一种利用UML的软件需求阶段风险评估方法

刘锦行 夏红霞
(武汉理工大学计算机科学与技术学院 武汉 430070)

摘 要 在软件开发活动早期阶段评估软件的风险及其影响程度将有效减少软件开发成本和降低软件开发风险。针对目前软件风险评估的研究主要集中在软件过程的中后期阶段的现状及遵循“尽早识别和控制风险”的实践准则,提出了一种在软件需求分析阶段,利用UML建模图形度量软件风险的方法。该方法主要关注在软件需求分析阶段预防软件风险,为降低风险在软件开发后期产生严重影响提供优化参考。

关键词 软件风险评估,软件需求分析,软件工程,软件质量控制
中图法分类号 TP311.53 **文献标识码** A

New Methods of Software Requirements Risk Assessment Using UML

LIU Jin-hang XIA Hong-xia
(School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China)

Abstract Evaluating software risks in the early stages of software development activities will effectively reduce the level of software development costs and the development risk. Aiming at the status that the current research on the software risk assessment mainly focuses on late stage of the software development, and following the practice principle which is “early identification and control software risk”, this paper presented a method for assessment software risk at requirements analysis phase using UML. This method primarily focuses on the software risk prevention at requirements analysis phase, and sequentially gives some reference to reduce the risk impact of the late phase.

Keywords Software risk assessment, Software requirements analysis, Software engineering, Software quality control

1 引言

软件危机的产生使软件开发人员和软件工程研究人员深刻认识到软件过程及其控制对软件质量的重要性。在软件开发过程中,软件的风险在每个阶段都可能产生。然而,研究表明50%以上的风险发生在软件需求阶段[1]。在软件开发中,越早控制软件风险,软件开发的成本越低[2]。因此,在软件需求分析阶段定量测量软件的风险显得非常必要。同时,在实践软件质量控制过程中,软件质量保证人员需要能够更好地度量软件需求分析的质量来判断软件需求分析与标准要求之间的差距,从而保证需求分析规格说明符合质量要求的标准。

目前对于软件的风险分析研究包括可靠性风险、保密性风险及基于安全的风险等方面的研究。而目前大部分软件风险评估的方法都集中在软件开发的后期阶段[3],例如最典型的就是在设计阶段和编码阶段。显然,这些方法虽然能够识别风险,但在预防风险出现方面有着很多的限制。

在软件需求阶段分析软件的风险,可以识别软件功能的风险高低及影响的大小。由于在需求阶段接受软件的变更更为灵活,因此在需求阶段分析和识别软件的风险就显得更为重要[18]。在软件实际开发中,需求阶段风险评估的通用方法是采用领域专家对需求描述的评价。这种风险评估方法的本身就是主观和非正式的评估。当然,在很多软件项目中,尤其是一些小型的软件开发团体里是不可能有这样的领域专家组成专家组进行软件需求的风险评估的。

鉴于以上的需求及现实问题,本文提出了一个不依赖领域专家的风险评估方法,该方法在软件开发阶段早期的需求分析阶段进行,能够为软件开发尽早识别风险并且不依赖或尽量少地依赖领域专家来识别风险。当然,本文提出的方法不能完全解决需求阶段的风险评估问题,在条件允许的情况下,领域专家的参与将会提高风险识别的概率。

软件风险由两个因素来定义[3]:1)缺陷率,2)缺陷引起的后果的严重等级。要度量软件在需求阶段的风险,就要计算软件组件的缺陷概率及这些缺陷引起的后果严重程度。根据这两者的计算即可估计软件的风险。目前,在软件需求阶段评估软件风险的研究,可查的文献非常有限,比较有影响的是Appukkutty[4]等人提出的一个使用UML评估软件需求风险的方法。该方法根据组件的环形复杂度计算软件组件的缺陷率,采用功能失效分析对风险进行分级来表示缺陷引起的后果验证程度。本文提出的方法与之不同的地方在于对风险影响严重等级的测量上。在软件风险影响严重等级的测量上,本文引入了软件网络的概念。

本文第2节介绍国内外研究的相关工作;第3节介绍本

到稿日期:2013-08-06 返修日期:2013-11-11 本文受国家重大基础研究(973)项目:需求工程——对复杂系统的软件工程的基础研究(2007CB310800)资助。

刘锦行(1986—),男,硕士,主要研究方向为软件工程、计算机应用技术;夏红霞(1960—),博士,教授,主要研究方向为软件工程。

文提出的需求阶段风险评估方法；第 4 节通过一个例子来介绍本方法的使用，并验证方法的有效性；最后对本文进行总结。

2 相关工作

按照规范的软件开发过程，软件质量控制活动在软件开发的每个阶段都要进行。尤其是现代软件系统趋向于复杂化和大规模化[5-7]，如果不控制好开发过程的每个阶段的活动，将给软件维护带来灾难性的后果。在引言中本文已经说明，在软件开发的生命周期的各阶段出现了很多风险度量方法[3]，但在软件需求阶段的风险度量上基本是采用领域专家的主观评估。正如我们所知，在需求阶段将风险识别出来会大大减少软件维护和开发的成本。

近几年，出现了一些软件风险评估方面的研究。2001 年，Gilliam[8] 等人[12] 提出了一个降低软件安全性风险的方法。该方法是通过识别软件脆弱点及执行基于属性的测试来识别安全性方面的风险。2002 年，Yacoub[9] 等人提出了一个利用 UML 模型在软件结构层评估软件可靠性风险的方法。2003 年，Hassan[3] 等人提出利用 UML 模型在软件结构层评估软件结构的风险。2011 年，Jian Xu[12] 等人也提出了一个利用 UML 模型在结构层评估软件可靠性风险的方法。

从以上可以看出，目前对软件风险评估的形式化研究更多是集中在软件的结构层，即在软件的结构设计阶段。显然，当我们的需求出现问题并没有被发现的时候，设计阶段是很难发现这种需求风险的，因为设计阶段是在需求阶段基线建立后开始的工作。不管是从软件开发、维护成本方面的要求还是软件质量控制方面的要求来讲，需求阶段的风险评估方法研究都是很有价值的。这也是本文研究的动机之一。

对于需求阶段的领域专家评估方式，正如我们前面所述，这样的条件在很多开发实践中是不具备的；而且，领域专家评估方式在需求阶段的风险评估中多是非形式化的，这也给评估描述带来了模糊和不准确的潜在可能。UML 作为一个目前通用的，起到建模语言标准作用的一个建模语言，其形式化描述的功能非常丰富，也足够起到准确沟通的作用。对软件需求阶段进行形式风险评估是本文研究的另一个动机。第 3 节给出本文的需求阶段风险评估方法。

综上所述，本文试图解决以下几个问题：

(1)在软件开发生命周期的早期阶段识别软件的风险，以期降低软件风险发生的概率；

(2)利用 UML 的形式化描述功能，以期改变软件需求阶段的非形式化描述的现状，减少软件需求阶段风险评估对领域专家的依赖。

3 需求阶段的风险评估方法

3.1 软件需求的 UML 分析模型

软件需求分析阶段，在捕获了需求之后需要对非形式化描述的领域需求进行形式化建模。形式化建模的目的是准确地描述对象。软件需求分析阶段关注的重点是对领域功能的准确描述。本文提出的需求风险评估方法就是依据需求的 UML 模型对软件功能的形式化描述。UML 的形式化描述是根据领域场景构建一系列的活动图。活动图由一系列的序列图、协作图建模进行细化描述。UML 的协作图描述了类之间的协作关系，即类之间的关联关系，协作图可以和序列图相互转换，它们从不同的视角描述同样的系统行为[10]。协作图反映了系统的类之间的耦合。状态图描述功能状态的转换，反映了一个场景的复杂性。本文提出的在需求阶段评估软件风险就是根据 UML 描述的系统功能的复杂性及系统功能之间耦合关系的强度来评估软件的风险。其中耦合关系反映了系统中模块对风险(软件缺陷)的累积及传播能力。下面介绍本文提出的风险评估方法。

3.2 基于 UML 的需求风险评估方法

本文提出的风险评估方法由两个度量指标构成，一是系统功能出现风险的概率，另一个则是风险影响(严重)程度。系统功能风险出现的概率与功能的复杂性有关，复杂性越高功能出现缺陷的概率越高，因此用功能的复杂性度量风险出现的概率是合适的[11]。对于第二个指标即风险影响(严重)程度，不同的研究方法有不同的定义，如文献[3,4,9,12,13]中采用软件失效模式，根据领域专家对失效模式的风险严重程度评级。软件失效模式有着标准的定义[14]，根据定义，要在需求阶段判断软件的失效模式是非常困难的。因此在软件需求阶段采用软件失效模式来定义软件风险影响(严重)程度是不合适的。鉴于此，本文采用功能缺陷传播的概率来定义软件功能风险的影响程度。本文采用构建软件网络的方式计算缺陷传播的概率，该方法由 WANG 等人在文献[15]中提出。

由于系统功能的复杂性在需求阶段不能像在设计阶段或编码阶段有更详细的细节来测量，因此要在系统需求阶段测量每个功能的复杂性就需要借助需求阶段的 UML 分析建模。包括风险影响程度的测量，本文采用需求阶段后期的分析建模的产物——协作图、序列图及类图来组成我们的测量对象，以在需求阶段提出定量的系统风险影响度量方法。

本文提出的需求阶段风险评估方法的步骤如下：

Step 1　获取需求后利用 UML 建立需求分析模型，包括协作图、序列图和分析类图及状态图；

Step 2　根据类图、序列图和协作图建立系统的加权软件网络；

根据建立的系统软件网络进行如下计算：

Step 3　计算每个软件网络结点(软件功能单元)的环形复杂度；确定每个系统功能的风险测量值，得到 $Complexity$；

Step 4　计算每个软件网络结点(软件功能单元)的传播影响值；确定每个功能的风险影响测量值，得到 $Severity$；

Step 5　根据上述第 3 步和第 4 步确定的功能风险值和风险影响值计算系统的风险度量值：$R_s = Complexity \times Severity$。

下面对本文提出的方法分节进行详细介绍。

3.3 功能组件的风险影响分析

对于软件复杂性度量，现有的研究方法很多，有些度量指标并没有很明显的软件复杂性相关性[16]。在文献[16]中 Fenton 等人通过实验比较了一些不同的环形复杂度的度量方法，其结果肯定了 McCabe[17] 于 1976 年提出的环形复杂度的度量方法的有效性。因此，本文采用传统的环形复杂度的度量方法来测量软件功能单元的复杂性。

环形复杂度广泛应用于软件模块的复杂性度量，使用模块代码的控制流图作为计算根据。定义为：

$$VG=e-n+2 \quad (1)$$

其中，e 为控制流图中边的数量，n 为图中结点的数量。

本文因为是在软件需求阶段评估软件功能单元的复杂性，故无法利用代码的控制流图来计算其复杂性。但在需求分析建模时产生的 UML 状态图描述了功能中执行状态的转换。因此，本文利用分析建模中功能单元的状态图计算其环形复杂度，该方法在文献[3]中也有使用，说明如下：

在每一个场景 S_x 中，用一个子集 C_i^x 存储功能单元 i 的被访问过的所有状态，一个子集 T_i^x 存储所有被遍历过的转换。相应地，用 $c_i^x=|C_i^x|$ 代表结点数量，$t_i^x=|T_i^x|$ 代表边的数量。则功能单元 i 的复杂性 doc_i^x 计算如下：

$$doc_i^x=t_i^x-c_i^x+2 \quad (2)$$

3.4 风险等级分析

风险等级有不同的定义，最常用的一种是根据软件组件影响的程度来定义[3,4,12,13]。文献[4]提出了一个软件需求阶段功能组件影响的测量方法。该方法称为 FFA(Functional Failure Analysis)，需要事先由领域专家对不同的失效模式进行评级。文献[4]提出的方法有其好处，即加入了领域专家的意见。但如果我们简单地使用领域专家的评级表给不同的软件功能进行风险评级显然过于僵化。本文提出的风险评估方法更专注于需求分析所得的功能对软件系统本身的影响，鉴于此，本文提出建立软件网络来计算功能单元对系统的影响。当然，如果进一步优化，可以将领域专家的评估加入需求阶段风险评估当中以提供更大的参考意义，这也是我们后阶段的工作之一。

本文采用的风险影响计算方法是根据 WANG 等人在的文献[15]中提出的 WSNNI(Weighted Software Network for Node Impact)概念的基础上设计而来。其中用到了软件网络的概念，定义如下[15]：

定义 1(软件网络) 一般用二元组来定义：$\Omega_s=(W_s,E_s)$，其中，$W_s=\{s_i\}(i=1,2,\cdots,N)$ 是 N 个类的集合，E_s 是软件模块(类、组件等)之间关系的集合。

软件网络可以由软件代码抽取结构或根据设计阶段的设计结构获得。在软件需求阶段，利用需求分析建模产生的协作图也可以构建软件网络[6]。下面介绍本文研究使用的软件网络模型构建方法。

3.4.1 需求阶段软件网络构建

利用协作图及类图可以在需求阶段构建系统的软件网络。下面以一个实例来说明软件网络在需求阶段的构建。图 1 所示是一个 POS 机系统的初始类图，它只是表明系统功能相互之间的一个依赖关系。

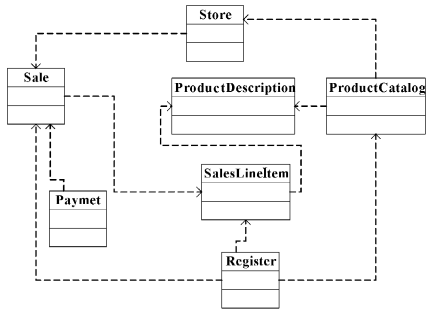

图 1 POS 机系统的初始类图

根据系统分析所用的这类建模图，可以建立软件网络，图 2 是根据上面类图构建的软件网络。从图 2 所示的软件网络，我们根据箭头的指向，可粗略地看出一些结点在系统中的影响比其他结点大，如 sale 和 SalesLineItem。但根据这个简单的软件网络我们不能准确地判断各功能单元在系统中的影响。结合需求分析建模的分析类图和序列图我们可构建一个带有权值的软件网络。需求分析建模是，分析类图列出每个类的协作类，每个职责对应一个协作类。这样我们能从这种对应关系中找出每个类相对其邻接点的关联数，这可看成是文献[15]中提出的加权软件网络的方法连接数，以此可以计算出每条边的权值。

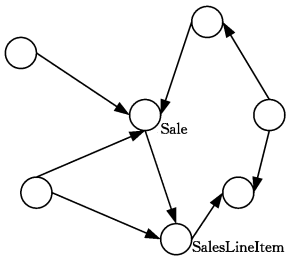

图 2 软件网络示意

3.4.2 风险等级计算方法

本文提出的风险等级的测量是采用功能单元在系统中的影响程度计算得来。在介绍风险等级计算方法之前先介绍本文改进的文献[15]提出的加权软件网络。限于篇幅，本文不介绍文献[15]提出的方法。下面介绍本文提出的需求阶段加权软件网络——功能结点影响加权软件网络 WSNFNI (Weighted Software Network for Function Node Impact)的构建。

定义 2(功能结点影响加权软件网络) 由一个二元组定义，其中包含一个软件网络的加权有向图，一个存储结点影响测度值的矩阵，表示如下：

$$WSNFNI=(G,MI) \quad (3)$$

其中，$G=(N,W)$，包括 N 个结点，以及一组带有权重的边 W。结点代表由分析建模细化而来的系统子功能。权重 W 描述结点对其邻接点的影响程度。矩阵 M_I 存储每个结点对其它结点的影响值。权值的定义如下。

WSNFNI 的权值 W 由分析类建模的 CRC(Class-Responsibility-Collaborator)中每个功能类的职责和对应的协作类计算得来。举例说明如下：

设有如下 A、B、C 3 个 CRC 卡，如表 1－表 3 所列。

表 1 类 A 的 CRC 卡

Class:A	
职责	协作类
A1	B
A2	B
A3	C
A4	无

表 2 类 B 的 CRC 卡

Class:B	
职责	协作类
B1	A
B2	C
B3	C
B4	A

表 3　类 C 的 CRC 卡

Class:C	
职责	协作类
C1	无
C2	无
C3	无

类 A、B、C 是系统的 3 个子功能。A、B、C 之间显然存在着关联关系。它们之间相互影响的程度即为它们之间关联的权值，计算如下：根据协作类的描述，类 A 与类 B 和 C 分别关联 2 次和 1 次，则 A 关联 B 和 C 的权值分别为 2 和 1，B 和 C 与其它类之间关联的权值依次计算。根据定义的权值，我们可以构建上述 3 个类组成的系统的加权软件网络（WSNFNI），如图 3 所示。

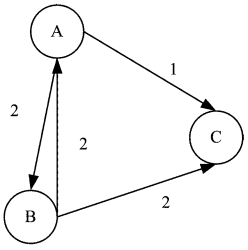

图 3　加权软件网络示意图

从图 3 中可以看出类 A、B、C 对系统的影响不同。其中，C 被 A 和 B 依赖共计 3 次，A 和 B 分别只被对方依赖 2 次。显然 C 对系统的影响是最大的。如果出现风险，则 C 对系统的危害将更为严重。

根据以上分析，我们给出如下风险等级的计算方法。

设系统 S 各权值之和为 W_s，各功能类 i 被依赖的权值之和为 W_i，则各功能类 i 的风险等级 S_i 计算如下：

$$S_i=\frac{W_i}{W_s} \tag{4}$$

为便于计算，我们用上面提到的结点影响矩阵 M_I 存储加权软件网络。则图 3 所示的加权软件网络的 M_I 如下：$\begin{bmatrix}4&2&1\\2&4&2\\0&0&3\end{bmatrix}$。矩阵中行和列分别按 A、B、C 的顺序排列，交叉处是行号到列号的权值，主对角线上的值为每个功能类的职责数，因每个结点对自己是完全依赖的。根据矩阵 M_I 我们很容易计算 W_s 和 W_i。

3.5　软件系统风险评估

在得到了各功能单元的复杂性测量值和风险等级值后我们即可计算系统的风险度量值。设功能单元 i 的复杂性测量值和风险等级测量值分别为 C_i 和 R_i，根据 3.2 节给出的计算方法，功能单元 i 的风险度量值为：

$$R_i=C_i\times S_i \tag{5}$$

系统的风险度量值 R_s 是各功能单元风险度量值之和：

$$R_s=\sum_{i=1}^{n}C_i\times S_i \tag{6}$$

其中，n 为系统需求分析所得的功能数。

下面给出一个本方法的使用实例。

4　一个实例

为清晰说明本方法的使用，我们选择了简化的 POS 机系统作为我们的实例对象。

限于篇幅，本文只选择了简化 POS 机系统中“处理销售”的复杂性计算。其状态图如图 4 所示。

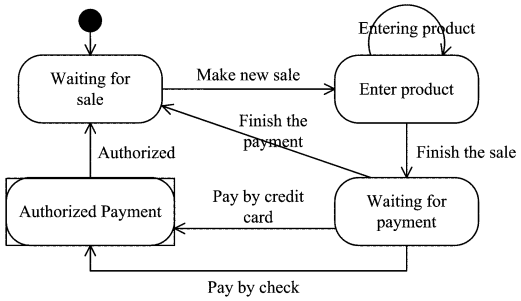

图 4　“处理销售”状态图

根据第 3 节介绍，“处理销售”功能的复杂性 C_i 计算如下：

$C_i=e-n+2=7-4+2=5$

图 1 给出了简化 POS 机系统的初始类图，其中有两个非系统开发的功能类，故忽略之。则简化的 POS 机系统我们只取 4 个功能单元：处理销售类、商品类、商品描述类、商品列表类作为本文的实例。限于篇幅，本文省略初始类图及系统的各 CRC 卡的详细描述，给出其加权软件网络图（如图 5 所示）及其结点影响矩阵。其中，A 代表销售类，B 代表商品类，C 代表商品描述类，D 代表商品列表类。

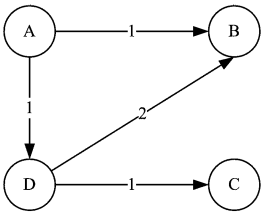

图 5　简化 POS 机系统加权软件网络图

相应的影响结点矩阵为：$\begin{bmatrix}4&1&0&1\\0&1&0&0\\0&0&2&0\\0&2&1&3\end{bmatrix}$。

则计算“处理销售类”的风险测量值为：$R_i=C_i\times S_i=5\times(4+1+0+1)/15=2$。需要说明的是，风险度量值是一个相对的数值，其风险的高低与开发计划的标准和具体的系统有关。这个值是在系统分析时给出一个风险认识的量化参考，以便于设计人员认识与改进系统的风险。

限于篇幅，本文省略了其它功能的风险测量值的计算，整个系统的风险度量值是各子系统风险值之和，因此在此也不再给出。

结束语　软件风险评估是软件开发生命周期中每个阶段都必须进行的一个活动。根据软件工程原理，越早发现软件的风险越能节约软件开发的成本，对软件开发进度也是非常有帮助的。鉴于目前大多数的风险度量研究集中在软件开发中后期，本文提出了一个在软件需求阶段进行风险评估的方法。该方法关注于软件风险的预防，为后续的软件设计提供良好的基础。

参 考 文 献

[1] Pressman R S. Software Engineering_A Practitioner's Approach [M]. 北京：机械工业出版社，2007

[2] 朱少民. 软件质量保证和管理[M]. 北京：清华大学出版社，2007

[3] Goseva-Popstojanova K, Hassan A, Guedem A, et al. Architectural-Level Risk Analysis Using UML[J]. IEEE Transactions on Software Engineering, 2003, 29(10): 946-959

[4] Appukkutty K, Ammar H H, Popstajanova K G. Software requirement risk assessment using UML[C]//3rd ACS/IEEE International Conference on Computer Systems and Applications, 2005. Cairo, Egypt, 2005:615-618
[5] 闫栋,祁国宁. 大规模软件系统的无标度特性与演化模型[J]. 物理学报,2006,55(8):3799-3806
[6] 何克清,李兵,马于涛. 软件网络[M]. 北京:科学出版社,2008
[7] 李兵,马于涛,刘婧,等. 软件系统的复杂网络研究[J]. 力学进展,2006,38(6):805-813
[8] Gilliam D P, Powell J D, Kelly J C. Reducing Software Security Risk through an Integrated Approach[C]//Presented at the The 26th Annual NASA Goddard Software Engineering Workshop. 2001
[9] Yacoub S M, Ammar H H. A methodology for architecture-level reliability risk analysis[J]. IEEE Transactions on Software Engineering, 2002, 28(6):529-547
[10] 郑明辉,周慧华,马光致. 基于 UML 需求分析模型的软件规模评估方法[J]. 计算机应用与软件,2004,21(3):23-25
[11] Munson J, Khoshgoftaar T. Software Metrics for Reliability Assessment[M]. McGraw-Hill, USA, 1996
[12] Jian X, Han Y, Qianmu L. A methodology for software reliability risk assessment[J]. Journal of Convergence Information Technology, 2011, 6(4):188-200
[13] Yacoub S M, Ammar H H, Robinson T. Methodology for architectural-level risk assessment using dynamic metrics[C]//11th International Symposium on Software Reliability Engineering (ISSRE 2000). San Jose, CA, USA, 2000:210-221
[14] 刘正高. 软件失效模式、影响及危害性分析问题探讨[J]. 电子产品可靠性与环境试验,2000,1:26-29
[15] Wang B, Wang L. Analysis of defects propagation in software system based on weighted software networks[J]. Journal of Convergence Information Technology, 2012, 7(17):63-77
[16] Fenton N E, Ohlsson N. Quantitative Analysis of Faults and Failures in a Complex Software System[J]. Proceeding(s) of the IEEE Transactions on Software Engineering, 2000, 26(8):797-813
[17] McCabe T. A Complexity Metrics [J]. IEEE Transactions on Software Engineering, 1976, 2(4):308-320
[18] 刘海,郝克刚. 软件缺陷原因分析方法[J]. 计算机科学,2009,36(1):242-244

(上接第 124 页)

结束语 本文提出基于推荐质量的信任感知推荐系统,用以综合评价相似度、领域信任度、亲密程度和领域相关度,求用户的推荐质量,并以此为基准选择推荐用户完成推荐。仿真表明本文方法能提高推荐系统在数据稀疏状况下的精确度以及冷启动用户的召回率,但由于仍然存在信任传递距离的限制,在解决评分覆盖率方面效果不是特别突出,同时也存在着时间和空间复杂度较高的问题。在未来的工作中可以应用聚类算法先将服务使用偏好相似的用户聚类,缩小搜索的范围,以便能够高效地搜索推荐质量高的推荐用户,有效降低方法的时间复杂度,同时摆脱信任传递距离的限制。

参 考 文 献

[1] Zhang L-J, Zhang J, Cai H. Services computing[M]. Springer and Tsinghua University Press, 2007
[2] Kang Guo-sheng. AWSR: Active Web Service Recommendation Based on Usage History[C]//19th International Conference on Web Services(ICWS). 2012:186-193
[3] Sarwar B, Karypis G, Konstan J, et al. Analysis of recommendation algorithms for e-commerce[C]//Presented at the Proceedings of the 2nd ACM Conference on Electronic Commerce. Minneapolis, Minnesota, United States, 2000
[4] Sinha R, Swearingen K. Comparing recommendations made by online systems and friends[C]//Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries. Puerto Vallarta, Mexico, 2001
[5] Crandall D, Cosley D, Huttenlocher D, et al. Feedback effects between similarity and social influence in online communities [C]//KDD 2008. 2008:160-168
[6] Massa P, Bhattacharjee B. Using trust in recommender systems: An experimental analysis[C]//Proceedings of iTrust2004 International Conference. 2004:221-235
[7] Chen Xiao-cheng, Liu Run-jia, Chang Hui-you. Research of collaborative filtering recommendation algorithm based on trust propagation model [C]//Computer Application and System Modeling (ICCASM). Taiyuan, 2010
[8] Jamali M, Ester M. TrustWalker: A Random Walk Model for Combining Trust-based and Item-based Recommendation[C]//KDD 2009. Paris, France, 2009
[9] Jamali M, Ester M. Using a Trust Network to Improve Top-N Recommendation [C]//Proceedings of the third ACM Conference on Recommender Systems. 2009:181-188
[10] Zarghami A, Fazeli S, Dokoohaki N, et al. Social Trust-aware Recommendation System: A T-Index Approach [C]//Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology. 2009:85-90
[11] Gao Yuan-liang, Xu Bo-yi, Cai Hong-ming. Information Recommendation Method Research Based on Trust Network and Collaborative Filtering[C]//IEEE 8th International Conference on e-Business Engineering(ICEBE). 2011:386-391
[12] Bedi P, Sharma R. Trust based recommender system using ant colony for trust computation[J]. Expert Systems with Applications, 2012, 39(1):1183-1190
[13] 张宇,陈华钧,姜晓红,等. 电子商务系统信任管理研究综述[J]. 电子学报,2008,36(10):2011-2020
[14] 蔡浩,贾宇波,黄成伟. 结合用户信任模型的协同过滤推荐方法研究[J]. 计算机工程与应用,2010,46(35):148-151
[15] Abdul-Rahman A, Hailes S. Supporting trust in virtual communities[C]//Proceedings of the 33rd Hawaii International Conference on System Sciences. USA, 2000
[16] Zhang Yu, Chen Hua-jun, Wu Zhao-hui, et al. A Reputation-Chain Trust Model for the Semantic Web[C]//IEEE 20th International Conference on Advanced Information Networking and Applications. 2006:719-723
[17] Moghaddam S, Jamali M, Ester M, et al. FeedbackTrust: Using Feedback Effects in Trust-based Recommendation Systems[C]//Proceedings of the Third ACM Conference on Recommender Systems. 2009:269-272
[18] 朱锐,王怀民,冯大为. 基于偏好推荐的可信服务选择[J]. 软件学报,2011,22(5):852-864
[19] Miller R, Perlman D, Brehm S. Intimate Relationships(4th edition)[M]. McGraw-Hill College, 2007
[20] Yulmetycv R M, Emelyanova N A, Cafarov F M. Dynamical Shannon entropy and information Tsallis entropy in complex systems[J]. Physica A, 2004, 341(11):649-676