

基于 URN 的特征冲突过滤方法

张睿, 刘晓霞

(西北大学信息科学与技术学院, 西安 710127)

摘要: 基于用例图的冲突过滤算法是在使用用例图描述呼叫流程的基础上, 利用矩阵运算进行冲突过滤。为解决原算法中矩阵合并开销大及过滤效率不高的问题, 为算法增加前提条件, 过滤部分不会发生冲突的业务组合, 减少结论中类型不确定的组合, 从而提高算法的过滤效率, 并与其他改进的冲突过滤算法进行比较。

关键词: 特征冲突过滤; 特征组合; 用户要求表示规范

Feature Interaction Filtering Method Based on URN

ZHANG Rui, LIU Xiao-xia

(College of Information Science and Technology, Northwest University, Xi'an 710127)

【Abstract】 Interaction filtering algorithm based on use case diagram uses matrix operation to filter interaction by describing call flow with use case diagram. In order to solve the problems of large time consumption for matrix operation and of low filtering efficiency in original algorithm, this paper adds pre-conditions to filter feature combinations that will not conflict, and reduces the combinations of uncertain types, so that filtering efficiency of the algorithm is improved. The algorithm is compared with another improved interaction filtering algorithm.

【Key words】 feature interaction filtering; feature combination; User Requirements Notation(URN)

1 概述

在软件系统中, 通过扩展系统的基本功能得到的各项附加功能称为特征, 软件系统可以表示为 $BF+F_1+F_2+\dots+F_n$, BF 表示系统的基本功能, F_i 表示特征, 一个特征可能会修改或者破坏另一个特征的正常功能, 这种情况称为特征冲突。在电信系统中, 特征冲突很普遍, 研究者提出了很多解决方法, 包括形式化方法, 如使用扩展有限状态机方法检测死锁、死循环、不确定性等冲突, 使用 Petri net 对电信业务进行建模^[1]。然而这类方法还未在工业界得到广泛的应用。为此引入各种可视化方法特别是用例图以帮助对特征冲突进行分析。在用例图研究的基础上, ITU-T 于 2000 年 11 月形成用户需求表示规范 URN (User Requirements Notation) 建议草案, 该草案是对用例图的进一步规范和扩展。该建议的目的是用可视的形式方法描述用户的业务需求, 且与实现机制无关, 可以避免过早涉及系统细节。另一方面, 待检测的组合数量会随着特征的数量迅速增长, 因此在特征冲突检测中通过增加过滤机制来降低检测数量逐渐受到研究者的重视。然而大多数研究仅处于用例分析阶段^[2]。文献[3]的方法使用用例图对呼叫流程进行描述并建立数学模型, 通过矩阵运算对冲突进行过滤, 然而结果中存在较多

冗余数据且运算量较大。本文通过为该算法增加前提条件, 在矩阵运算之前过滤掉部分不会发生冲突的业务组合, 从而减少矩阵运算次数及结果中的冗余数据。

2 基于 URN 的冲突过滤方法

基于 URN 的冲突过滤方法^[3]首先通过用例图对基本呼叫流程进行描述, 如图 1 所示。

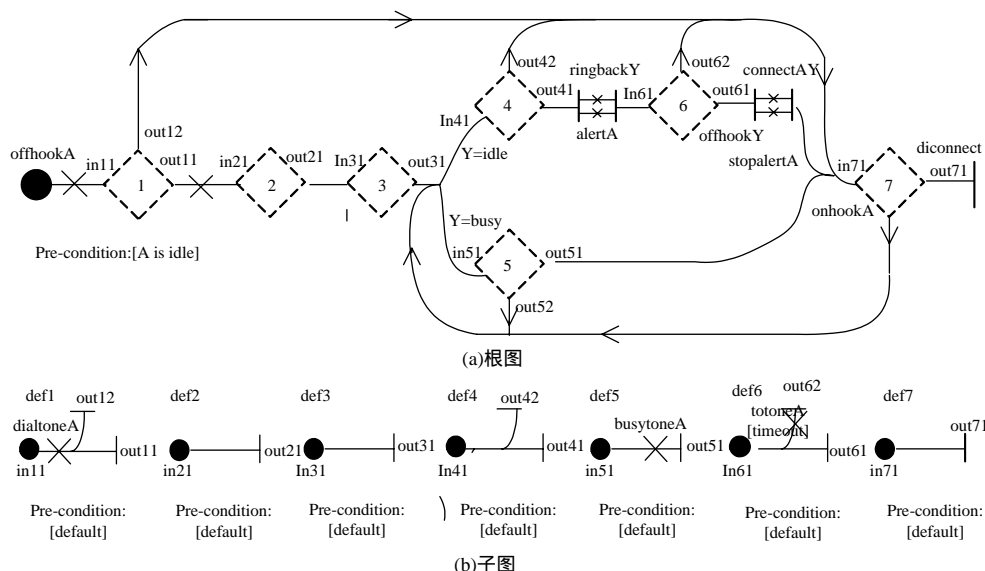


图 1 基本呼叫流程的 URN 描述

基金项目: 陕西省自然科学基金资助项目(2006F50); 航空科学基金资助项目(2006ZC 31001)

作者简介: 张睿(1984-), 女, 硕士研究生, 主研方向: 网络信息处理; 刘晓霞, 教授

收稿日期: 2009-02-20 **E-mail:** zhangrui1234566@163.com

根图对呼叫流程进行描述,包括7个桩节点(根图中的虚菱形),每个子图代表在相应节点的默认操作,在缺省情况下分别装配于桩节点。从根图的起点(圆点)到终点(粗线段)的每一条可能路径形成一次可能的呼叫过程。智能网业务是对基本流程的某些部分(桩节点)进行修改或增强,对这些核心功能进行描述,并将其装配于根图中的桩节点,即可对智能业务进行描述。

用户定制智能业务后,桩节点会被具体业务的子图替代。为了对特征进行统一描述,将桩节点描述为矩阵形式,如:当用户B定制了CNDB特征时,用户B的呼叫流程可表示为: $Vb=(def1, ccA2, cndbB3, def4, def5, def6, def7)$,其中,def代表对应节点激活的是默认操作;cndbB3代表该节点激活的是CNDB特征的操作。对每个用户的呼叫流程进行描述后,即可得基桩矩阵,如式(1)所示。得到每个特征的矩阵描述后,即可通过矩阵的合并和冲突检测规则进行研究。

$$TCS_A = \begin{bmatrix} def1 & def2 & def3 & def4 & def5 & def6 & def7 \\ def1 & def2 & def3 & def4 | tcsA4 & def5 & def6 & def7 \\ def1 & def2 & def3 & def4 | tcsA4 & def5 & def6 & def7 \end{bmatrix}$$

$$CNDB_B = \begin{bmatrix} def1 & def2 & def3 & def4 & def5 & def6 & def7 \\ def1 & def2 & cndbB3 & def4 & def5 & def6 & def7 \\ def1 & def2 & def3 & def4 & def5 & def6 & def7 \end{bmatrix} \quad (1)$$

3 基于URN的特征冲突过滤方法的改进

3.1 算法分析与改进思路

每种特征组合都要进行矩阵合并并对合并后矩阵进行判断,开销比较大。如果有 n 个特征,组合的次数即为 n^2+n 。并且最后的结论中,冗余数据较多,在文献[3]的试验结果中,64种组合中能确定是否发生冲突的组合数为14,仅占22.9%,其余50种仍需要继续进行检测(占78.1%),而其中实际发生冲突的仅23种,占46%,冗余数据比较多。

鉴于以上缺点,本文通过增加前提条件在矩阵合并之前对不会发生冲突的情况进行过滤。根据参与者对特征的描述,提出规则并对其正确性进行证明。

双方电信业务根据参与者数量的不同可以分为非前转类业务和前转类业务,前者包含主叫号码和被叫号码2个参与者;后者包含主叫号码、被叫号码和前转号码3个参与者(对业务特征也采用相同的分类方式)。通过特征的参与者对特征进行描述,如:特征R的参与者为主叫号码A,被叫号码B,呼叫流程即可表示为A-B,同理,特征U可表示为C-D。如果2个特征发生冲突,两者必定存在关联,2个特征有重合的参与者^[4],特征R和U冲突必然存在A=C且B=D的关系,即2个呼叫流程可以统一为相同的主叫用户和被叫用户。为原算法增加以下前提条件,在矩阵合并之前进行过滤(暂不考虑电话会议类业务):

前提条件1 当2个特征都属于非前转类时,特征X的参与者为A,B,呼叫流程表示为(A-B)(对于闲时回叫这类会造成主被叫用户互换的特征,将其表示为B-A,后面步骤不变)。特征Y的参与者为C,D,呼叫流程表示为(C-D)。2个特征存在冲突的条件为: $\{A=C, B=D\}$ 。

前提条件2 当2个特征同时包括非前转类和前转类时,将前转类特征表示为Y:(A-B) (A-C)。另一特征表示为Z:(D-E),此时若满足条件 $\{D=A, E=B\}$ 或 $\{D=A, E=C\}$,则可能发生冲突。

对于前转类特征Y,当特征的定制者为B且C是转接对象时,对于其他呼叫流程,如B拨打A、B拨打C,特征Y

不会被激活,呼叫过程为A-B-C,对于非前转类特征Z,只有(D-E)与(A-B), (B-C)或(A-C)可以统一为相同的主被叫用户时,才可能发生冲突,即满足以下条件之一:(1) $\{D=A, E=B\}$; (2) $\{D=A, E=C\}$; (3) $\{D=B, E=C\}$ 。但是满足条件(3)时,2个特征并不一定同时激活,在呼叫转接后,如果(B-C)是用户B定制的特征(即Z由用户D定制),在A呼叫B后,由于B是被叫用户,特征Z并不会被激活,因此不会冲突;如果(B-C)是用户C定制的特征,则可能与Z冲突,但此时产生的冲突与条件(2)产生的冲突原因相同。因此,将非前转特征Y表示为:(A-B) (A-C),由此得到前提条件2。

前提条件3 当2个特征都属于前转类时。将其表示为:Y:(A-B) (A-C), Z:(D-E) (D-F),此时,如果满足以下条件: $\{D=A, E=B\}$, $\{A=D, C=E, A=D, B=F\}$,就可能发生冲突。

由前提条件1可知, $\{D=A, E=B\}$ 可能引起冲突。当 $\{A=D, C=F\}$ 时,代表2个特征由同一主叫用户激活且前转号码相同,如果此时2个特征的被叫用户相同,则冲突情况同前;如果不同,2个特征并不会同时被激活,即不会发生冲突。当 $\{A=D, C=E, A=D, B=F\}$ 时,代表特征Y被激活后,所发生的呼叫转移会进一步激活特征Z,同理,特征Z的激活会进一步激活特征Y,此时发生活锁冲突。而当 $\{A=D, C=E\}$ 与 $\{A=D, B=F\}$ 有一个条件不满足时,活锁冲突不会发生。

3.2 改进的算法

将上述比较规则应用于原算法矩阵合并之前,不满足条件的组合不会发生冲突,可以将其过滤,借此减少下一步矩阵合并的次数。

(1)根据3个前提条件过滤特征在激活时相互没有关系的组合,其余组合执行步骤(2)。

(2)对矩阵进行合并: $H=F \oplus G$,如式(2)所示(采用式(1)所描述的特征):

$$TCS_A \oplus CNDB_B = \begin{bmatrix} def1 & def2 & def3 & def4 & def5 & def6 & def7 \\ def1 & def2 & cndbB3 & def4 | tcsA4 & def5 & def6 & def7 \\ def1 & def2 & def3 & def4 | tcsA4 & def5 & def6 & def7 \end{bmatrix} \quad (2)$$

(3)如果H中包括ng,得出结论:冲突发生;否则,执行步骤(4)。

(4)如果H中第i行等于矩阵F中第i行或者矩阵G中第i行,且H的每行都满足该条件,则得出结论:不会发生冲突;否则,得出结论:可能发生冲突。

4 算法结果比较

算法结果如图2所示,其中所检测的特征选自文献[5](特征冲突竞赛),分别为遇忙前转CFB、儿童限制呼叫TL、被叫屏蔽TCS、对方付费RC、呼叫号码传送限制CNDB、闲时回叫RBF、信箱提示VM、分摊付费SB,这8种特征包括前转业务与非前转业务,其中,RBF属于会使主被叫用户发生互换的特征,类型覆盖较全,可以保证适用性。根据文献[5]的结果,图中的阴影代表确定发生冲突的情况,其中,第2行的s代表2个特征由同一用户定制,d代表2个特征由不同用户定制。对于过滤结果中出现2个字母的情况,第1个字母代表原算法的过滤结果,第2个字母代表使用前提条件可以过滤的情况,其中,a代表冲突;b代表不冲突;c代表可能冲突。8种特征共存在64种组合情况,通过前提条件可以过滤23种不会发生冲突的组合,占总组合数目的36%,显著减少了下一步矩阵合并的次数。其次,前提条件可以过滤一部分由于不能同时激活而不会发生冲突的组合,即改进的算

法可以过滤更多不发生冲突的组合，因此，最后结论中的 c 类减少了 12 种，c 中实际发生冲突的组合增加为 60.5% (原为 46%) 过滤比率(a 与 b 的比例)增加为 40.6% (原为 22.9%)，减少了结果中的冗余数据。再次，前提条件可以过滤大部分原算法结论中不会发生冲突的组合，具体来说，原算法结论中不发生冲突的 11 种组合通过前提条件可以过滤掉 10 种。

CFB		TL		TCS		RC		CNDB		RBF		VM		SB		
s	d	s	d	s	d	s	d	s	d	s	d	s	d	s	d	
	c	bb	b	c	c	c	c	bb	c	a	c	c	c	c	c	CFB
		bb	bb	c	bb	c	c	bb	c	cb	c	cb	c	bb	c	TL
				cb	c	cb	bb	c	cb	c	cb	c	c	cb	c	TCS
						cb	bb	c	cb	c	cb	c	a	cb	c	RC
								bb	c	c	c	c	bb	c	c	CNDB
										c	c	c	c	c	c	RBF
												c	c	c	c	VM
														cb	c	SB

图 2 算法结果比较

在另一组实验中，选取了同样的 8 种特征。实验结果表明，通过本文方法可以过滤 26 种组合(占 41%)，所有组合中不会发生冲突的组合(20 种)可以被前提条件全部过滤。进一步验证了前面的结论。

在实验中发现：特征类型较全时过滤效果较好；由于前转业务的复杂性，前 2 个条件的过滤效果优于条件 3；通过与文献[5]中的结果比较，几组实验中被过滤的组合实际上都不会发生冲突，保证了冲突过滤方法的正确性。

另外，前提条件的使用减少了需要使用原算法进行过滤的特征组合数目，剩余特征组合的冲突情况与原算法相同。前提条件的增加虽然增加了算法的运行时间，但同时减少了矩阵合并及判断的次数，并且由于前提条件仅需判断 1 次，低于矩阵运算复杂度(n^2)，因此算法的整体运行时间减少了。

文献[3]对可能引起冲突的一类特征组合(结论 c)进行了进一步研究，将用例图中的动态桩节点替换为实际中会执行的特征，从中得出 2 条规律，并根据该规律对可能发生冲突的组合进行分类，不满足任一规律的组合不会发生冲突。该方法将算法的过滤比率从 22.9% 提高到 34.4%。但该方法需要对呼叫流程的细节进行分析，本文方法可以在忽略流程细

节的情况下提高算法的过滤效率，减少下一步需要进行矩阵合并的次数，易于操作，更适用于冲突检测之前的过滤。并且由于所增加的前提条件可以过滤绝大部分不会发生冲突的组合，因此可以单独抽取出来使用。

5 结束语

本文通过改进文献[3]的算法，减少了矩阵的运算次数、判断次数以及下一步需要检测的组合数，有效提高了冲突特征过滤方法的效率，还进一步提高了算法过滤的比率，减少了最后结论中不确定是否发生冲突的组合数。由于所增加的前提条件对于不可能发生冲突的组合的过滤效率较好，因此可以单独使用。在不同的抽象层次对特征冲突进行研究时，先在不涉及细节的情况下过滤一部分特征组合，可以减少下一步根据细节进行过滤的次数，有利于简化对特征冲突的研究。随着技术的发展，智能网业务的类型也在不断发展，复杂业务的呼叫模型及过滤策略将是下一步的研究方向，对 Web 服务中的冲突检测也将是一个研究热点。

参考文献

- [1] 王嘉辉, 陆以勤, 利文乐. 基于 CPN TOOLS 的电信业务建模及其应用[J]. 计算机工程, 2006, 32(14): 221-223.
- [2] Calder M, Kolberg M, Magill E H, et al. Feature Interaction: A Critical Review and Considered Forecast Computer Networks[J]. International Journal of Telecommunication and Computer Network, 2003, 41(1): 115-141.
- [3] Leelaprute P, Nakamura N, Matsumoto K, et al. Design and Evaluation of Feature Interaction Filtering with Use Case Maps[Z]. [2008-11-30]. <http://www.nectec.or.th/NTJ/No16/papers/01.pdf>.
- [4] Sefidcon A, Khendek F. A Pragmatic Approach for Feature Interaction Detection in Intelligent Networks[C]//Proc. of the 8th International Conference on Computer Communications and Networks. Boston, USA: [s. n.], 1999: 622-627.
- [5] Kolberg M, Magill E H, Maples D, et al. Second Feature Interaction Contest[C]//Proc. of FIW'00. Scotland, UK: IOS Press, 2000: 293-310.

编辑 张帆

(上接第 44 页)

```

    this.password = password;
}
public String getPassword() {
    return this.password;
}
public boolean doLogin(String userName, String password) {
    //TODO Auto-generated method stub
    return false;
} }

```

6 结束语

本文开发的 ComponentStudio 工具能帮助开发者对构件进行可视化建模，验证模型的正确性并自动生成代码框架。目前，该工具已被应用于上海万达信息股份有限公司，它作为其开发工具 CuteStudio 中的一个子工具，能很好地支持构件开发、提高软件开发效率。

但 ComponentStudio 工具目前只支持 POJO 构件和 Spring

构件的制作与组装，将来的方向是支持更多种类的构件，如 EJB 构件和工作流构件。

参考文献

- [1] Kleppe A, Warmer J, Bast W. MDA Explained: The Model Driven Architecture: Practice and Promise[M]. [S. l.]: Addison-Wesley, 2003.
- [2] 叶菲. 一个基于 Java 的面向 Web 应用开发的构件组装工具[D]. 上海: 复旦大学, 2006.
- [3] Riba N, Cervantes H. A MDA Tool for the Development of Service-oriented Component-based Applications[C]//Proc. of the 8th Mexican International Conference on Current Trends in Computer Science. [S. l.]: IEEE Computer Society, 2007.
- [4] OSOA. SCA Assembly Specification[Z]. (2007-04-12). http://www.osoa.org/download/attachments/35/SCA_AssemblyModel_V100.pdf.

编辑 陈晖