这 6 种依赖关系进行形式化分析,再分别分析 6 种依赖关系的基本性质与时态性;第 2 节对时态实体依赖图进行定义,然后提出 4 种时态实体依赖图节点与边的属性及其度量公式;第 3 节利用 Maven 数据集中的数据,对第 2 节提出的属性进行实验分析,以验证其可行性;最后总结全文。

## 1 依赖关系的基本概念

### 1.1 依赖关系的基本定义

依赖关系的定义是由 Stevens 等于 1974 年首先提出的,其定义“依赖关系是衡量软件系统中一个构件对另一个构件依赖的程度。构件之间的关联越少、越简单,便越容易不借助

傅妤婧(1992—),女,硕士生,主要研究方向为时态数据库理论与应用,E-mail:824437084@qq.com(通信作者);张俊(1971—),男,博士,教授,主要研究方向为数据库理论与技术、数据库信息检索、智能信息处理和语义网;王毅恒(1993—),男,硕士生,主要研究方向为时态数据库理论与应用。

其他构件对一个构件进行分析了解”<sup>[2-3]</sup>。

根据以上定义,可以定义实体依赖关系如下。

**定义 1(实体依赖关系)** 假定  $S$  为所有实体的集合,记作:  $S = \{C_1, C_2, \dots, C_n\}$ ,  $n$  为集合中实体的数目。依赖关系  $D$  为定义在集合  $S$  上的二元关系,设存在两个任意的实体  $C_i$  和  $C_j$ ,且  $C_i \in S, C_j \in S$ ,如果实体  $C_i$  被修改、删除或升级后,实体  $C_j$  也会随之出现相应的变化,则称实体  $C_j$  依赖于实体  $C_i$ ,记为:

$$D = \{\langle C_j, C_i \rangle | C_i \in S, C_j \in S\}$$

## 1.2 依赖关系的分类

不同的研究领域和研究方向对依赖关系的划分各不相同。从研究数据的角度,依赖关系可以大致分为:基于代码级的依赖关系研究和基于进程级的依赖关系研究。前者利用软件的代码进行依赖关系的研究,后者则是实体层面的依赖关系研究。本文的研究则主要基于后者。

基于代码级的依赖关系,最常见的有属性所引起的依赖,称为数据依赖,记作  $R_{DD}$ ;由操作所引起的依赖称为方法依赖,记作  $R_{MD}$ <sup>[4]</sup>。除此以外,许多学者还提出了任务间的数据依赖、语义依赖、上下文依赖、参数依赖、分歧依赖、通讯依赖等依赖关系<sup>[5-7]</sup>。

而实体间的依赖关系,则由于实体的定义范围较广、涉及领域较多等原因,没有较统一的定义。Arias 等在其综述性论文中,将依赖关系大致分成了 3 类:行为依赖、结构依赖和可追踪性依赖<sup>[8]</sup>;Wilde 等在针对基于构件的可复用软件系统的研究中将依赖关系分成了定义依赖、调用依赖、函数依赖和数据流依赖<sup>[9]</sup>。

本文对实体间可能存在的依赖关系进行了分析归纳,提出了通用性更强的实体依赖关系的分类,为了能够进行依赖性分析,对实体之间的依赖关系进行推理和筛选,并进行形式化表示,另外还针对不同的依赖关系类型,对实体依赖关系的基本性质以及时态性进行分析。实体依赖关系分类如下。

### (1) 协作依赖

协作依赖(cooperation dependency),简称为 cpd。其表示在时间区间  $T$  内,一个实体的行为需因另一个实体的行为的存在而存在。

形式化:假设  $A$  与  $B$  为两个实体,则实体  $A$  与实体  $B$  存在协作依赖关系,当且仅当:

在时间区间  $T$  内,

$$\forall A \forall B (A \wedge B) \Rightarrow A \xrightarrow{\text{cpd}} B$$

### (2) 排斥依赖

排斥依赖(exclusion dependency),简称为 exd。其表示在时间区间  $T$  内,当一个实体的行为存在时,另一个实体的行为不可存在,否则将产生排斥从而发生错误。

形式化:假设  $A$  与  $B$  为两个实体,则实体  $A$  与实体  $B$  存在排斥依赖关系,当且仅当:

在时间区间  $T$  内,

$$\forall A \forall B ((A \wedge \neg B) \vee (\neg A \wedge B)) \Rightarrow A \xleftrightarrow{\text{exd}} B$$

### (3) 时间依赖

时间依赖(time dependency),简称为 td。其表示一个实体的行为优先于另一个实体的行为。时间依赖也可以称作顺序依赖,时间依赖是一种特殊的协作依赖或排斥依赖。

形式化:假设  $A$  与  $B$  为两个实体,则实体  $A$  与实体  $B$  存在时间依赖关系,当且仅当:

在时刻  $T'$ ,

$$\forall A \forall B (A \wedge \neg B) \Rightarrow A \xrightarrow{\text{td}} B$$

### (4) 因果依赖

因果依赖(cause dependency):简称为 cd。其表示在时间区间  $T$  内,一个实体的行为源于另一个实体的行为,或者说一个实体的行为是由另一个实体的行为导致的。

形式化:假设  $A$  与  $B$  为两个实体,则实体  $A$  与实体  $B$  存在因果依赖关系,当且仅当:

在时间区间  $T$  内,

$$\forall A \forall B (A \rightarrow B) \Rightarrow A \xrightarrow{\text{cd}} B$$

### (5) 继承依赖

继承依赖(inherit dependency):简称为 ind。其表示一个实体是另一个实体升级或变更后产生的新实体,两者为父子关系。

形式化:假设  $A$  与  $B$  为两个实体,且  $S_1$  与  $S_2$  分别为实体  $A$  与  $B$  的元素集合, $a_1 \dots a_n$  与  $b_1 \dots b_n$  分别为集合  $S_1$  与  $S_2$  中的元素,则实体  $A$  与实体  $B$  存在继承依赖关系,当且仅当:

$$\exists A \exists B (A \rightarrow B) \Rightarrow A \xrightarrow{\text{ind}} B \text{ 且}$$

$$\exists a_1 \dots a_n \exists b_1 \dots b_n ((a_1 \rightarrow b_1) \vee (a_2 \rightarrow b_2) \vee \dots \vee (a_n \rightarrow b_n))$$

元素集合满足下列条件:

$$S_1 \cap S_2 \neq \emptyset$$

### (6) 包含依赖

包含依赖(contain dependency):简称为 cod。其表示一个实体的组成部分之一为另一个实体。

形式化:假设  $A, A_1, A_2, \dots, A_n$  为实体,且  $S, S_1, S_2, \dots, S_n$  分别为实体  $A, A_1, A_2, \dots, A_n$  的元素集合,则实体  $A$  与实体  $A_n$  存在包含依赖关系,当且仅当:

$$\forall A \forall A_1 \dots \forall A_n (A = A_1 \wedge A_2 \wedge \dots \wedge A_n) \Rightarrow A \xrightarrow{\text{cod}} A_n$$

元素集合满足下列条件:

$$S_1 \cup S_2 \cup \dots \cup S_n \subseteq S$$

$$S \cap S_n = S_n$$

$$S \cup S_n = S$$

这 6 种依赖关系包含了常见的实体间依赖关系。

## 1.3 依赖关系的基本性质

实体间依赖关系的基本性质可以分为 3 种:自反性、对称性和可传递性<sup>[10]</sup>。

### (1) 自反性

假设实体  $A$  发生变更,导致实体  $A$  本身发生变化,则认为实体  $A$  的关系具有自反性。

依赖关系是基于不同实体之间的关系为前提,也就是说,依赖关系指代的是在两个或多个不同实体之间,一个实体的变化对于其他实体的影响,实体本身的变化不能够称作依赖关系,因此依赖关系是反自反的。

### (2) 对称性

假设  $A$  与  $B$  为两个实体,实体  $A$  依赖于实体  $B$ ,实体  $B$  也依赖于实体  $A$ ,则认为实体  $A$  与实体  $B$  之间的依赖关系具有对称性。

依赖关系的对称性证明如下:

协作依赖:若实体  $A$  协作依赖于实体  $B$ ,则满足  $A \wedge B$ ,

即“ $A$  真  $B$  真”;若实体  $B$  协作依赖于实体  $A$ ,则满足  $B \wedge A$ ,即“ $B$  真  $A$  真”。 $A \wedge B = B \wedge A$ ,因此协作依赖满足对称性。

**排斥依赖:**若实体  $A$  排斥实体  $B$ ,则满足“ $A$  真  $B$  假或  $A$  假  $B$  真”;若实体  $B$  排斥实体  $A$ ,则满足“ $A$  假  $B$  真或  $A$  真  $B$  假”。两者条件相同,因此排斥依赖满足对称性。

**时间依赖:**若实体  $A$  优先于实体  $B$ ,则实体  $A$  的执行时间先于实体  $B$ ;若实体  $B$  优先于实体  $A$ ,则实体  $B$  的执行时间应该先于实体  $A$ 。两者条件互斥,因此时间依赖不满足对称性。

**因果依赖:**若实体  $A$  导致实体  $B$ ,则需要满足条件“ $B$  真  $A$  假”;若实体  $B$  导致实体  $A$ ,则可以满足条件“ $B$  真  $A$  假”。条件冲突,因此因果依赖不满足对称性。

继承依赖关系与包含依赖关系是父子关系,不存在对称性。

### (3)可传递性

假设  $A, B, C$  为 3 个实体,实体  $A$  依赖于实体  $B$ ,实体  $B$  依赖于实体  $C$ ,则实体  $A$  依赖于实体  $C$ ,这种情况下认为实体  $A$  与实体  $B$  之间的依赖关系,以及实体  $B$  与实体  $C$  之间的依赖关系是可传递的。

依赖关系的传递性证明如下:

**协作依赖:**若实体  $A$  协作依赖于实体  $B$ ,实体  $B$  协作依赖于实体  $C$ ,则需要满足条件  $A \wedge B$  且  $B \wedge C$ ,即“ $A$  真  $B$  真且  $B$  真  $C$  真”,由该条件可得“ $A$  真则  $C$  真”,即  $A \wedge C$ ,因此协作依赖满足可传递性。

**排斥依赖:**若实体  $A$  排斥实体  $B$ ,则满足条件“ $A$  真  $B$  假或  $A$  假  $B$  真”;若实体  $B$  排斥实体  $C$ ,则满足条件“ $B$  真  $C$  假或  $B$  假  $C$  真”。若要使实体  $A$  排斥实体  $C$ ,则要满足条件“ $A$  真  $C$  假或者  $A$  假  $C$  真”,由现有条件无法推导出这一条件,因此排斥依赖不满足可传递性。

**时间依赖:**若实体  $A$  优先于实体  $B$ ,实体  $B$  优先于实体  $C$ ,则实体  $A$  优先于实体  $C$ ,因此时间依赖满足可传递性。

**因果依赖:**若实体  $A$  导致实体  $B$ ,则满足条件“ $A$  真  $B$  真”;若实体  $B$  导致实体  $C$ ,则满足“ $B$  真  $C$  真”。由上述条件可知“ $A$  真则  $C$  可能为真”,因此因果依赖满足可传递性。

**继承依赖:**若实体  $A$  继承于实体  $B$ ,实体  $B$  继承于实体  $C$ ,则实体  $C$  是实体  $A$  的初始版本,即实体  $A$  继承于实体  $C$ 。因此继承依赖满足可传递性。

**包含依赖:**若实体  $A$  包含实体  $B$ ,实体  $B$  包含实体  $C$ , $S_1, S_2, S_3$  分别为实体  $A$ 、实体  $B$ 、实体  $C$  的元素集,则满足  $S_1 \supset S_2$  且  $S_2 \supset S_3$ ,由此可得  $S_1 \supset S_3$ ,实体  $A$  包含实体  $C$ 。因此包含依赖满足可传递性。

6 种类型的实体依赖关系的性质总结如表 1 所列。

表 1 实体依赖关系的性质

依赖类型	协作依赖	排斥依赖	时间依赖	因果依赖	继承依赖	包含依赖
自反性	×	×	×	×	×	×
对称性	√	√	×	×	×	×
可传递性	√	×	√	√	√	√

## 1.4 实体依赖关系的时态性

时态实体依赖关系的时态性主要体现在两个方面:1)时态实体依赖关系本身的时态约束;2)依赖关系对相关连的两个实体的时态约束。

时态实体依赖关系本身的时态约束,顾名思义,就是依赖

关系的属性“有效时间”对其产生的约束。在对时态实体依赖关系进行分析和查询等操作时,需要充分考虑依赖关系本身存在的时态约束,只有满足时态约束的依赖关系才是有效的依赖关系,在特定的时间区间内一旦发生了变更,就会对整个时态实体依赖网络产生影响。

时态实体依赖关系本身的时态约束如图 1 所示。

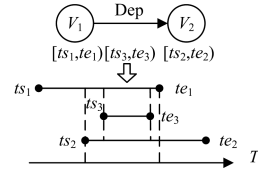


图 1 时态实体依赖关系的时态约束

依赖关系对相关连的两个实体的时态约束体现在,不同的时态实体依赖关系类型对依赖关系的依赖实体与被依赖实体在时态上的约束是不同的。

分别分析不同类型的实体依赖关系:

**协作依赖:**协作依赖要求依赖关系两端的实体同时存在,即一个实体的存在需要另一个实体的存在。基于这一定义,要求依赖实体与被依赖实体的运行时间必须重合,而有效时间也必须同时满足协作依赖的有效时间。

**排斥依赖:**排斥依赖与协作依赖相反,要求当一个实体存在时另一个实体不能存在。排斥依赖是对实体运行时间的约束,即两个实体不能在同一时间区间内运行,而排斥依赖对实体本身的有效时间没有特别的约束,只需要依赖实体与被依赖实体满足排斥依赖的有效时间即可。

**时间依赖:**时间依赖也对依赖实体与被依赖实体的运行时间进行了约束,即依赖实体在运行时间上必须优先于被依赖实体。在有效时间上,依赖实体与被依赖实体只需要满足时间依赖的有效时间即可。

**因果依赖:**因果依赖表现了实体之间的因果关系,被依赖实体导致依赖实体,即被依赖实体为因,依赖实体为果。因此因果依赖要求被依赖实体的有效时间早于依赖实体的有效时间。只有因先存在,才能够导致果的发生。

**继承依赖:**继承依赖表现的是实体之间的父子关系,依赖实体为子,被依赖实体为父。因此继承依赖要求被依赖实体的有效时间早于依赖实体的有效时间。只有被依赖的实体首先是真实的,依赖实体才有可能存在。

**包含依赖:**包含依赖表现的是实体之间部分与整体的关系,先有部分的存在,才能够有整体的存在,且部分实体的有效时间的终止时间也应该晚于整体实体的终止时间,如果部分实体变更或者失效了,整体实体也应该发生变更或者失效。因此包含依赖要求被依赖实体的有效时间的起始时间早于依赖实体,终止时间晚于依赖实体。只有被包含的部分先为真,包含的整体才可能为真。

时态实体依赖关系对相关连的两个实体的运行时间的时态约束如图 2 所示。

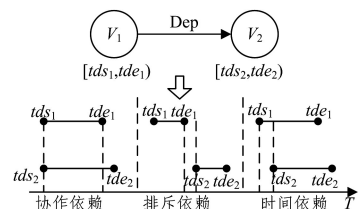


图 2 依赖关系对实体运行时间的约束

时态实体依赖关系对相关连的两个实体的有效时间的时态约束如图3所示。

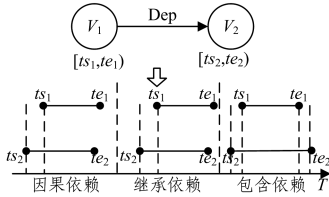


图3 依赖关系对实体有效时间的约束

## 2 时态实体依赖图

依赖关系的表示方法多种多样,依赖图是最常见的依赖关系表示方法,但是在国内外的研究文献中,依赖图的模型也是多样的,针对图的节点与边的属性度量也各有不同,本文结合国内外研究成果,提出更适应于实验要求的时态实体依赖图模型以及属性度量方法。

### 2.1 依赖图的定义

为了方便理解,人们一般用图的形式来表示各种依赖关系,并提出许多依赖图的模型,主要有程序依赖图 PDG<sup>[11]</sup>、系统依赖图 SDG<sup>[12]</sup>、对象依赖图 OPDG<sup>[13]</sup>和类依赖图 CDG<sup>[14]</sup>等。

而除了常用的一些依赖图之外,还可以通过依赖观察矩阵<sup>[15]</sup>、依赖树<sup>[16]</sup>等一些方法对实体、程序、函数等之间的依赖关系进行展示和描述。

本文使用有向依赖图来表示实体间的依赖关系,并且为传统的依赖图添加时态属性的标识。

在此,给出时态实体依赖图 TEDG 的定义如下。

**定义 2(时态实体依赖图)** 设时态实体依赖图(Temporal Entity Dependency Graph)  $TEDG = (V, E, ET, T)$ 。其中,  $V$  是节点  $v_i$  的集合,表示不同的实体;  $E$  表示实体间的依赖关系的有向边的集合,例如  $\langle v_i, v_j \rangle$ ,表示节点  $v_i$  依赖于节点  $v_j$ ;  $ET$  是节点边类型的集合,表示依赖关系的类型;  $T$  是一个时间区间的集合,用于标注节点与边的时态属性,限定节点与边的有效时间。

一个时态实体依赖图如图4所示。

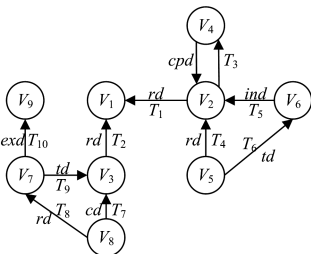


图4 时态实体依赖图示例

从这个时态实体依赖图可以看到,除了协作依赖之外,其余的依赖关系都是不对称的。

### 2.2 节点与边属性的度量

依赖图中,节点与边的重要强度和强度都有所不同,当其中的某个节点或边发生变更时,其对于整个依赖图的影响程度也不同,尤其是重要的中心节点,如果发生变化,则可能会对整个依赖图产生影响。

为了能够以最小的代价保护依赖图,以及评估因节点变

更可能会产生的影响,对于节点与边属性的度量就显得极为重要。

复杂网络中,常见的中心节点的度量方法有节点的中心度度量、接近中心度度量、介数中心度度量、特征向量中心度度量等<sup>[17]</sup>。常用的 PageRank 算法<sup>[18]</sup>就是一种特征向量中心度度量方法。

结合传统的节点中心度度量方法,以及时态实体依赖图特有的时态属性特征,本文提出了时态实体依赖图节点与边的度量公式。

#### (1) 节点的中心性

节点的中心性可以表现出节点在一个依赖关系图中,与其他节点直接相关联的程度,通常情况下,一个节点的中心性越高,代表当这个节点发生变更时,能够直接影响到节点也越多,也越有可能对其余非直接依赖的节点产生影响。

定义复杂网络中对节点中心度的度量为:

$$C_V(i) = \text{indeg}(i) + \text{outdeg}(i) \quad (1)$$

其中,  $\text{indeg}(i)$  代表节点  $i$  的入度,  $\text{outdeg}(i)$  代表节点  $i$  的出度<sup>[19]</sup>。

但是在时态实体依赖图中,需要考虑时态属性对依赖图节点数的影响,在不同时间段,时态实体依赖图的节点数量可能会发生变化,节点的入度与出度也有所不同,这都会对节点的中心性产生影响,因此对时态实体依赖图的节点的中心性定义如下:

$$C_V(i, t) = \frac{\text{indeg}(i, t) + \text{outdeg}(i, t)}{g(t) - 1} \quad (2)$$

其中,  $g(t)$  表示  $t$  时刻依赖图中所有节点的数量;  $\text{indeg}(i, t)$  表示  $t$  时刻节点  $i$  的入度,  $\text{outdeg}(i, t)$  表示  $t$  时刻节点  $i$  的出度。

如果以  $\text{deg}(i, t)$  表示  $t$  时刻节点  $i$  的度,则式(2)也可以表示为:

$$C_V(i, t) = \frac{\text{deg}(i, t)}{g(t) - 1} \quad (3)$$

#### (2) 节点的重要性

节点的重要性与中心性类似,都需要根据节点的度进行度量,但节点的重要性与节点的入度关联更强,即节点的重要性更多取决于依赖于这个节点的其余节点的数量。当依赖于某一节点的节点数量越多,当这个节点发生变化时,会有更多的节点直接受到影响。

节点的重要性被定义为直接依赖于该节点的节点中心性之和:

$$D(i, t) = \sum \text{in}C_V(j, t) \quad (4)$$

其中,  $\text{in}C_V(j, t)$  为  $t$  时刻直接依赖于节点  $i$  的节点  $j$  的中心性。

#### (3) 节点的依赖度

时态实体依赖图一个节点对另一个节点的依赖程度即为该节点的依赖度。

节点的依赖度表示,在  $t$  时刻,对于存在依赖关系的两个实体,当被依赖的实体发生了变更后,依赖于该实体的实体有多大的可能性会随之发生改变,表示为  $S(i, j, t)$ 。例如,  $A$  和  $B$  是两个实体,且实体间存在关系  $A \rightarrow B$ ,则当实体  $B$  发生变更后,实体  $A$  也一定会发生变更,则两个节点之间的依赖度为 1;反之,当实体  $B$  发生变更后,实体  $A$  不会发生变更,则两

个节点之间的依赖度为 0。

首先引入几个概念。

节点依赖贡献:在时刻  $t$ ,与节点  $i$  存在依赖关系的各节点贡献值之和即为节点的依赖贡献,表示为  $G(i, set, t)$ 。Set 为所有与  $i$  存在依赖关系的节点的集合。

计算方法如下:

各节点的初始依赖分量  $R$  设为 0.5,当节点  $h$  依赖于 3 个节点  $q, k, m$  时,节点  $h$  将它的初始贡献值的  $\frac{1}{3}$  分给节点  $q$ ,称作节点  $h$  对节点  $q$  的依赖分量,记作  $R(h, q, t)$ 。

节点依赖贡献的计算通常只考虑两层依赖关系:

(1)当没有任何节点依赖于节点  $i$  时,则节点  $i$  的依赖贡献为初始依赖分量 0.5。

(2)当存在节点依赖于节点  $i$  时,节点  $i$  的依赖贡献首先来自于直接依赖于节点  $i$  的第一层节点的依赖分量,而第一层节点的依赖分量来自于第二层节点的依赖分量。计算公式如下:

$$G(i, set, t) = \sum_{k=1}^{set} R(k, i, t) \quad (5)$$

$R(k, i, t)$  的计算公式如下:

$$R(m, k, t) = \sum_{m=1}^{set'} (R/outdeg(m)) \quad (6)$$

$$R(k, i, t) = \sum_{k=1}^{set'} (R(m, k, t)/outdeg(k)) \quad (7)$$

节点的依赖度  $S(i, j, t)$  表示加入节点  $j$  之后,节点  $i$  的依赖贡献变化与依赖贡献的比值,即:

$$S(i, j, t) = \frac{G(i, set', t) - G(i, set, t)}{G(i, set', t)} \quad (8)$$

节点依赖度能够很好地表现出节点之间依赖关系的强弱,当节点之间的节点依赖度较高时,节点之间依赖关系的强度也相对更高。

针对上述 6 种依赖关系进行具体分析:

假设两个实体  $A$  与  $B$  之间存在依赖关系,且实体  $A$  依赖于实体  $B$ ,在时刻  $t$ ,实体  $A$  对实体  $B$  的节点依赖度为  $S(B, A, t)$ 。

协作依赖: $S(B, A, t)$  的值越高,表示实体  $A$  对实体  $B$  的依赖程度越高,即实体  $A$  存在时越需要实体  $B$  的协作。

排斥依赖: $S(B, A, t)$  的值越高,表示实体  $A$  与实体  $B$  之间的排斥越严重,实体  $B$  存在时,实体  $A$  发生错误的可能性越大。

时间依赖: $S(B, A, t)$  的值越高,表示实体  $A$  与实体  $B$  之间的时间依赖程度越高,即在时刻  $t$ ,实体  $A$  执行时越要优先于实体  $B$ 。

因果依赖: $S(B, A, t)$  的值越高,表示实体  $A$  与实体  $B$  之间的因果依赖程度越高,即由实体  $B$  导致实体  $A$  发生的可能性越高,当实体  $B$  发生变更时,实体  $A$  出现相应变化的可能性也越大。

继承依赖: $S(B, A, t)$  的值越高,表示实体  $A$  中有越多的元素继承于实体  $B$ ,实体  $B$  与实体  $A$  之间的父子关系越强,则当实体  $B$  发生变更后,实体  $A$  中继承于实体  $B$  的元素受到影响从而导致实体  $A$  变更的可能性越高。

包含依赖: $S(B, A, t)$  的值越高,表示实体  $B$  在实体  $A$  所包含的所有实体中所占的比例越大,即实体  $B$  所包含的元素

在实体  $A$  所包含的所有元素中所占的比例越高,则当实体  $B$  发生变化后,对实体  $A$  的影响也越大,也更有可能导致实体  $A$  的变更。

(4)边的重要性

对于边的重要性,需要考虑的因素较多,除了节点间的依赖度之外,还需要考虑两个顶点的重要性/中心性对边的重要性的影响。结合节点的介数中心性度量方法,提出边的重要性度量公式,具体定义如下。

当节点  $v_i$  直接依赖于节点  $v_j$  时,

$$D_c(i, j, t) = S(j, i, t) * \left[ \frac{D(i, t)}{outdeg(i, t)} + \frac{D(j, t)}{indeg(j, t)} \right] \quad (9)$$

当节点  $v_i$  不直接依赖于节点  $v_j$  时,

$$D_c(i, \dots, j, t) = \frac{B(i, j, t)}{L(i, j, t)} * \sum_n^j D_c(i, n, t) \quad (10)$$

$B(i, j, t)$  为  $t$  时刻节点  $i$  与节点  $j$  之间该路径的介数中心性:

$$B(i, j, t) = \frac{L_s(i, j, t)}{L(i, j, t)} \quad (11)$$

其中,  $L_s(i, j, t)$  为  $t$  时刻节点  $i$  与节点  $j$  之间的最短路径长度<sup>[18]</sup>;  $L(i, j, t)$  为  $t$  时刻所求路径长度。当  $t$  时刻所求路径即为最短路径,则  $B(i, j, t)$  为 1。

边的重要性更能够体现一条路径对于路径上的相关节点甚至局部时态网络的影响程度大小。一条路径的重要性越高,则该路径上发生的变更对路径上的相关节点的影响程度越高。如果一条路径的重要性非常高,当这条路径发生变更后,可能会对节点产生非常大的影响,从而影响到局部时态网络的变更状况。

通常情况下,两个节点之间的最短路径的重要性最高,即两个节点之间的最短路径对两个节点以及路径上的其他节点的影响程度最高。路径上涉及到的节点越多,反而会降低对初始节点与终止节点的影响。而路径上的节点过多,当路径上的依赖关系发生变更时,对于任一节点的影响都比较有限,引起局部网络大面积变更的可能性便越小。

### 3 实验分析

本文以 Maven Repository 数据集<sup>[21]</sup>作为实验分析的基础,将 Maven Repository 数据集中文件之间的依赖关系作为分析对象。具体的实验环境如下:操作系统:Windows 7 专业版 32bit;CPU: Intel(R) Core(TM) i5-4590 3.30 GHz;内存:4.00 GB;开发语言:JAVA, Cypher;开发工具: jdk1.8.0\_121, mysql-5.7.17, MyEclipse, Neo4j CE 3.1.1。

在 Maven Repository 数据集中,文件相当于一个实体(也称为构件),文件之间存在引用、参考等多种操作,这些操作即构成了文件之间的依赖关系。文件的发布时间、失效时间以及文件之间引用的时间就是文件以及文件间依赖关系的时态性。而这些文件与文件间的依赖关系以及它们所具有的时态属性即构成一个时态实体依赖图。

本文首先从 Maven Repository 数据集中提取了 7000 个文件以及 6783 条文件间的依赖关系,并对这部分数据进行了初步的筛选。首先对文件间依赖关系的时态属性进行评估,对于不满足节点的时态属性的依赖关系进行剔除,最终得到 4411 条有效的依赖关系数据,再对节点进行筛选,对节点中

并不参与这 4411 条依赖关系的独立节点进行查询和剔除,最终得到了 5173 条有效的节点数据。对得到的有效的依赖关系数据以及节点数据进行处理,最终得到一个时态实体依赖图,部分构件依赖图如图 5 所示。

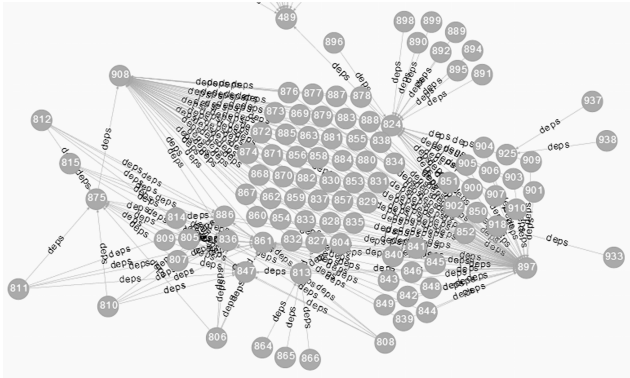


图 5 构件依赖关系图

基于这些有效数据,对上述理论进行实验验证与分析。

指定 10 个时间点  $T_1 - T_{10}$ ,再随机选取 3 个节点,在不同时间点对 3 个节点的中心性、重要性的变化情况进行分析。

节点中心性的变化情况如图 6 所示,图 6 中横坐标表示不同的时间点,纵坐标表示节点中心性度量值。

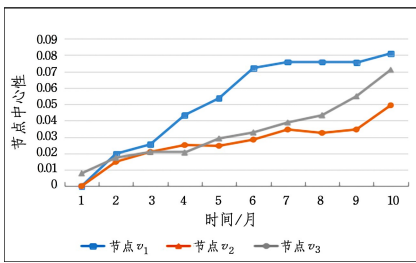


图 6 节点中心性变化

由图 6 可以看出,随着时间的推移,时态实体依赖图中的节点数有所增加,但是与节点直接相关联的节点数也有所增加,在这种情况下,在大部分时间点,节点的中心性都在小幅度上升,但是当某一节点在某一时间范围内的关联节点的增加比率低于该时间范围内整体节点的增长比率时,该节点的中心性才会略微降低。

节点重要性的变化情况如图 7 所示,图 7 中横坐标表示不同的时间点,纵坐标表示节点的重要性度量值。

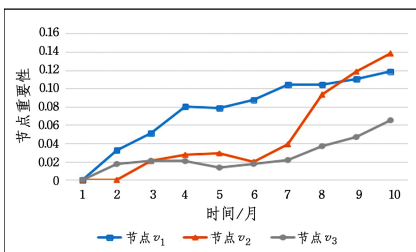


图 7 节点重要性变化

节点重要性的变化情况相较于节点中心性的变化情况更加复杂。随着时间的推移,节点总数发生了变化,与节点相关联的节点数也在发生变化,但是当某一节点的出度发生大幅变化时并不会对该节点的重要性产生巨大的影响,反而是在某一时间点,若有多个节点依赖于该节点,将大幅增加该节点的重要性。

由节点中心性变化情况与节点重要性变化情况的对比可知,节点的中心性与节点的重要性变化趋势是不一样的,随着节点入度的增大,以及直接依赖于节点的节点中心性的增大,节点的重要性变化也可能更大。

指定 10 个时间点  $T_1 - T_{10}$ ,再从数据库中随机选取 3 个节点对,对节点的依赖度随时间的变化情况进行分析。

节点的依赖度变化情况如图 8 所示,图 8 中横坐标表示时间点  $T_1 - T_{10}$ ,纵坐标表示节点依赖度的度量值。

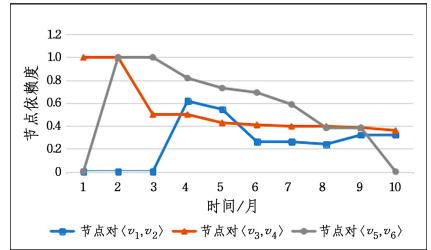


图 8 节点依赖度变化图

由图 8 中的数据可以看出,当指定的节点对之间并不存在依赖关系时,两个节点之间的依赖度为 0;当指定的节点对中,节点唯一依赖于另一个节点,则该节点对另一个节点的依赖度为 1。

一个节点对另一个节点的依赖度降低一般存在两种可能性:1)依赖于该节点的节点总数增加,使得该节点的依赖贡献总量增加,从而降低了另一节点的依赖分量对该节点的影响;2)依赖于另一个节点的二层节点数减少,从而降低了另一节点的依赖分量。

指定 10 个时间点  $T_1 - T_{10}$ ,并从时态实体依赖图中任意选取两条路径  $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4$  以及  $v_5 \rightarrow v_6 \rightarrow v_7$ ,对边的重要性随时间变化的情况进行分析。

边的重要性变化情况如图 9 所示,图 9 中横坐标表示时间点  $T_1 - T_{10}$ ,纵坐标表示边的重要性的度量值。

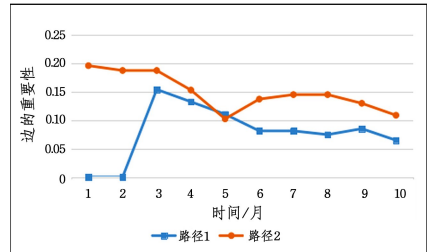


图 9 边的重要性变化图

当路径不通或者某节点不存在时,边的重要性为 0。

影响边的重要性的元素过多,对于任一固定路径,路径上节点的中心性以及重要性的变化均会对边的重要性产生影响。

**结束语** 本文通过对传统依赖图构建方法的总结和分析,结合时态属性特性,对实体的依赖关系进行了定义。又结合复杂网络中节点中心性的度量方法,提出时态实体依赖图的节点与边的中心性、重要性等属性度量方法。通过实验分析节点和边的度量随时间变化的规律。依据本文提出的时态实体依赖关系的分类和度量方法,未来可以构建有效的时态实体依赖性分析算法。

参考文献

[1] 王炜,李彤,何云,等.一种软件演化活动波及效应混合分析方法

- [J]. 计算机研究与发展, 2016, 53(3): 503-516.
- [2] STEVENS W, MYERS G, CONSTANTINE L. Structured design[J]. Systems Journal, 1974, 13(2): 115-139.
- [3] VIEIRA M, RICHARDSON D. Analyzing dependencies in large component-based systems[C]// 17th IEEE International Conference on Automated Software Engineering. ASE 2002: 241-244.
- [4] 欧阳, 胡顺仁, 蒋西明. 对象类之间依赖关系度量分析[J]. 计算机工程与应用, 2004, 40(25): 81-83.
- [5] NORMAN W, ROSS H, SCOTT H. Dependency analysis tools reusable components for software maintenance[C]// Conference on Software Maintenance. 2015: 126-131.
- [6] 易彤. 面向 UML 模型的依赖性分析及应用的研究[D]. 南昌: 江西财经大学, 2006.
- [7] LEE M, OFFUTT J, ALEXANDER R. Algorithmic Analysis of the Impacts of Changes to Object-oriented Software[C]// 34th International Conference on Technology of Object-Oriented Languages and Systems. 2000: 61-70.
- [8] ARIAS T B C, SPEK P V D, AVGERIOU P. A practice-driven systematic review of dependency analysis solutions [J]. Empirical Software Engineering, 2011, 16(5): 544-586.
- [9] WILDE N, HUITT R, HUITT S. Dependency analysis tools reusable components for software maintenance[C]// Conference on Software Maintenance. 1989: 126-131.
- [10] 胡顺仁, 陈伟民, 廖昌荣, 等. 基于 UML 类图的类之间依赖关系图论问题研究[J]. 计算机工程, 2006, 32(12): 1-2.
- [11] 郭婧, 吴军华. 基于程序依赖图的克隆检测及改进[J]. 计算机工程与设计, 2012, 33(2): 595-600.
- [12] LI B X, WANG Y F, ZHANG Y Z, et al. An Approach of Static Coarse-Grained Slice Based on Simlified System Dependence Graph[J]. 软件学报, 2001, 12(2): 204-211.
- [13] 谭毅, 朱平, 李必信, 等. 一个分层切片工具模型[J]. 软件学报, 2001, 28(6): 1810-1817.
- [14] 成小芹, 王一莉. 基于 UML 类图的软件可测试性分析方法研究[J]. 计算机工程与设计, 2011, 32(4): 1326-1329.
- [15] 卢田毅. 基于依赖观察矩阵的软件故障定位技术研究[D]. 大连: 大连海事大学, 2013.
- [16] WANG L X, YANG J, WAN R X, et al. Reducing combinatorial test suite based on tree-model by using input parameters relationships[J]. 计算机应用研究, 2010, 27(3): 928-932.
- [17] WANG R, HUANG R, QU B. Network-based analysis of software change propagation[J]. The Scientific World Journal, 2014 (2014): 237-243.
- [18] 陈康, 郑纬民. 云计算: 系统实例与研究现状[J]. 软件学报, 2010, 27(2): 429-433.
- [19] 姚静, 赵彤洲. 复杂社会网络节点重要性研究[J]. 计算机与数字工程, 2016, 44(1): 80-82.
- [20] ABBASI A, ALTMANN J, HOSSAIN L. Identifying the effects of co-authorship networks on the performance of scholars: a correlation and regression analysis of performance measures and social network analysis measures [J]. Journal of Informetrics, 2011, 5(4): 594-607.
- [21] <http://mvnrepository.com>.

(上接第 496 页)

系统开发中扮演着越来越重要的角色。

1) 作为软件的雏形, 软件体系结构是所有软件需求的一种客观表述;

2) 优秀的软件体系结构的设计可以减少和避免软件错误的产生, 保证软件产品的质量;

3) 进行软件体系结构分析可以识别体系结构设计中的潜在风险, 验证系统的质量需求是否在设计中得到了体现, 及早发现软件理解、设计中的错误, 通过及时修改软件以降低软件的开发代价及日后维护阶段的高昂代价;

4) 对于不熟悉系统的维护员来说, 软件体系结构是维护工作的切入点, 可通过软件体系结构来熟悉软件系统。

由上述分析可知, 软件体系结构既是连接需求分析和详细设计的桥梁, 也是连接需求分析人员和程序员的重要枢纽。可见, 软件 RM 测试系统体系结构已迅速成为决定软件可靠性与维护性测试系统开发成功与否的关键因素。因此, 本文将“软件可靠性与维护性测试系统体系结构设计”作为基于软件体系结构的软件 RM 测试系统开发的研究重点。

**结束语** 本文分析了两种系统思维方式: 1) 硬系统思维方式, 擅长找出问题的最优解决方案; 2) 软系统思维方式, 主要处理目标模糊的多因素复杂问题。面对“由软件失效引发故障”这一问题, 本文应用软硬系统综合方法逐步理清了问题思路, 在利用软系统方法论将问题转化为开发软件可靠性与

维护性测试系统后, 采用硬系统思维方式找出了系统开发的最优方式——基于软件体系结构的软件 RM 测试系统开发。最后, 基于软件体系结构在软件 RM 测试系统开发过程中的关键性作用, 将软件 RM 测试系统体系结构设计确定为问题的重点研究内容。

## 参 考 文 献

- [1] 宋太亮. 设备综合保障实施指南[M]. 北京: 国防工业出版社, 2006.
- [2] 赵盼. 软件维护性综合评估技术[D]. 石家庄: 军械工程学院, 2003.
- [3] 齐有忠, 高军, 黄勇. 通用武器装备供应系统设计中的硬、软系统方法整合研究[J]. 军械工程学院学报, 2003, 15(4): 42-47.
- [4] 龚强, 胡运权. 论软系统方法与城市综合功能 GIS 建设[J]. 学术交流, 2002(5): 110-114.
- [5] 中国国防科学技术报告: 嵌入式软件可靠性仿真测试与验证技术的研究[R]. 北航可靠工程研究所, 2000.
- [6] 叶飞. 软件供应保障技术与方法研究[D]. 石家庄: 军械工程学院, 2004.
- [7] 孙凌宇, 刘国买. 项目风险分析中的软系统方法[J]. 职业圈: 现代软科学, 2006(2): 82-84.
- [8] 孙昌爱, 金茂忠, 刘超. 软件体系结构研究综述[J]. 软件学报, 2002, 13(7): 1228-1237.
- [9] 李坤. 面向领域的软件体系结构复用与演化[D]. 武汉: 华中科技大学, 2007.