

基于情境感知事件的主动服务发现方法*

莫同[†], 褚伟杰, 李伟平, 吴中海, 林慧苹

(北京大学 软件与微电子学院, 北京 100871)

Active Service Discovery Method Based on Context Aware Event

MO Tong[†], CHU Wei-Jie, LI Wei-Ping, WU Zhong-Hai, LIN Hui-Ping

(School of Software and Microelectronics, Peking University, Beijing 100871, China)

+ Corresponding author: E-mail: motong@ss.pku.edu.cn

Mo T, Chu WJ, Li WP, Wu ZH, Lin HP. Active service discovery method based on context aware event. Journal of Software, 2011, 22(Suppl. (2)): 41-51. <http://www.jos.org.cn/1000-9825/11025.htm>

Abstract: With the development of service computing and Internet of things, software systems can discover and provide service to customer actively based on context information. Compared with traditional service discovery, the service requirement is unknown in active service discovery. The system needs to analyze current demands from customer's context-aware information and choose the proper service to provide. By using this as the focus, an active service discovery method based on context-aware event is proposed. First, the change of context-aware information is defined as context event, and the relationship between context events is expressed by event driven graph. Second, an event-service FP-TREE is built by mining service log. On the basis of the two definitions, a service discovery algorithm is realized according to the current context-aware event. Experimental results show that comparing with broadcasting which is the general active approach; this approach can promote the ratio of precision of active service discovery.

Key words: Internet of thing; active service discovery; context aware; event driven graph; event-service FP-TREE

摘要: 随着服务计算和物联网技术的发展, 应用软件系统可以根据情境信息主动地为用户发现和提供服务. 与传统的服务发现相比, 主动服务发现的服务需求未知, 系统需要根据采集的情境信息分析顾客服务需求, 然后选择恰当的服务并加以提供. 为此, 提出一种基于情境感知事件的主动服务发现方法. 首先, 将感知的情境信息变化定义为情境事件, 通过事件驱动图的方式定义情境事件之间的驱动关系; 其次, 对已有服务记录进行挖掘, 通过事件-服务 FP-TREE 描述服务与事件的关联关系; 然后, 在二者的基础上, 根据当前感知的情境事件给出一种服务发现算法. 实验结果表明, 与现有广播式主动服务方法相比, 该方法具有更高的查准率.

关键词: 物联网; 主动服务发现; 情境感知; 事件驱动图; 事件-服务 FP-TREE

随着信息通信技术与传感器技术的迅猛发展和广泛而深入的应用, 地球上的万事万物逐渐变得可感知和互联互通. 互联网和物联网将现实世界和信息世界充分覆盖与融合, 为信息采集、传递和服务决策提供强有力

* 基金项目: 国家自然科学基金(61033005); 国家高技术研究发展计划(863)(2009AA04Z120); 丹麦科技创新部国际合作研究项目(2106-08-0046)

收稿时间: 2011-05-02; 定稿时间: 2011-11-07

的技术支撑.基于情境感知的主动服务是在这些技术背景下产生的一种能够极大改善人们生活的新的服务方式^[1].

情境信息是那些与系统功能和用户需求密切相关的各种信息,如用户的基本资料、位置、时间、自然环境、计算环境等^[2].通过实时获取这些情境信息并据此做出综合判断,进而由应用系统主动地为用户提供合适的服务就是基于情境感知的主动服务.其目标是通过系统自动进行而非用户驱动,以期通过主动的方式向用户提供及时、准确、个性化的服务,提升用户的服务体验.主动服务的核心问题在于如何根据获取的情境信息发现服务提供的时机和内容^[3].

目前,相关的研究工作主要面向被动式服务的发现,系统通过用户的服务请求驱动发现并提供服务,主要技术包括基于 UDDI 注册中心、通过 Web 服务网站、Web 服务专用发现引擎以及通用搜索引擎 4 种^[4],其核心问题是将已知的服务描述与服务需求约束进行匹配,通过匹配度计算和排序,过滤查找出合适的服务.服务发现首先是基于功能的发现,主要方法有描述文档(如 WSDL)识别^[5]、建立服务索引^[6]以及功能匹配^[7]等.针对具有相同功能的服务,对不同用户而言,它们的在响应时间、连通率、吞吐量和可靠性等质量参数上的表现有所不同,而只有当一个 Web 服务能够满足用户在这些方面的要求时,用户才会考虑这个 Web 服务所提供的功能,因此很多学者在基于功能发现的基础上,提出了基于 QoS 的服务发现^[8,9].随着服务的日益复杂,上述针对语法级服务描述语言和基于关键词的服务发现方法的服务查准率越来越低.这是由于,基于关键字的语法级描述方式无法准确表达复杂的服务需求,导致服务发现无法正确“理解”发现目的.为此,语义 Web 被引入到服务发现领域,通过本体的定义和推理进行基于语义的服务描述与发现^[10].在这方面较具代表性的工作包括 DAML-S Matchmaker^[11]和 Akkiraju 提出的 UDDI 增强^[12].

综合来看,已有的服务发现研究主要面向需求已知的用户驱动发现.这种情况下,服务需求已知,服务发现的目标明确,即使存在动态服务发现,其动态原因是服务资源、运行环境和用户需求发生变化,系统也需要根据变更后的新需求和新的服务候选集重新进行服务发现.而在物联网环境下的主动服务中,服务发现的时机和内容未知,系统需要通过实时监控用户的情境信息判断用户是否需要服务以及服务需求内容.此外,由于用户行为的不确定性,情境信息随时可能发生变化,这些不确定性给服务发现带来新的问题和挑战.

为此,本文提出一种基于情境感知事件的主动服务发现方法.将情境信息定义为情境事件,通过事件驱动图的方式描述情境信息各事件的驱动关系.根据已有的服务历史记录,通过事件-服务 FP-TREE 建立服务与情境事件之间的关联关系.然后,根据感知的当前情境事件,结合事件驱动图和事件-服务 FP-TREE,推测用户当前和未来可能需求的服务,实现主动服务发现.

本文第 1 节进行主动服务模式分析,分析主动服务发现的机理,给出相关概念定义和问题描述.第 2 节介绍主动服务发现方法,包括服务环境分析、服务查找和服务发现算法.第 3 节是案例分析.最后是总结与展望.

1 主动服务模式分析

主动服务模式是指不需要用户提出服务请求,而是由系统根据自动采集的信息,分析用户可能的服务需求,然后主动向其提供的一种新型服务模式.为描述主动服务的实现机理,相关概念的定义如下:

定义 1(服务领域 sd). 服务领域 sd 是指针对特定环境、用户群体所提供的一系列服务的统称.例如机场服务、社区服务、校园服务等.

定义 2(情境信息 cxt). 情境信息 cxt 是标识用户特征的信息,情境信息包含固定信息和动态信息两部分.固定信息标识用户个人特征,如性别、年龄、教育程度、职业、兴趣爱好等;动态信息标识用户所处服务环境特征,如时间、位置、温度、湿度等.

定义 3(情境事件). 情境事件 $event$ 是在服务领域 sd 中,情境信息 cxt 可能发生的具有业务含义的取值改变.

定义 4(服务环境). 假设在服务领域 sd 中存在 m 种情境信息($cxt_1, cxt_2, \dots, cxt_m$),用户当前所处的服务环境 $environment$ 由服务领域内各种情境信息的当前事件所组成,记作 $environment_1 = \{event_{1a}, event_{2b}, \dots, event_{mn}\}$.

定义 5(服务需求). 服务需求 r 是一个二元组 $r = \langle evn, s \rangle$,其中,

- (1) 服务环境 $env=(e_1, e_2, \dots, e_n)$ 是需求 r 发生时的情境事件集合;
- (2) 服务 s 表示在服务环境 env 中, 用户需求的服务.

根据上述定义, 主动服务的实现机理可以概括为, 系统通过周期性地采集情境信息, 感知发生的情境事件, 并根据情境事件推测用户当前或即将所处的服务环境, 然后, 根据服务环境提供相应的服务以满足用户的服务需求. 该实现机理需要解决两个核心问题:

- (1) 服务环境分析: 如何根据感知的情境事件, 分析用户当前或即将所处的服务环境;
- (2) 服务查找: 在某个服务环境下, 需要向用户提供哪种服务.

2 主动服务发现方法

为解决服务环境分析和查找问题, 提出事件驱动图和事件-服务 FP-TREE 概念, 事件驱动图保存情境事件之间的关联关系, 事件-服务 FP-TREE 通过树形结构存储用户需求的服务与发生的情境事件之间的关联关系及其频繁程度. 主动服务发现可以根据感知到的用户当前服务环境及事件驱动图, 分析用户当前及以后可能所处的情境环境, 然后根据服务环境中的情境事件查找事件-服务 FP-TREE, 发现对应的服务.

2.1 服务环境分析

主动服务系统通过感知情境事件确定用户的服务环境. 对于某个情境信息 cxt 而言, 根据业务含义, 其相关的情境事件之间存在一定的发生偏序关系. 这种偏序关系可以通过图来加以表示.

定义 6(事件驱动图). 服务领域 sd 中情境信息 cxt 相关的情境事件可以被定义成一个有向图 $EDG=(sd, cxt).V, E$, 其中,

- (1) 顶点集 $(sd, cxt).V=\{v_1, v_2, \dots, v_m\}$ 为情境事件集, 表示在服务领域 sd 中, 情境信息 cxt 可能发生的具有业务含义的取值改变.
- (2) 边集 $(sd, cxt).E=\{e_{12}, e_{13}, \dots, e_{mn}\}$ 为事件驱动关联集, 有向边 $e_{ij}=(v_i, v_j)$ 表示事件 v_i 的发生驱动事件 v_j 的发生的次数, 即事件 v_j 发生时, 相同情境的上一个发生的事件为事件 v_i 的次数.

根据服务领域与情境信息的种类不同, 事件驱动图可以被定义成如下 4 类, 如图 1 所示.

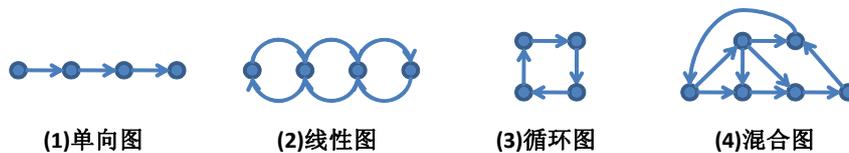


Fig.1 Classification of event driven graph

图 1 事件驱动图分类

其中,

- (1) 单向图中, 各事件存在偏序关系, 前序事件驱动唯一的后序事件, 该过程不可逆;
- (2) 线形图可以看作是单向图的一种扩展, 即单向图中各后序事件也可以逆向驱动前序事件发生;
- (3) 循环图可以看作是单向图的一种扩展, 即单向图中的结束事件可以驱动开始事件发生;
- (4) 混合图中, 各事件间存在任意的驱动关联关系.

事件驱动图表示同一情境信息不同事件之间的驱动关系, 由于情境信息变化通常受自然环境或人为因素的影响, 其原因较为复杂, 本文只探讨具有直接关系的事件驱动, 即两事件在事件驱动图之间存在边相连. 事件 x_1 发生将驱动事件 x_2 发生的概率 $p(x_1, x_2)$ 的计算公式如下:

$$p(x_1, x_2) = \frac{e(v_1, v_2)}{num(v_1)}, v_1 = x_1, v_2 = x_2 \quad (1)$$

其中, v_1 和 v_2 分别对应事件 x_1 和 x_2 在事件驱动图中的节点, $num(v_1)$ 为顶点 v_1 发生的次数.

在实际的主动服务过程中, 随着用户的活动以及自然环境的变化, 情境信息的各种事件不断发生, 事件驱动

图的内容需要实时进行更新,事件驱动图的更新算法伪代码如下所示:

算法 1. Event driven graph update.

Input: Old event driven graph EDG , new event e ;

Output: New event driven graph EDG' .

METHOD:

New edg $EDG'=EDG$

New event $laste=getLastEvent()$

If e in $EDG'.V$

If $\langle laste,e \rangle$ in $EDG'.E$

$EDG'.E.\langle laste,e \rangle=EDG'.E.\langle laste,e \rangle+1$

Else

Add $\langle laste,e \rangle=1$ into $EDG'.E$

End if

Else

Add e into $EDG'.V$

Add $\langle laste,e \rangle=1$ into $EDG'.E$

End if

Return EDG'

在实际服务中,通常各服务领域的情境事件及事件驱动关联关系根据服务需求预先可知,即事件驱动图的顶点和边可预定义,只是边的事件驱动关联次数随着服务过程中的事件发生而增加.随着服务过程的持续,虽然各边的事件驱动关联次数仍会不断增加,但顶点各边之间的事件驱动关联比例相对趋于稳定.由于情境事件不断发生,事件驱动图的更新占用大量系统资源,因此,当事件驱动图趋于稳定后,可不必持续更新,以提高系统效率.

假设在服务领域 sd 中存在 m 种情境信息 $(cxt_1, cxt_2, \dots, cxt_m)$, 系统感知用户当前所处的服务环境 $environment_1 = \{event_{1a}, event_{2b}, \dots, event_{mn}\}$, 设概率阈值为 θ , 通过事件驱动图, 可分析出用户可能处于的概率大于 θ 的各种服务环境.

服务环境概率与其组成情境事件概率相关, 设服务环境为 $environment = \{event_1, event_2, \dots, event_n\}$, 各情境事件的概率为 $p(event)$, 则服务环境概率 $p(environment)$ 的计算公式为

$$p(environment) = \prod_{i=1}^n p(event_i) \quad (2)$$

具体分析算法过程描述如下:

算法 2. 服务环境分析.

Input: 服务领域 sd 各情境信息的事件驱动图集合 EDG , 当前服务环境 $environment$, 概率阈值 θ ;

Output: 满足概率阈值的的服务环境集合 $ENVIRONMENT$.

METHOD:

- 1 初始化 $ENVIRONMENT$, 将 $environment$ 加入 $ENVIRONMENT$;
- 2 对 $environment$ 中的情境事件 $event_{ij}$, 在 EDG 中情境信息 cxt_i 对应的驱动图 edg_i 中查找事件 $event_{ij}$ 对应的顶点和以该顶点为初始顶点的各条边, 根据公式(1)计算并找出由事件 $event_{ij}$ 驱动的概率 $p(event_{ij}, event_{ix})$ 大于 θ 的所有事件, 记录各个事件及其概率, 组成概率事件集 $EVENT_i$, 将事件 $event_{ij}$ 加入到 $EVENT_i$ 中, 其概率为 100%;
- 3 遍历 $environment$ 中的各个情境事件, 得到各个情境信息的概率事件集 $EVENT_1, EVENT_2, \dots, EVENT_m$;
- 4 对各情境信息的概率事件集进行排序, 根据公式(2)计算过滤概率大于 θ 的、通过各情境信息概率事件

集的笛卡尔积得到的服务环境及其概率集合 $ENVIRONMENT=EVENT_1 \times EVENT_2 \times \dots \times EVENT_m$.

2.2 服务查找

根据服务需求的定义,用户需求的服务与感知的情境事件之间存在关联关系.因此,为进行主动服务发现,可以从宏观角度,对大量需求进行分析,找出其关联规律,然后根据该规律和已知发生在用户身上的情境事件,发现其可能需求的服务.该过程的核心是对用户需求的服务与情境事件的关联关系进行表示和抽取.本文借鉴数据挖掘中的 FP-TREE^[13],将二者关系定义为事件-服务 FP-TREE,具体如下所述.

定义 7(事件-服务 FP-TREE). 事件-服务 FP-TREE 是满足下列条件的一个树结构:它由一个根节点、事件前缀项(树干)、服务叶子节点和一个频繁项头表组成,其中,

- (1) 根节点为 null;
- (2) 事件前缀项是一个三元组 $event_node=(event_name,event_count,event_link)$,其中:
 - a) $event_name$ 记录节点表示的事件的标识;
 - b) $event_count$ 记录到达该节点的子路径次数;
 - c) $event_link$ 用于连接树中的相同事件标识的下一个事件节点,如果不存在下一个节点,则为 null;
- (3) 服务叶子节点是一个三元组 $service_node=(service_name,service_count,service_link)$,其中:
 - a) $service_name$ 记录节点表示的服务的标识;
 - b) $service_count$ 记录到达该节点的子路径次数;
 - c) $service_link$ 用于连接树中的相同服务标识的下一个服务节点,如果不存在下一个节点,则为 null;
- (4) 频繁项头表保存各事件、服务的标识和指向第 1 个该事件、服务标识的树节点指针.

事件-服务 FP-TREE 的生成类似于传统 FP-TREE,对大量已知的情境事件关联的服务需求进行频繁项集查找^[13],算法过程描述如下:

算法 3. 事件-服务 FP-TREE 生成.

Input: 情境事件关联的服务需求集 R ,支持度阈值 λ ;

Output: 事件-服务 FP-TREE.

METHOD:

- 1 扫描原始需求集 R ,得到频繁事件集 F 及各事件的支持度,对 F 中的事件按支持度 λ 进行降序排序,根据过滤掉支持度低于 λ 的事件,得到排序 L ;
- 2 创建事件-服务 FP-TREE 的根节点 $root$,以 null 标记;
- 3 再次扫描原始需求集 R ,对 R 中的每条需求 r ,将其事件按照 L 进行排序和过滤,然后将服务放在序列末尾,设排序后的序列为 $[p|P]$.其中, p 是第 1 项, P 是剩余项.调用 $insert_tree([p|P],root)$;
- 4 $insert_tree([p|P],root)$ 的执行过程如下:
 - a) 如果 $root$ 有子女 $child$ 使 $child.name=p.name$,则 $child.count$ 加 1;
 - b) 否则创建 $root$ 的一个子女 $child,child.name=p,child.count=1$,链接 $child$ 到它的父节点 $root$;在频繁项头表中查找名称为 p 的频繁项,依次找到 p 指针的末尾,将其指向 $child$;
 - c) 如果 P 非空,递归地调用 $insert_tree(P,child)$.

表 1 给出一个服务领域的原始服务需求集及其过滤排序后的结果,该服务领域包含 6 个情境信息,设支持度为 3.

根据表 1,各事件的频繁度排序为 $\{e_{52}:5,e_{11}:4,e_{43}:4,e_{22}:3,e_{61}:3\}$.建立的事件-服务 FP-TREE 如图 2 所示.

Table 1 List of service requirement

表 1 服务需求集

序号	原始需求集	整理后的需求集
1	$e_{11}, e_{22}, e_{31}, e_{43}, e_{52}, e_{61}, s_1$	$e_{52}, e_{11}, e_{43}, e_{22}, e_{61}, s_1$
2	$e_{11}, e_{21}, e_{32}, e_{42}, e_{52}, e_{64}, s_2$	e_{52}, e_{11}, s_2
3	$e_{11}, e_{22}, e_{34}, e_{41}, e_{52}, e_{62}, s_3$	$e_{52}, e_{11}, e_{22}, s_3$
4	$e_{11}, e_{23}, e_{36}, e_{43}, e_{52}, e_{63}, s_4$	$e_{52}, e_{11}, e_{43}, s_4$
5	$e_{12}, e_{24}, e_{33}, e_{43}, e_{51}, e_{61}, s_2$	e_{43}, e_{61}, s_2
6	$e_{13}, e_{22}, e_{35}, e_{43}, e_{52}, e_{61}, s_3$	$e_{52}, e_{43}, e_{22}, e_{61}, s_3$

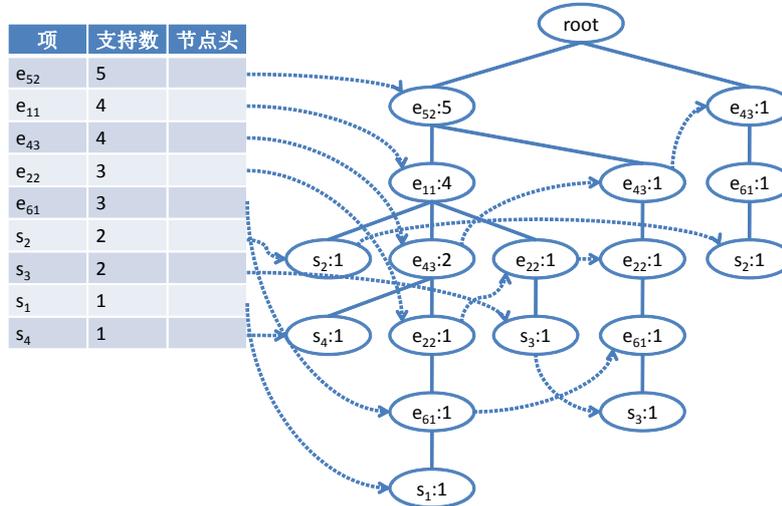


Fig.2 Event-Service FP-TREE

图 2 事件-服务 FP-TREE

对于系统感知用户当前所处的服务环境 $environment_1 = \{event_{1a}, event_{2b}, \dots, event_{mn}\}$, 通过该环境中的情境事件查找事件-服务 FP-TREE, 可以找出当前服务环境下用户可能需求的服务. 其基本思想是根据事件-服务 FP-TREE 的频繁项头表中的频繁度对服务环境的各个情境事件进行排序过滤, 然后根据排序过滤后的结果在事件-服务 FP-TREE 中找到相应的分支, 最后统计该分支所包含的服务叶子节点及其频繁数. 每个服务的需求可能性 $p(environment_1, s)$ 的计算公式为

$$p(environment_1, s) = \frac{\sum f(s)}{f(event_{ij})} \tag{3}$$

其中, $\sum f(s)$ 是该分支中所有项名为 s 的叶子节点的频繁数之和, $f(event_{ij})$ 是该分支最末端项名为 $event_{ij}$ 的节点的频繁数.

服务查找算法过程描述如下:

算法 4. 服务查找.

Input: 服务环境 $environment = \{event_{1a}, event_{2b}, \dots, event_{mn}\}$, 事件-服务 FP-TREE $fp-tree$;

Output: 当前服务环境下用户可能需求的服务集合 $SERVICE$.

METHOD:

- 1 初始化 $SERVICE$;
- 2 将 $environment$ 中的情境事件按照 $fp-tree$ 中频繁项头表的频繁度进行降序排序, 过滤掉频繁项头表中不存在的事件, 得到排序后的序列为 $[p|P]$, 其中, p 是第 1 项, P 是剩余项. 调用 $find_service([p|P], fp-tree.root)$;
- 3 $find_service([p|P], root)$ 的执行过程如下:

- a) 如果 P 非空且 $root$ 有子女 $child$ 使 $child.name=p.name$, 则递归地调用 $find_service(P, child)$;
- b) 遍历 $fp-tree$ 中以节点 p 为根的子树, 找到所有服务叶子节点插入 $SERVICE$, 并根据公式(3)计算其可能性.

2.3 主动服务发现算法

主动服务发现根据系统感知用户当前所处的服务环境 $environment_1 = \{event_{1a}, event_{1b}, \dots, event_{1m}\}$, 通过服务环境分析得出用户可能处于的服务环境及其概率, 再对每个可能的服务环境进行服务查找, 找到各个服务环境下用户需求的服务及其可能性, 然后将找到的服务进行汇总, 计算其优先级, 并按照优先级进行排序过滤, 得到最终的发现结果. 服务优先级 $priority(s)$ 的计算函数为

$$priority(s) = \sum_{i=1}^n p(environment_i, s) \times p(environment_i) \quad (4)$$

其中, $p(environment_i, s)$ 是用户可能处于的各服务环境中该服务被需求的可能性, $p(environment_i)$ 是该服务环境的概率.

主动服务发现算法过程描述如下:

算法 5. 主动服务发现.

Input: 服务环境历史记录 $ENVIRONMENT_H$, 已知的服务需求集 R , 用户当前服务环境 $environment$, 支持度阈值 λ , 服务环境概率阈值 θ ;

Output: 当前服务环境下用户可能需求的服务集合 $SERVICE$.

METHOD:

- 1 初始化 $SERVICE$;
- 2 通过算法 1, 根据服务环境历史记录 $ENVIRONMENT_H$ 构造事件驱动图集合 EDG ;
- 3 通过算法 2, 根据服务需求集 R 和支持度阈值 λ 构造事件-服务 $FP-TREE$ $fp-tree$;
- 4 通过算法 3, 根据事件驱动图集合 EDG 和服务环境概率阈值 θ 对用户当前服务环境 $environment$ 进行服务环境分析, 得到可能的服务环境集合 $ENVIRONMENT$;
- 5 通过算法 4, 在事件-服务 $FP-TREE$ $fp-tree$ 上对服务环境集合 $ENVIRONMENT$ 进行服务查找, 得到 $ENVIRONMENT$ 中各服务环境下用户需求的服务集合 $SERVICE_i$;
- 6 对各个服务集合进行合并, 通过公式(4)计算各个服务的优先级, 并按照优先级进行排序, 返回排序后的结果.

2.4 算法复杂度分析

主动服务发现算法的时间复杂度的计算包含 4 个部分: 根据服务环境历史记录构造事件驱动图集合; 根据服务需求集和支持度阈值构造事件-服务 $FP-TREE$; 然后, 根据当前服务环境和事件驱动图集合进行服务环境分析, 得到用户可能的服务环境集合; 最后, 根据服务环境分析的结果和事件-服务 $FP-TREE$, 进行服务查找与合并.

假设服务环境历史记录 $ENVIRONMENT_H$ 中的情境信息种类为 c , 每种情境信息的平均历史事件数目为 e , 这些事件共构成了 r 个服务需求, 则事件驱动图生成算法的事件复杂度为 $O(c \times e)$; 事件-服务 $FP-TREE$ 的生成算法的时间复杂度为 $O(r)$; 服务环境分析的时间复杂度受概率阈值 θ 影响, 当 $\theta=1$ 时, 环境分析时间复杂度取最小值 $O(c)$, 当 $\theta=0$ 时, 环境分析时间复杂度取最大值 $O(c \times e)$; 服务查找与合并的时间复杂度与服务环境分析后得出的服务环境个数有关, 当 $\theta=1$ 时, 环境分析得出的服务环境个数取最小值 1, 当 $\theta=0$ 时, 环境分析得出的服务环境个数取最大值 r , 该步骤对应的的时间复杂度分别为 $O(c)$ 和 $O(c \times r)$. 综上所述, 主动服务发现算法的时间复杂度受概率阈值的影响, 其最小值为 $O(c \times e) + O(r) + O(c) + O(c) = O(c \times e + r + 2c)$, 其最大值为 $O(c \times e) + O(r) + O(c \times e) + O(c \times r) = O(2c \times e + r + c \times r)$.

3 案例分析

3.1 案例背景

本文以机场服务作为案例服务领域.乘客在机场需要遵照一定流程乘机,如办理乘机手续、安检、登机,乘客需要相应的查询服务,如登机时间查询、登机口查询等.由于办理手续可能需要排队等候,为防止误机,乘客通常会提前到达机场,并且很可能需要在机场等待一段时间,在这段时间内,乘客可能需要就餐、购物、娱乐或者其他服务.此外,当乘客专注于某些事情,如购物时,可能由于忘记时间而耽误乘机,因此还需要相应的提醒服务,如登机提醒等.

目前,上述服务的实现方式主要通过广播方式进行,机场通过设置电子屏、告示板或者语音广播向所有乘客提供.一方面,这种提供方式到达率较差,除语音广播外,电子屏、告示板等需要乘客走到相应位置才能看到,对机场环境或乘机流程不熟悉的乘客帮助甚微;另一方面,这种方式缺乏时效针对性,乘客的用户体验难以提升.

基于情境感知的主动服务系统可以帮助机场提升其服务质量.机场可通过部署传感器通过用户手机感知其位置和移动速度,还可从其应用系统中获取乘客的相关信息,设定相关情境.例如,可根据当前时间和乘客航班的起飞时间设定乘客当前剩余时间,根据乘客当前位置和航班登机口设定乘客是否在正确的候机位置,以及乘客目前的乘机流程状态等.本文案例中待选的服务、情境信息及其相关情境事件见表 2.

Table 2 List of service, context information and context events in airport service

表 2 机场服务中的服务、情境信息及情境事件

服务	情境信息				
	位置	剩余时间	乘机状态	候机位置状态	乘客类型
办理乘机手续提醒	机场大厅	登机($t < 15\text{min}$)	登机	正确	普通乘客
安检提醒	柜台	即将登机($15\text{m} < t < 30\text{m}$)	候机	错误	银卡乘客
登机提醒	安检处	候机($30\text{m} < t < 1\text{h}$)	已办理乘机手续		金卡乘客
娱乐推荐	候机通道 1	等待($1\text{h} < t < 2\text{h}$)	未办理乘机手续		
书吧推荐	候机通道 2	休息($2\text{h} < t$)			
购物推荐	候机厅 1				
就餐推荐	候机厅 2				
	商场				
	书店				
	娱乐区				
	餐厅				

根据表 2 中的情境信息及其情境事件,通过算法 1 建立各情境信息的事件驱动图,其中,情境信息-位置的事件驱动图如图 3 所示.

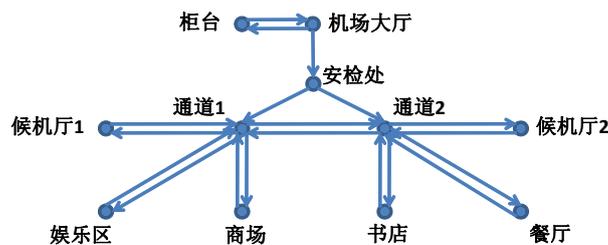


Fig.3 Event driven graph for context-location

图 3 情境信息-位置的事件驱动图

3.2 实验环境及方法

本文通过 SQL Server 2005 作为情境事件和服务需求的存储数据库,通过存储过程实现上述算法.为分析数据集大小对算法效果的影响,以及对比基于情境感知事件的主动服务发现方法与广播式主动服务方法的效果,

在上述案例的背景下,本文设计了两组实验.

实验 1 用以分析情境事件历史数据集大小对算法效果的影响.采用 3 组不同大小的情境事件历史数据集生成事件驱动图和事件-服务 FP-TREE,然后测试 3 种情况下基于情境感知事件的主动服务发现方法的查全率和查准率.实验首先通过随机方式产生 3 组(5,10,20)具有一定实际意义的情境事件集,然后通过调研 5 位用户在上述情境事件历史数据集中的服务需求,分别形成 3 组实验数据(5×5=25,10×5=50,20×5=100)作为历史数据,在该历史数据的基础上分别建立事件驱动图和事件-服务 FP-TREE.然后再随机生成 10 个具有一定实际意义的情境事件集,与 10 位用户(包括生成历史数据集的 5 位用户)形成 10×10=100 个测试服务环境,用以进行服务发现;并调研对应用户的服务需求,计算查全率和查准率.

实验 2 用以对比基于情境感知事件的主动服务发现方法(方法 1)与广播式主动服务方法(方法 2)的服务效果.实验 2 采用实验 1 中的历史数据集 3(20×5=100)、测试服务环境和调研的服务需求作为测试数据.对方法 1,按照上述方法计算查全率和查准率.对方法 2,在历史数据集中查找与一个测试服务环境里的各个情境事件相关的服务,对所有 10 位用户进行广播,然后与调研的服务需求进行对比,计算其查全率和查准率.

3.3 实验结果及分析

在本文中,服务查准率 $r(\text{precision})$ 表示系统主动提供的服务为用户所需服务的比例,其计算函数为

$$r(\text{precision}) = \frac{\sum \text{每次提供的服务中为顾客所需的服务数}}{\sum \text{每次提供的服务数}} \quad (5)$$

服务查全率 $r(\text{recall})$ 表示系统主动提供的服务对用户所需服务涵盖的比例,其计算函数为

$$r(\text{recall}) = \frac{\sum \text{每次提供的服务中为顾客所需的服务数}}{\sum \text{顾客所需服务数}} \quad (6)$$

实验 1 的结果如图 4 所示.

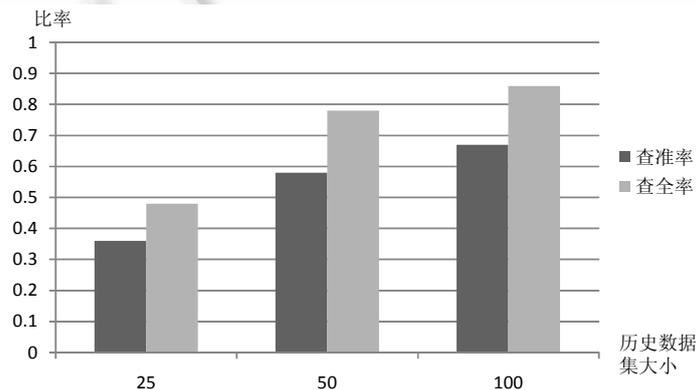


Fig.4 Result of test 1

图 4 实验 1 的结果

通过图 4 可以看出,随着历史数据集数据量的增加,基于情境感知事件的主动服务发现方法的查准率和查全率均有所提高.这是由于该方法中的事件驱动图和事件-服务 FP-TREE 需要根据历史数据集建立,随着历史数据集数据量的增加,二者所蕴含的对用户未来事件的预测和事件与服务的相关更加准确,更能客观地反映现实.因此,查准率和查全率会有所提高.但随着数据集数量的增加,查准率和查全率的提高趋势渐缓.

实验 2 的结果如图 5 所示.

通过图 5 可以看出,广播式主动服务的查准率很低,这是因为广播式服务提供给所有用户,而并未针对不同事件下的用户而有所区分,因此导致用户收到大量与己无关的服务推荐信息.基于情境感知事件的主动服务通过感知的情境事件分析用户当前及未来可能所处的服务环境,进而利用服务环境与所需提供服务的关联为用户有针对性地选择服务,因而其查准率较广播式有显著提升.

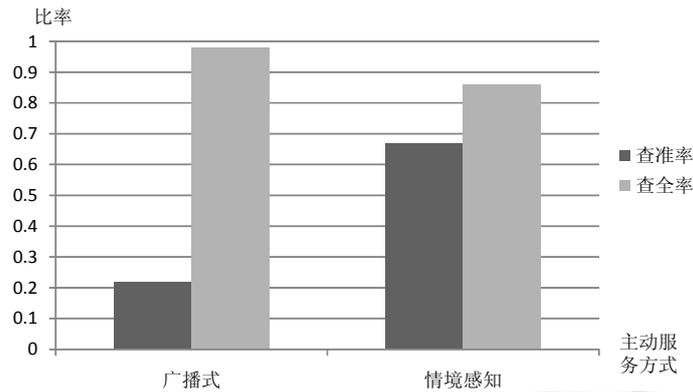


Fig.5 Result of test 2

图 5 实验 2 的结果

但是,基于情境感知事件的主动服务在服务查全率方面不如广播式主动服务,这主要包括以下 3 个方面的原因.第一,服务的查找依赖于根据历史数据建立的事件-服务 FP-TREE.如果该用户所需服务在历史数据中出现次数较少,则可能受阈值影响而未出现在事件-服务 FP-TREE 的分支中,因此导致该服务无法被查到;第二,服务查找依赖于用户当前服务环境中的情境事件.情境事件(或事件组合)在历史数据中出现的频率越低,在事件-服务 FP-TREE 代表该服务环境的情境事件就越少,进而导致服务查找的结果失真,无法找到恰当的服务;第三,由于用户自身的差异.不同用户在相同的 service 环境中的服务需求可能不同,甚至即便是同一用户也可能出现这种情况,此时也会导致查找的服务与用户的需求不符.而在本文实验 2 中的广播式服务则是将所有与当前情境事件相关的服务都向所有用户推荐,因而具有很高的服务查全率.

4 结束语

本文提出了一种基于情境感知事件的主动服务发现方法,该方法依赖于基于服务历史记录建立的表达事件之间关联关系的事件驱动图和表达事件与服务之间关联关系的事件-服务 FP-TREE,通过感知的情境事件及事件驱动图推测用户当前及未来可能发生的情境事件,根据这些情境事件查找事件-服务 FP-TREE,找到用户可能需要的服务.通过案例分析,该方法较目前普遍存在的广播式主动服务具有更高的服务查准率,有助于提升服务主动提供的针对性.

接下来的进一步工作包括:研究事件驱动图中时间因素对驱动关联的影响;继续完善事件-服务 FP-TREE 的建立,使之能够根据新的服务记录进行更新;结合领域本体与用户特征本体,考虑用户自身特征对主动服务发现的影响.

致谢 在此,我们向对本文工作给予帮助和建议的哈尔滨工业大学徐晓飞教授表示感谢.

References:

- [1] Mo T, Li WP, Chu WJ, Wu ZH. CABS3: Context-Awareness based smart service system. In: Proc. of the 6th Int'l Conf. on Wireless Communications, Networking & Mobile Computing. Chengdu, 2010. 23-25.
- [2] Staunstrup J, Tung F, Yu L, *et al.* Services in context. Computer Systems and Applications, 2009,6:161-167.
- [3] Mo T, Li WP, Chu WJ, Wu ZH. Framework of context-aware based service system. Chinese Journal of Computers, 2010,(11): 2084-2092 (in Chinese with English abstract).
- [4] Zhang LJ, Zhou Q, Chao T. A dynamic services discovery framework for traversing Web services representation chain. In: Proc. of the Int'l Conf. on Web Services (ICWS 2004). 2004.
- [5] Eyhab AM, Qusay HM. Investigating Web services on the World Wide Web. In: Proc. of the Int'l Conf. on World Wide Web. 2008.

- [6] Dong X, Halevy A, Madhavan J, *et al.* Similarity search for Web services. In: Proc. of the Int'l Conf. on Very Large Data Bases. 2004.
- [7] Yau SS, Liu JW. Service functionality indexing and matching for service-based systems. In: Proc. of the 2008 IEEE Int'l Conf. on Services Computing (SCC 2008). 2008.
- [8] Zhou C, Chia L, Silverajan L, *et al.* UX—An architecture providing QoS-aware and federated support for UDDI. In: Proc. of the Int'l Conf. on Web Services (ICWS 2003). 2003. 171–176.
- [9] Vu LH, Hauswirth M, Porto F, *et al.* A search engine for QoS-enabled discovery of semantic Web services. Int'l Journal on Business Process Integration and Management (Special Issue), 2006,1(4):244–255.
- [10] Bellur U, Vadodaria H. On extending semantic matchmaking to include preconditions and effects. In: Proc. of the IEEE Int'l Conf. on Web Services (ICWS 2008). 2008.
- [11] McGuinness DL, Fikes R, Hendler J, Stein LA. DAML+OIL: An ontology language for the semantic Web. Intelligent Systems, 2002,17(5):72–80.
- [12] Akkiraju A, Goodwin R, Doshi P, Roeder S. A method for semantically enhancing the service discovery capabilities of UDDI. Technical Report, IBM T.J. Watson Research Center, 2004.
- [13] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation. In: Proc. of the 2000 ACM-SIGMOD Int'l Conf. on Management of Data. Dallas: ACM Press, 2000. 1–12.

附中文参考文献:

- [3] 莫同,李伟平,褚伟杰,吴中海.一种情境感知服务框架.计算机学报,2010,(11):2084–2092.



莫同(1981—),男,辽宁沈阳人,博士,主要研究领域为服务计算,物联网,数据挖掘.



吴中海(1968—),男,博士,教授,博士生导师,CCF高级会员,主要研究领域为情境感知服务,软件服务工程,嵌入式软件与系统.



褚伟杰(1980—),男,讲师,主要研究领域为软件工程.



林慧苹(1974—),女,副教授,主要研究领域为情境感知服务,SOA,云计算,中间件技术,企业业务流程管理与优化.



李伟平(1973—),男,博士,教授,主要研究领域为软件工程,服务计算,情境感知服务.