

基于对象着色 Petri 网的 UML 模型研究

王红英,张桂戌

(华东师范大学 计算机科学与技术系,上海 200062)

摘要:UML 广泛应用于软件建模,但缺乏有效的模型检测的方法,使用形式化方法对 UML 模型进行分析,可以发现 UML 模型的设计问题,提高 UML 模型的质量。对象着色 Petri 网是一种拥有接口库所模块化着色 Petri 网,既是一种图形化建模工具,又是具有严格的语法语义定义的形式化方法。通过引入事件托肯,改进了将 UML 模型转换为对象着色 Petri 网的方法,结合实例将 UML 状态图和协作图映射为对象着色 Petri 网模型。并用着色 Petri 网的方法和工具对模型进行了分析,验证了模型的一系列性质。

关键词:UML;对象着色 Petri 网;软件建模

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2007)04-0182-04

Research on UML Model Based on Object Colored Petri Net

WANG Hong-ying, ZHANG Gui-xu

(Department of Computer Science and Technology, East China Normal University, Shanghai 200062, China)

Abstract:UML is widely used in software modeling which has a problem on model checking. By using formal method to analyze and check UML models, the design problems can be found in early phase so as to improve the quality of UML models. To solve the problems of model checking of UML models, introduces a high-level Petri net called object colored Petri net which is not only a graphical modeling tool but also a kind of formal method. And by invoking event token, a method of transforming UML model to object colored Petri net is improved when mapping UML statechart diagrams and collaboration diagrams to object colored Petri net. Then analysis and verification of the model are illustrated to verify a series of properties.

Key words:UML;object colored Petri net;software modeling

0 引言

UML 的出现得到了工业界的广泛支持,成为当今流行的规模开发工具。UML 作为一种定义良好、易于表达、功能强大,且普遍适用的建模语言为复杂信息系统的设计和分析提供了有效的手段。但是 UML 作为一种半形式化的建模语言在建造模型时,不能提供模型检验,不能用于保证系统的质量。因此,可引入形式化方法和工具对 UML 模型进行检验^[1]。Petri 网^[2]既是一种图形化建模工具,又是具有严格的语法语义定义的形式化方法,具备强大的数学理论的支持,不但能有效地对系统进行描述和建模,对系统的并发性、异步性和不确定性也有很强的动态分析能力。一种将面向

对象技术和 Petri 网结合起来的对象着色 Petri 网既保持了一般 Petri 网的性质,又简化了一般 Petri 网的规模,并提供了良好的模块性和封装性,因此,将 UML 模型转换为对象着色 Petri 网模型,有利于使用成熟的 Petri 网技术对模型进行分析和验证,提高系统的正确性和可靠性。

1 UML 模型和 Petri 网模型

1.1 UML

UML^[3]是一种可视化建模语言,它用模型来描述系统的结构或静态特征,以及行为或动态特征,主要分为结构分类、动态行为和模型管理三方面。结构分类描述系统和其他事物的关系,包括类、用例、组件和节点。动态行为描述系统的行为。模型管理描述模型的组织层次结构。

UML 用一组视图反映系统的各个方面,每个视图代表系统表述中的一个抽象,显示系统中一个特定的方面。结构分类视图包括静态视图、用例视图和执行视图。动态行为视图包括状态视图、活动视图和交互

收稿日期:2006-07-01

基金项目:国家重点基础研究(973)计划(2005CB321904);上海市教育委员会局管基金重点项目(04JG05063)

作者简介:王红英(1982-),女,四川蓬溪人,硕士研究生,研究方向为 Petri 网、模型验证;张桂戌,副教授,研究方向为软件设计方法学、Petri 网理论与应用。

视图。模型管理视图包括其他视图的管理和组织。

每个视图由一组图构成,图中包含了强调系统中某一方面的信息。UML 用类图和对象图来描述系统的静态结构;用状态图、活动图、顺序图、协作图来描述系统的动态结构。

类图用来表示系统中类与类之间的关系,对象图是类图的变体。状态图描述了一个单一对象在它生命期内的行为模型。活动图反映一个连续的活动流。顺序图用来反映随着时间的变化,对象之间交互的动态协作关系。协作图主要描述发送和接收消息的结构组织的交互。顺序图和协作图是相同底层模型的不同视图,它们是同构的,不同点在于顺序图着重于体现对象之间的消息传递的时序,协作图更关注对象交互的结构方面。

1.2 对象着色 Petri 网

对象着色 Petri 网(OCPN)是一种面向对象的高级 Petri 网,它将对象的状态和状态转换用着色 Petri 网描述,形成的着色 Petri 网称为对象子网,对象子网之间按照消息的传递通过接口库所相连。OCPN 不仅保持了普通 Petri 网的对并发、异步的处理能力,还具有良好的封装性和模块性。

定义 1 一个对象子网是一个五元组:

$$O_i = (P_i, T_i, F_i, C_i, M_{oi})$$

其中: $P_i = OP_i \cup EP_i \cup MP_i$ 为有限库所集; OP_i, EP_i, MP_i 分别为原始库所集、事件库所集和接口库所集, $MP_i = OMP_i \cup IMP_i$, 分别代表发出消息的输出库所集和接受消息的输入库所集; T_i 为有限变迁集; $F_i: (P_i \times T_i) \cup (T_i \times P_i)$ 为 O_i 自身的状态流关系; C_i 为 O_i 的颜色集; $M_{oi}: P_i \rightarrow \text{bag}(C_i)$ 为初始标识集。

定义 2 对象着色 Petri 网模型由对象子网和它们之间的关系构成:

$$M = (O, R)$$

其中: O 为模型中的对象子网 $O_i (i = 1, 2, 3, \dots, n)$ 的集合; R 为不同对象子网之间的连接关系 $R_{ij} (i, j = 1, 2, 3, \dots, m, i \neq j)$ 的集合。

R_{ij} 表示不同对象子网之间的连接关系,表示为四元组:

$$R_{ij} = (OMP_i, LP, C, IMP_j), i \neq j$$

其中, LP 为连接对象子网的连接库所, C 为 LP 关联的颜色集。 OMP_i 和 IMP_j 分别代表对象子网的输出、输入库所集。

2 UML 模型到 OCPN 的转化

UML 模型中,UML 状态图描述了对象在事件驱

动下状态转移的流程,反映了一个对象的生命期内的行为,因此,可从状态图中获取信息,构造对象着色 Petri 网的对象子网^[4]。得到对象子网后,将对象子网按照一定的原则连接起来,形成整个系统的 OCPN 模型。UML 协作图描述了对象之间的交互关系,对象间消息的交互可以表现为对象子网间发送或接收外部消息的行为。因此,可从协作图中获取消息的连接关系。

对象子网中,为表示事件的传递定义了事件托肯(event, object), event 为对象之间传递的事件, object 为接收该事件的对象。对象子网中,其他对象产生的事件托肯根据托肯的对象名由接收的对象子网的 IMP 导入,对象子网中产生的所有事件托肯传递给事件库所 EP 处理,然后 EP 根据 object 的值将给自己的事件托肯传给 IMP,给其他对象的事件托肯传给 OMP 导出。

状态图包含状态和迁移。状态包含名称、入口/出口动作、内部迁移、子状态和延迟事件。迁移包含 5 部分:源状态、事件触发器、监护条件、动作和目标状态。

将状态图转换为对象子网时,状态图的状态对应对象子网的库所,状态中的内部迁移和内部活动被忽略。状态图的迁移对应对象子网的变迁,迁移的监护条件对应变迁的发生条件。入口动作、出口动作和动作都产生一个事件发送给 EP 处理。

将状态图转换为对象子网的步骤如下:

步骤 1. 将含有复合状态(状态中有嵌套的子状态)的状态图转化为简单状态图。

(1)因为顺序复合状态中超状态的所有迁移都被每个子状态继承,所以从超状态到其他状态的迁移可以从任何一个子状态迁移,如图 1(a)所示;

(2)并发复合状态时,离开超状态需要两个子状态机同时完成,或其中一个子状态机接收到事件或者结束,并迁移到超状态外面的状态。如图 1(b)所示。

步骤 2. 将简单状态图转换为对象子网。

(1)将状态图的状态变为对象子网的原始库所;

(2)将状态图的迁移变为对象子网的变迁;

(3)增加一个发送消息的库所 OMP、一个接受消息的库所 IMP 和一个处理事件的库所 EP;

(4)入口/出口动作分别产生一个事件托肯,它们分别在进入该状态时和离开该状态时产生。

(5)外部迁移中的动作产生一个该动作的事件托肯,在守卫条件为真时产生。

(6)内部迁移和延迟事件被忽略,因为它们不会导致状态的迁移。状态图中的动作没有相应的对象处理时,该动作在 Petri 网的对象子网中也将被忽略,而不会产生事件托肯。

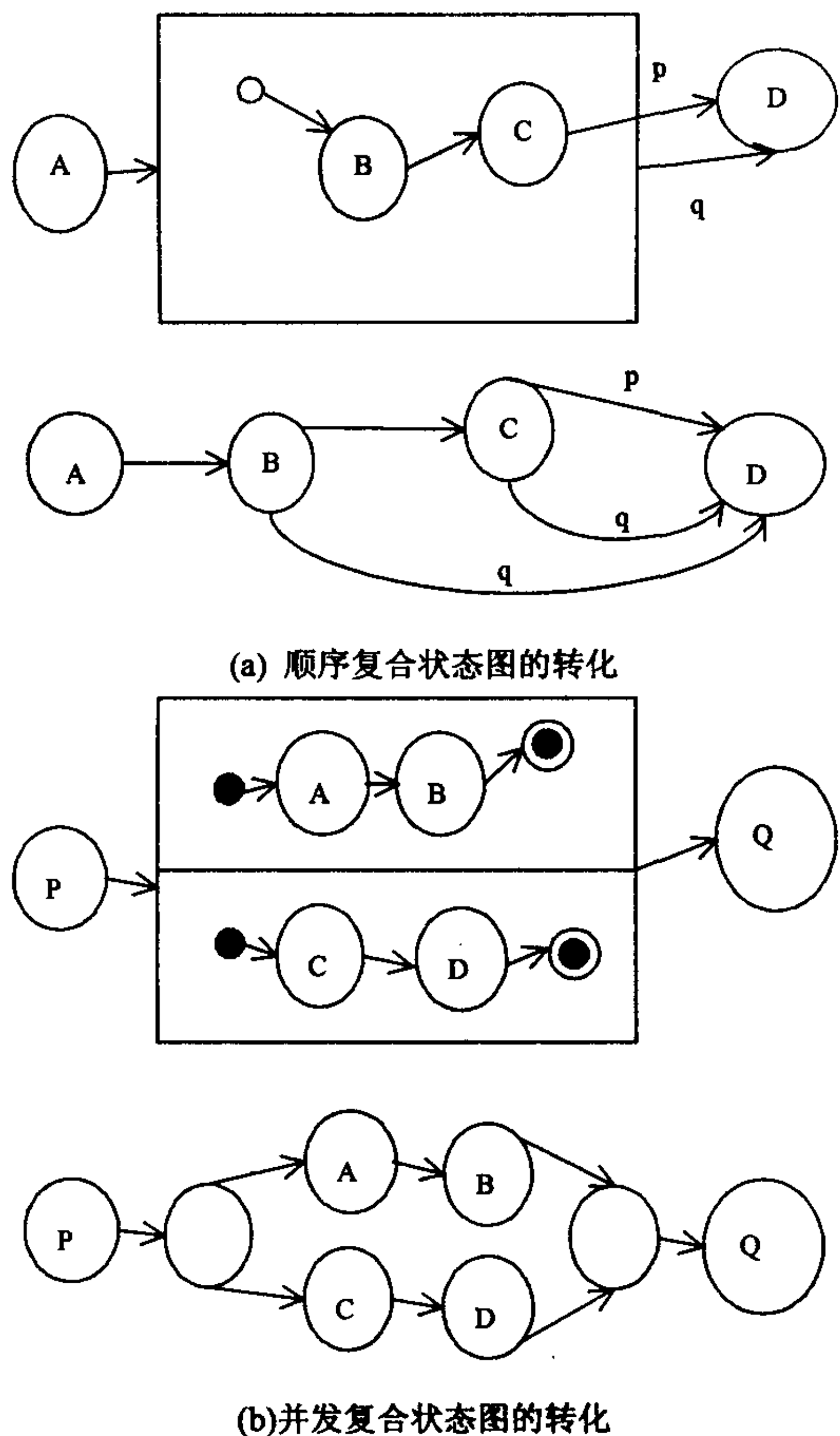


图 1 将复合状态图转换为简单状态图

步骤 3. 根据 UML 协作图将得到的对象子网通过连接库所 LP 连接起来, 得到系统层次的 OCPN 网。LP 负责环境事件托肯的产生和对象子网间事件托肯的转发。它接收对象子网从输出库所传出的事件托肯, 并把事件托肯发送给相应的对象子网的输入库所。

转化过程中, 不对没有接收对象的动作进行建模, 如入口 / 出口动作中, 对简单的动作如果 UML 中没有对动作的对象进行建模, 那么在构造对象子网时, 将不产生该动作的事件托肯, 这样可以简化对象子网的规模, 也可以避免 EP 无法转发产生的事件托肯。对象子网中引入事件托肯, 事件托肯中标示出事件托肯的接收对象, 使 EP 根据对象名决定转发给自己还是外部对象。LP 也可以直接根据对象名找到转发的目标对象。

3 实例

图 2(a)^[5] 为控制器状态图, 控制器初始在 Initial 状态, 自然迁移到 Idle 状态, 然后等待 alarm 信号进入 Active 状态或者用户发出的 attention 信号进入 Command 状态。控制器处于 Active 状态时, 有顺序子状态 Check, Call 和 Waiting。在 Active 的任意子状态中, 可以通过一个 clear 信号回到 Idle 状态或者通过用户发

出的 attention 信号迁移到 Command 状态。

转换后相应的对象子网为图 3(a), 触发变迁需要的事件托肯由 IMP 导入, 产生的事件托肯传给 EP, 从 Idle 迁移到 Check 时产生事件托肯 C1: (setAlarm, Alarm), 发送给 Alarm 对象。从 Check 迁移到 Call 时产生事件托肯 C2: (callCenter, Center), 发送给 Center 对象。从 Check, Call, Waiting 遇到信号 clear 时都迁移到 Idle, 产生事件托肯 C3: (clearAlarm, Alarm), 发送给 Alarm 对象。在图 3(a) 中标记 3 的变迁处各有一条从变迁到 EP 的弧, 弧上的托肯为 C3, 由于复杂度未在图上标出。C1, C2, C3 均发送给 EP, 由 EP 将托肯中对象名不是自己的转发给 OMP 导出。图 2(b) 为用户状态图, 转换后相应的对象子网为图 3(b), 产生的事件托肯通过 EP 转发给输出库所 OMP。

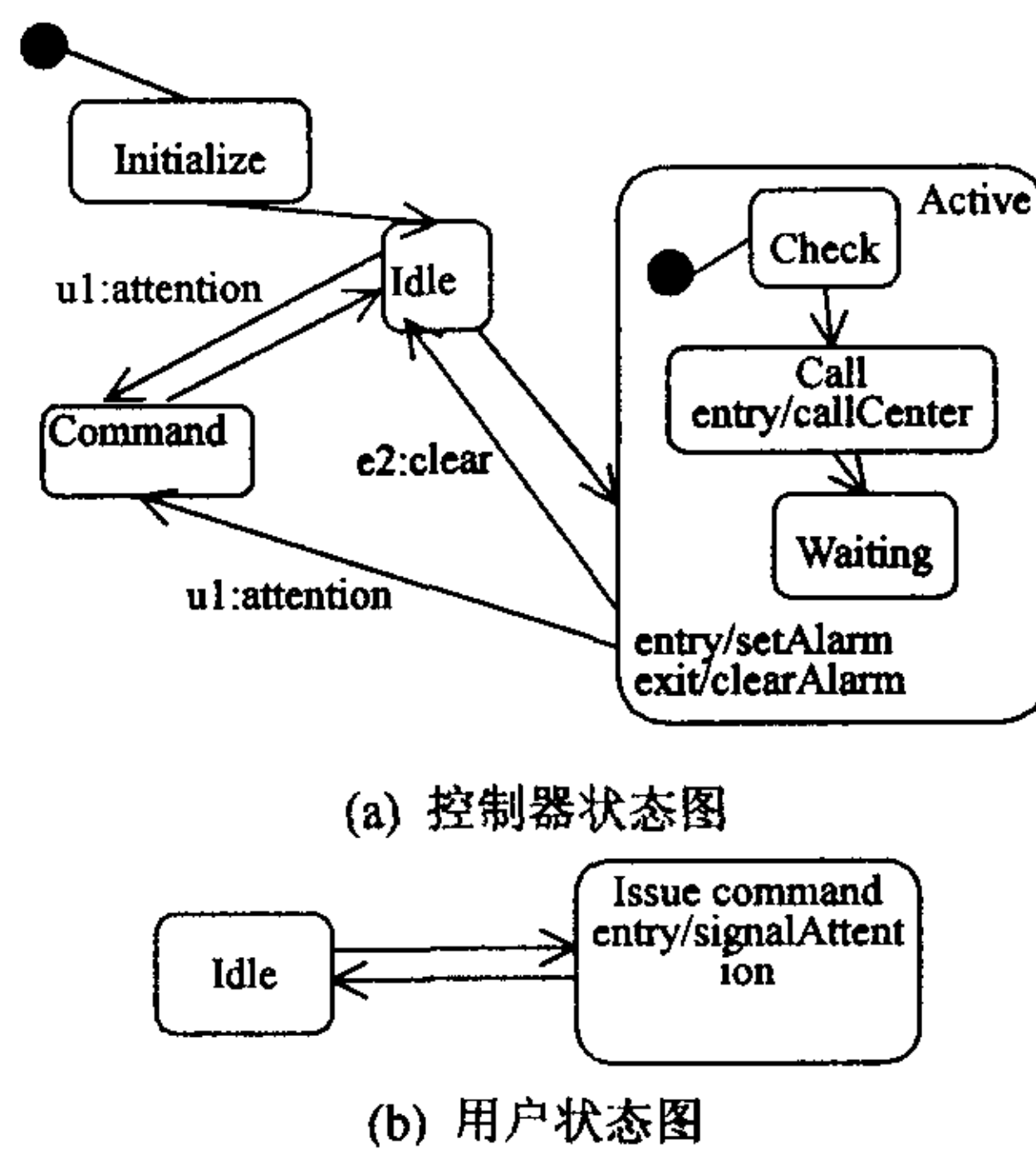


图 2 控制器状态图 and 用户状态图

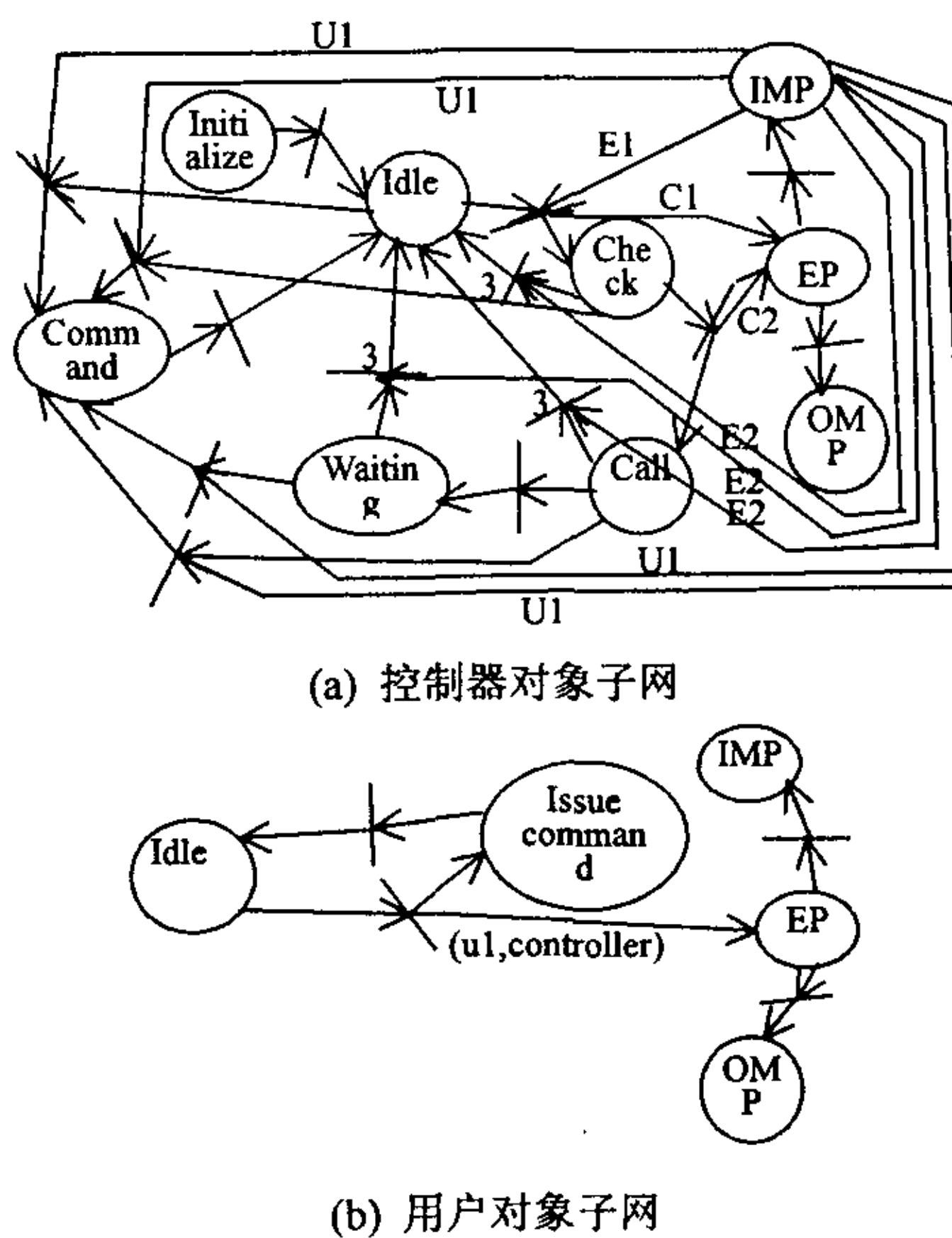


图 3 控制器对象子网和 用户对象子网

图 3 中,
 U1:(attention, controller);
 E1:(alarm, controller);

E2:(clear, controller);
 C1:(setAlarm, Alarm);
 C2:(callCenter, center);
 C3:(clearAlarm, Alarm)

图 4(a)为控制器和用户的协作图,图 4(b)为根据协作图将对象子网连接的系统 OCPN 模型。LP 接收用户对象子网 OMP 传出的事件托肯,并将托肯转发给控制器对象子网的 IMP。

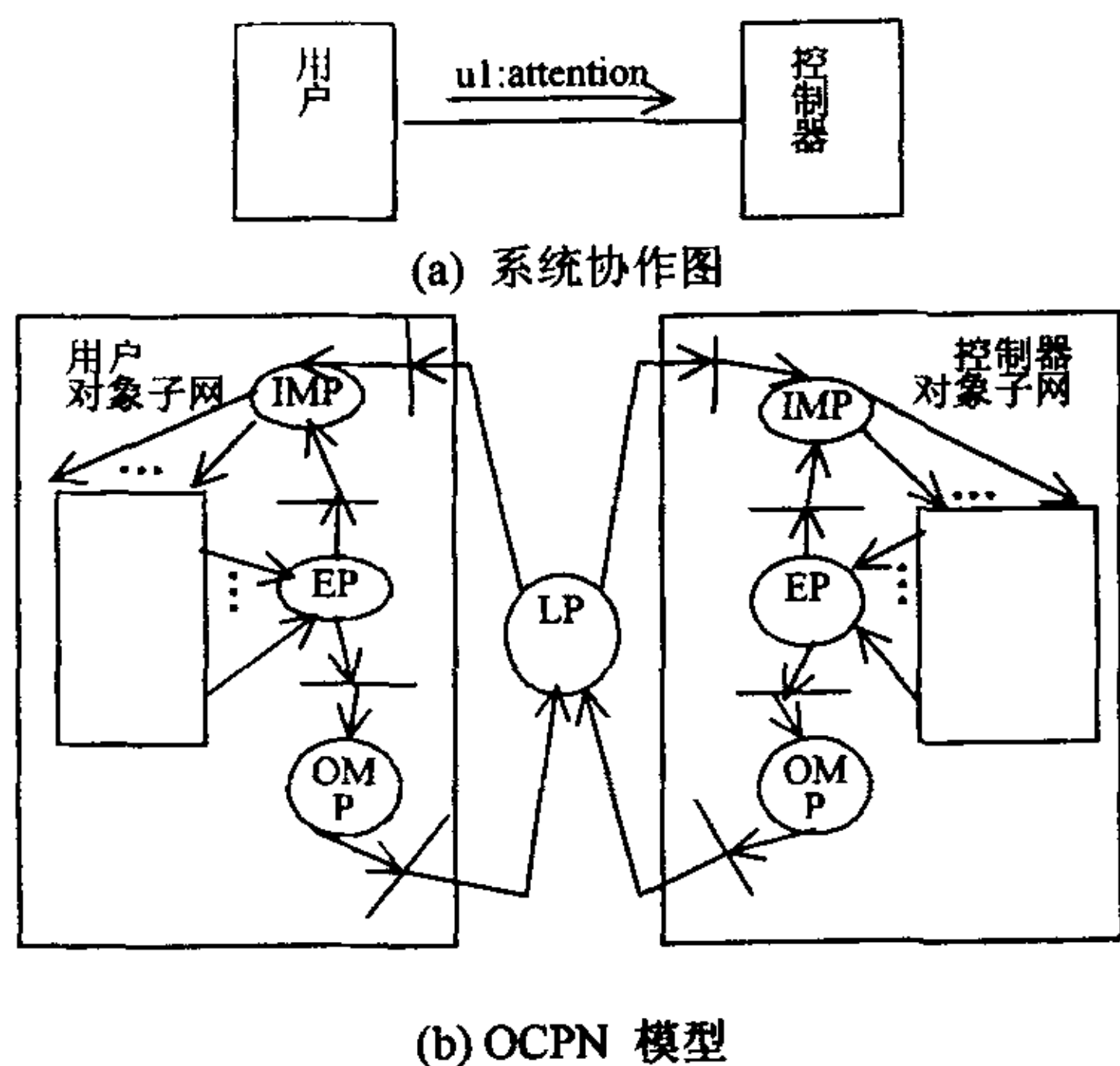


图 4 系统协作图和 OCPN 模型

4 模型分析

模型建立后,需要验证模型是否满足一些好的性质以及不具备坏的性质(如:准活性、无死锁性、活性、有界性、家态、不可回避状态等)。

OCPN 作为着色 Petri 网的扩展形式,具有着色 Petri 网的所有性质。因此,可利用着色 Petri 网的方法对模型进行分析。Petri 网的验证方法十分多,主要有:基于状态空间方法(基于偏序方法、符号化和参数化方法、on the fly 方法);结构分析方法(线性代数技术与凸几何学、图论);演绎和基于进程代数的方法等。

1)无死锁性:指从系统任何可达状态出发,不存在除终止标识的死标识。死标识是指在标识下,没有变迁是可发生的,即系统没有任何部分可以演进,到达一个全局死锁状态。无死锁性是活性的必要条件。

用线性代数的技术分析,设 S 是一个具有关联矩阵 C 的网系统,如果 m 通过发生序列 σ 是从 m_0 可达的,则有 $m_0 + C \cdot \sigma$ 。该状态方程的自然解 (m, σ) 的集合 $RS(s)$ 定义了可达集合的一种线性化。设 $\alpha: T \rightarrow P$ 是一个映射,将每个变迁赋值为它的输入库所之一。如果不存在系统

$$m = m_0 + C \cdot \sigma \quad m \geq 0, \sigma \geq 0$$

$$m[\alpha(t)] < Pre[\alpha(t), t]; \forall t \in T$$

有解,那么网系统 $\langle N, m_0 \rangle$ 是无死锁的。网系统的无

死锁性可以通过验证一个线性不等式系统的无解来证明,该方法是可判定的。

在图 3(a)的 Petri 网用 CPN Tools(Design CPN)中建模,若初始状态时接口库所 IMP 中的托肯为 1'E1, Initialize 中为任意托肯 1'I,系统在状态 Waiting 处没有变迁可以发生,即在这种初始标识下系统会陷入死锁状态。

2)可达性:描述从初始标识开始,一些标识是否可达。可达标识集是从初始标识 M_0 出发有限步之内可到达的标识组成的集合。可达标识是状态方程的解。

3)活性:设 $t \in T$ 为任一变迁,若对任一可达标识 $M \in [M_0 >$, 总有从 M 可以到达的标识 $M' \in [M >$, 使得 $M'[t >$ (即 t 在 M' 有发生权),就说变迁 t 是活的;若对所有变迁 $t \in T$ 都是活的,就说系统是活的。

4)有界性:若对所有 $M \in [M_0 >$, 存在正整数 k , 使得 $\forall p \in P: M(p) \leq k$, 就说系统是 k 有界的。

在图 3(a)的 Petri 网中,因为是描述控制器的单一对象,在同一时间只有一个事件发生,因此描述对象状态的原始库所中最多只有一个托肯,是 1 有界的。

5)活锁:活锁是指系统陷入循环后没有可跳出循环的标识。

图 3(a)的 Petri 网不存在活锁。用 CPN Tools 对 Petri 网进行建模和仿真,可以得到模型的状态空间、有界性、活性等信息。

5 结束语

文中改进了一种将 UML 模型转化为对象着色 Petri 网的方法,并对建造的模型进行分析验证。在硬件和协议设计领域,模型验证已经取得很大成功,而在软件设计领域,由于软件的规模和复杂度,还没有有效的模型分析检测的方法,通过将 UML 模型转化为 Petri 网模型,然后对 Petri 网模型的分析验证,可以找出系统设计的缺陷,提高系统的正确性和安全性。如何对模型进行简化,降低模型的复杂性,是今后研究的方向。

参考文献:

[1] Schäfer T, Knapp A, Merz S. Model Checking UML State Machines and Collaborations[J]. Electronic Notes in Theoretical Computer Science, 2001, 55(3): 357 - 369.
 [2] Girault C, Valk R. 系统工程 Petri 网——建模、验证与应用指南[M]. 北京:电子工业出版社, 2005.
 [3] Rumbaugh J, Jacobson I, Booch G. The unified modeling language reference manual[M]. 北京:科学出版社, 2004.

(下转第 188 页)

等。这个时候不同用户输入不同的短语期望得到的结果应该是类似的。另一方面,由于使用不同的元搜索引擎,每个搜索引擎都有自己对搜索结果的排序算法,使得不同搜索引擎的搜索结果的排序无法比较,搜索结果与用户查询之间的相关程度也无法建立统一的度量标准,因此查询结果整合问题成了搜索引擎研究的核心问题。所以文中提出的模型利用聚类有关方法对网页集合按照语义信息进行初步归类,然后对每个类中的网页利用传统的链接分析算法计算网页权重。基本步骤是:

(1)对各搜索引擎的查询结果进行聚类分析,形成对查询结果的自动分类。

聚类是由计算机系统按照一定的要求将相似或者相同特征的对象聚合在一起的过程。自动聚类根据对象的不同特征划分成不同的类,使得同一个类中的对象之间的差别尽可能地小,而不同类中的对象之间的差别尽可能地大。聚类分析在机器学习、模式识别等领域已有不少应用,也已提出了不少聚类算法,如 C-均值聚类法、层次聚类法、密度聚类法、网格聚类法等,其中 K-均值聚类法以其计算的高效率而被广泛采用。

网页聚类算法如下:

首先生成初始类,在网页集中任意选取一个网页作为第一个类,计算其余网页与此网页的相似度(链接相似度),选取相似度最小的一个网页作为第二个类,计算其余网页与此二网页的相似度,取与这两个网页相似度之和最小的网页作为第三个类。依此直到产生 K 个类(K 为事先给定的正整数),然后根据网页对类的隶属度反复迭代直到所有类不再发生变化为止。

(2)分析对各类和用户查询请求之间的相关度,并根据相关度确定类排序。

在确定了各个类之后需要进一步对类内各结果项进行排序,查询结果排序类似于类排序的策略,通过计算每一个结果项和查询请求之间的相似度并结合在元搜索引擎中的排序,最终确定各个结果项在类中的排序,最后根据和用户特征之间的相似度,确定类排序。

(3)将相关度最大的类中的结果返回用户。

2.4 用户个性化模型的建立

最终确定的类排序与用户特征有着密切的联系,系统根据不同的用户特征给出的输出结果是不一样

的,从而体现个性化的输出结果,这一功能的实现依赖于用户个性化模型的建立。系统可以使用户在 I/O 界面中输入自己的账号和密码,用户信息收集模块用来记录用户的历史搜索记录,以及 Web 日志中包括像 IP、访问时间、请求页、访问页、会话号、浏览器版本等信息,同时用户浏览返回的结果,确认是否为感兴趣文档,若为感兴趣文档则做一标记,每个被确认的文档都有其对应的关键词,将这些关键词添加到用户所对应的数据库中,用于下次搜索时的用户行为分析。用户群聚类模块可以根据这些关键词的集合以及日志信息对不同的用户进行聚类,如果两个用户的关键词集合的相似度在给定的阈值 f 之上,则认为两个用户是同类的用户。所有这些信息被记录在用户个性化模式库中。

3 结 论

随着 Internet 的不断发展,搜索引擎已经成为人们获取网上信息最重要的途径,但存在着命中率低、成本高等不足^[5]。文中介绍的个性化搜索引擎针对传统搜索引擎的不足进行了相应的改进,将元搜索引擎技术和聚类技术结合到一起,元搜索引擎技术提高了传统搜索引擎的查准率和查全率,聚类技术充分考虑到用户的个性化,因此有很强的理论意义和实际意义,具有一定的智能性,能够应用到 Internet、信息检索、电子商务等诸多领域。

当然系统中也有一些问题:如何用更加优化的算法提高搜索的效率;如何使用多种显示方式对搜索结果进行显示,以满足不同用户的浏览习惯,增加可读性和生动性,这是今后工作需要进一步研究的问题。

参考文献:

[1] 曹二堂,刘玉林. 基于语义理解的智能搜索引擎的研究[J]. 情报杂志,2005(6):58-59.
 [2] 胡亮,许永诚,高文,等. 个性化高效元搜索引擎的设计与实现[J]. 计算机工程与设计,2005,26(4):896-899.
 [3] 刘丽,孙燕唐. 智能型元搜索引擎的设计与实现[J]. 计算机工程,2003,29(6):118-120.
 [4] 刘炜,陈俊杰. 一种基于 Agent 的智能元搜索引擎框架[J]. 计算机工程与应用,2005(3):137-138.
 [5] 左雄辉,糜麒. 个性化搜索引擎研究[J]. 计算机工程与应用,2005(17):190-192.

(上接第 185 页)

[4] Saldhana J A, Shatz S M. UML diagrams to Object Petri Net Models: An approach for Modeling and Analysis[C]//Proceedings of Twelfth International Conference on Software En-

gineering and Knowledge Engineering (SEKE2000). Chicago, USA: Knowledge System Institute, 2000:103-110.
 [5] Booch G, Rumbaugh J, Jacobson I. UML 用户指南[M]. 北京:机械工业出版社,2001.