

# 一种基于情景感知的学习服务发现算法

朱郑州

(北京大学软件与微电子学院 北京 100871)

**摘要** 针对学习者对 e-Learning 服务发现系统提供的学习服务不满意或者满意程度不稳定的问题,引入了情景因子,设计了一个学习服务发现算法——eLSDACA。该算法感知学习者的学习情景,形成学习者情景本体,参与服务发现的过程。情景因子有两个作用:一是作为输入和学习者输入同时输入学习服务发现系统;二是作为学习服务发现系统的修正因素,利用修正函数对已发现的发布服务的综合匹配度进行更新,并给出相应的输出。仿真实验表明,在学习者进行学习服务发现的次数超过 10 时,该算法的查全率和查准率均达到 90% 以上,优于传统的学习服务发现算法。

**关键词** 学习服务,领域本体,服务发现算法,情景感知,修正函数

**中图分类号** TP393 **文献标识码** A

## E-Learning Services Discovery Algorithm Based on Context Aware

ZHU Zheng-zhou

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

**Abstract** Based on e-Learning service discovery system, an e-Learning service discovery algorithm-eLSDACA was proposed. The algorithm awares the learning context of learners, and then comes into students context ontology. E-Learning context element is used in 2 aspects. The first is that it is used as input properties. The second is that it is the modification element for the e-Learning service discovery system. With the help of the amendatory function, the total matching degree of discovered advertised e-Learning services is modified. Simulation experiments indicate that this algorithm's precision and recall of e-Learning service discovery are all over 90%, and they are all better than the traditional algorithms, after times that learner uses the system to discovery e-Learning service are more than 10.

**Keywords** E-Learning services, Domain ontology, Service discovery algorithm, Context aware, Amendatory function

## 1 引言

随着互联网、移动设备和无线传感网的发展,参加远程学习的学习者规模大大增加。学习者群体的个性特征丰富多样,学习环境不断变化,他们的远程学习方式也由单纯的课件点播发展到个性化自主学习、协同学习。在这种新的学习环境下,提高学习的质量和效率的关键是快速准确地发现学习者所需的学习服务。在现有的学习服务发现系统中,根据学习者的请求能够找到一些匹配度很高的学习服务,但是学习者并不一定满意,这其中有优质服务数量有限、服务请求者对所需服务的描述不准确等原因,更深层次的原因是系统所采用的服务发现算法不合理。通过分析我们发现,现有的算法<sup>[1,2]</sup>都是只把服务自身的属性匹配度作为衡量服务匹配效果优劣的唯一指标,并没有考虑到学习者在远程学习时所处的情景。这就不能完全把握学习者对学习服务的真实需求,从而降低了学习者对学习服务发现系统的满意度。本文提出了一种基于情景感知的学习服务发现算法——eLSDACA,该算法把学习者的情景信息一部分作为输入信息,和学习者输入的请求信息一并作为服务发现的请求信息;另一部分作为反馈

信息,根据设定的修正函数,动态更新已发现的发布服务的各个属性的匹配度权值,同时更新学习者的情景本体。实验表明,本算法既能提高学习服务发现的查准率和查全率,又能提高学习者对学习服务发现系统的满意度。

本文第 2 节介绍学习服务以及学习服务的描述模型;第 3 节介绍基于情景感知的学习服务发现算法;第 4 节介绍该算法的学习情景更新部分的实现;第 5 节介绍实验环境以及对实验结果的分析;最后是总结与展望。

## 2 学习服务

学习服务即为在 e-Learning 过程中所用到的各种学习工具和学习软件,如协同编辑软件、邮件服务工具和监控软件等。本文通过定义 1 描述一个学习服务。

**定义 1** 一个学习服务描述模型可以定义为四元组  $eLS: = \{ INPUT_s, OUTPUT_s, QoS, CA \}$ 。其中,  $INPUT_s$  是服务的输入信息,它是学习服务的输入属性的集合;  $OUTPUT_s$  是学习服务的输出属性的集合,  $INPUT_s$  和  $OUTPUT_s$  集合中的元素都是领域本体中的概念;  $QoS$  是学习服务的服务质量(Quality of Service, QoS),描述了学习服务满足消费

者需求的能力,包含内容很多。出于实用性和可测量性的考虑,本文定义一个学习服务的 QoS,如定义 3。

目前,很多的基于语义的服务发现方法都采用领域本体作为语义基础。用于服务发现的领域本体描述了领域内的概念和概念之间的语义关系,服务匹配要计算领域概念之间的语义关系和相似程度,从而给出服务的匹配程度。Su 和 Lo 将语义关系归纳为 9 种<sup>[3]</sup>。因为除类属关系和特征关系外的其它 7 种关系与学习服务发现这个主题无关,且学习服务概念间还存在同义词和缩写,所以本文用于服务发现的领域本体中的概念关系限定为同义关系、类属关系和特征关系。在 e-Learning 领域,学习服务本体就是 e-Learning 领域内的概念和概念之间的语义关系,如定义 2。

**定义 2** 一个学习服务本体(e-Learning Service Ontology)可以用二元组来表达, $O_i = (C, R)$ 。其中, $C$  为本体  $O_i$  范围内的元素集合; $R$  为元素之间的关系集合,取值为  $\{ER, PR, FR\}$ <sup>[4]</sup>。其中, $ER$  是同义词关系(Equivalent Relation),表示一个概念与另一个概念所表示的意义相同或者一个是另一个的缩写; $PR$  即类属关系(Partner Relation),表示一个概念与它的子类概念之间的关系; $FR$  即特征关系(Feature Relation),是指某一概念与其特征属性之间的关系。

按照定义 1,文献[5]中的图 2 给出了一个学习服务本体实例,它用领域概念之间的弧代表语义关系,把学习服务本体表示成一个网状的结构。

根据文献[6-9],本文提出了一个学习服务描述模型。

**定义 3** 一个学习服务质量模型(QoS)可以定义为一个三元组, $QoS_i = \{RT, CT, AL\}$ 。其中, $RT$  表示响应时间(Response Time),指学习者从提交服务请求到获得服务响应所花的时间; $CT$  是服务价格(Cost),指每次服务调用所需支付的费用; $AL$  是服务的可用性(Availability),指学习服务正常运行的概率。这 3 个 QoS 属性由服务提供者在服务发布时给出初始值,随后在系统控制下动态变化。

**定义 4** 可用性(Availability)定义为  $AL = s / (s + f)$ ,  $s$  表示成功执行的次数,  $f$  表示失败执行的次数。可用性的取值范围为  $[0, 1]$ <sup>[10]</sup>。

CA 是学习者的学习情景(e-Learning Context Aware, CA),详细介绍请参考本文第 3 节。

### 3 基于情景感知的学习服务发现算法

#### 3.1 情景感知

学习者在远程学习过程中,周围存在丰富的情景信息。为更好地进行学习服务发现,首先需要对学习者的学习情景进行有效刻画,并在此基础上为情景建模。这是基于情景感知的学习服务发现的前提和基础。

对于情景的定义,Dey 等人认为,可以用于刻画实体所处状态的任何信息<sup>[11]</sup> 都是情景;顾君忠则把情景分为计算情景、学习者情景、物理情景、时间情景和社会情景等 5 类<sup>[12]</sup>。本文在以上定义的基础上,结合学习服务发现的需要,对学习情景做如下定义。

**定义 5** 学习情景是指学习者在远程学习时所处的即时环境。一个学习情景模型可以表述为一个四元组, $CA_i = \{Profile, PerCha, LeaPre, TimSpa\}$ 。其中, $Profile$ 、 $PerCha$ 、 $LeaPre$ 、 $TimSpa$  分别表示基本信息、个性特征、学习偏

好、时空信息等 4 类学习情景信息,如表 1 所列。

表 1 学习情景分类

| 学习情景类别                    | 学习情景信息                 | 取值范围             |
|---------------------------|------------------------|------------------|
| 基本信息(Profile)             | 性别(Gender)             | 男/女              |
|                           | 年龄(Age)                | 正整数              |
| 个性特征(Personal Character)  | 性格维度(E/D)              | 外向型/内向型          |
|                           | 知觉维度(S/N)              | 理智型/情感型          |
|                           | 理性维度(T/F)              | 思考型/感觉型          |
|                           | 态度维度(J/P)              | 判断型/感知型          |
| 学习偏好(Learning Preference) | 资源偏好(Resource Type)    | 音频/视频/文字/动画/游戏   |
|                           | 学习模式(Learning Mode)    | 自主学习/协同学习        |
|                           | 交互模式(Interactive Mode) | 非实时交互/实时交互       |
| 时空信息(Time/Space)          | 时间(Time)               | 学习者发送请求服务的时间     |
|                           | 地点(Location)           | 学习者发送请求服务的 IP 地址 |

为便于研究动态情景对学习服务发现的影响,本文只选择了性别和年龄作为基本信息。根据梅厄布雷格指示器,把学习者的个性特征分为性格维度、知觉维度、理性维度和态度维度等 4 类<sup>[13]</sup>。从资源偏好、学习模式和交互模式等 3 个角度描述学习者的学习偏好。时空信息则包括学习者发送请求服务的时间和地点等两类信息。

#### 3.2 学习服务发现算法

该算法在现有服务发现算法<sup>[4]</sup>的基础上,引入了学习情景因子,让学习者的学习情景也参与到学习服务发现的过程中,并依据学习者的情景信息对发现的学习服务进行排序,选出匹配度权值最高的学习服务。

##### 算法 1 eLSDACA

输入:请求服务,发布服务库,学习者情景信息

输出:匹配服务,匹配度权值

- (1) 建立发布服务库;
- (2) 建立结果服务库;
- (3) 接受学习者输入;
- (4) 感知学习者情景信息;
- (5) 根据情景信息完善学习者的输入;
- (6) 匹配输入属性;
- (7) 匹配输出属性;
- (8) 匹配 QoS;
- (9) 将满足输入、输出和 QoS 条件的服务置入结果服务库;
- (10) 计算结果服务库中各发布服务的综合匹配度 TotalMatchDegree;
- (11) 根据学习者情景信息修正 TotalMatchDegree;
- (12) 根据 TotalMatchDegree 对结果服务库中各发布服务进行排序,并把排序后的结果返回给学习者;
- (13) 搜集学习者对返回服务的使用情况 UR;
- (14) 根据 UR 更新学习者情景本体库。

系统最后返回给学习者的综合匹配度 TotalMatchDegree 计算如下:

$$TotalMatchDegree = \xi_1 * Match\_INPUTs + \xi_2 * Match\_OUTPUTs + \xi_3 * Match\_QoS \quad (1)$$

式中, $0 \leq \xi_1, \xi_2, \xi_3 \leq 1$ ,且  $\xi_1 + \xi_2 + \xi_3 = 1$ ,下面详细介绍 Match\\_INPUTs, Match\\_OUTPUTs 和 Match\\_QoS。

#### 3.3 IO 匹配

学习服务的 IO 匹配就是将请求服务的各个 INPUTs 属性和发布服务的所有 INPUTs 属性进行匹配,将请求服务的

各个  $OUTPUT_s$  属性和发布服务的所有  $OUTPUT_s$  属性进行匹配。假如请求服务和发布服务各有  $n$  个  $OUTPUT_s$  属性, 则输出属性  $i(1 \leq i \leq n)$  的匹配度为  $Match\_output_i(out_{RS}, out_{AS})$ , 学习服务的输出匹配度为:

$$Match\_OUTPUT_s = \sum(v_i * Match\_output_i(out_{RS}, out_{AS})) \quad (2)$$

式中,  $1 < i < n, 0 \leq v_i \leq 1, \sum v_i = 1$ 。

假设请求服务和发布服务的输出属性集分别为  $P_{O_s}^{RS}$  和  $P_{O_s}^{AS}$ ,  $\forall out_{RS} \in P_{O_s}^{RS}, \forall out_{AS} \in P_{O_s}^{AS}$ , 它们的匹配度  $Match\_output_x(out_{RS}, out_{AS})$  按算法 2 计算<sup>[14]</sup>。

#### 算法 2 Match\_output<sub>x</sub>

输入: 请求服务的输出属性  $out_{RS}$ , 发布服务的输出属性  $out_{AS}$

输出: 匹配结果

if( $out_{RS} == out_{AS}$ )  $\vee$  ( $out_{RS}$  subclassOf  $out_{AS}$ )

return exact;

else if( $out_{RS}$  subsumes  $out_{AS}$ )

return plugIn;

else if( $out_{RS}$  subsumes  $out_{AS}$ )

return subsumes;

else if( $out_{RS} \cap out_{AS} \neq \emptyset$ )

return Intersection;

else return Fail;

同理, 根据算法 2, 可以计算出请求服务和发布服务  $m$  个  $INPUT_s$  属性的匹配度  $Match\_input_i$ , 然后求出  $INPUT_s$  属性的综合匹配度  $Match\_INPUT_s$ :

$$Match\_INPUT_s = \sum(u_i * Match\_input_i(in_{RS}, in_{AS})) \quad (3)$$

式中,  $1 < i < m, 0 \leq u_i \leq 1, \sum u_i = 1, in_{RS}$  和  $in_{AS}$  分别是请求服务和发布服务的一个输入属性。

### 3.4 QoS 匹配

请求服务和发布服务的 QoS 匹配度定义如下:

$$Match\_QoS = w_1 * Match\_QoS_{RT} + w_2 * Match\_QoS_{CT} + w_3 * Match\_QoS_{AL} \quad (4)$$

式中,  $0 \leq w_1, w_2, w_3 \leq 1$ , 且  $w_1 + w_2 + w_3 = 1$ ,  $Match\_QoS_{RT}$ 、 $Match\_QoS_{CT}$  和  $Match\_QoS_{AL}$  分别是服务响应时间、价格和可用性的匹配度, 其值域为  $\{-1, 0, 1\}$ ; 若被满足, 则其匹配度为 1; 若不满足, 则为 -1; 若该度量没有被请求服务或发布服务描述, 则为 0。

## 4 学习情景更新

定义 6(学习服务利用率, e-Learning Service Use Rate, UR) 学习者对学习服务的利用程度。任一学习服务利用率 UR 可以表示为:

$$UR = \frac{t_0}{t_A} \quad (5)$$

式中,  $t_0$  表示学习者使用该学习服务的时间,  $t_A$  表示其它学习者使用该学习服务的平均时间。

从定义 6 可以看出, UR 表示了该学习情景信息对某学习者发现学习服务的重要度。

根据学习者使用学习服务的信息, 更新学习者的学习情景本体库。这种更新仅限于参与学习服务发现的情景信息, 对学习服务发现没有任何帮助的情景信息不做任何改变。比如, 某学习者的交互模式情景信息为实时交互, 系统据此发现一个即时通信的学习服务给学习者, 而学习者选用了另一非实时交互服务, 则算法会弱化交互模式情景信息的重要性。

基于文献[15]中的用户满意度更新算法, 本文建立了学习情景的更新流程, 如图 1 所示。

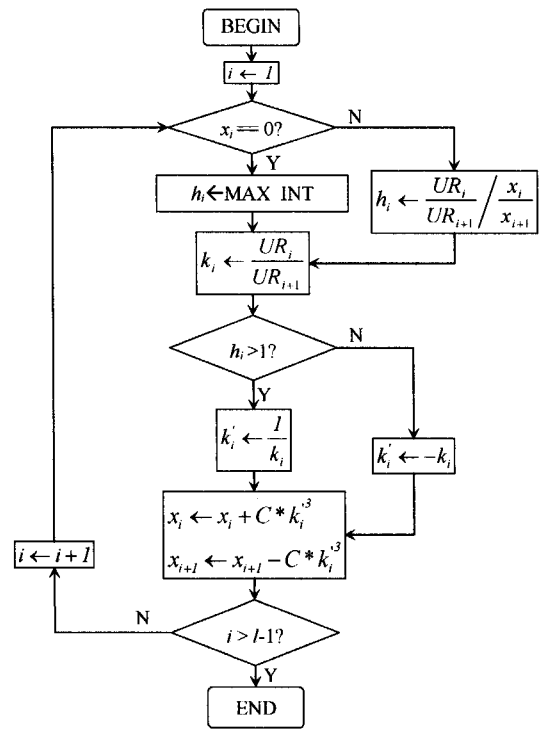


图 1 学习情景修正函数的工作流程

假设学习服务的某类情景信息(如 Profile, PerCha, LeaPre 或 TimSpa)的数目为  $l$ , 其学习服务利用率分别为  $x_1, x_2, \dots, x_l$ , 这些权值的初始值均为  $1/l$ , 学习者对该服务属性的使用度为  $UR_1, UR_2, \dots, UR_l$ , 则有

$$h_i = \begin{cases} \text{MAX\_INT}, & x_i = 0 \\ \frac{UR_i}{UR_{i+1}} \cdot \frac{x_i}{x_{i+1}}, & \text{otherwise} \end{cases}, i \in [1, l] \quad (6)$$

$$k_i = \frac{UR_i}{UR_{i+1}}, i \in [1, l] \quad (7)$$

$$k_i' = \begin{cases} \frac{1}{k_i}, & h_i > 1 \\ -k_i, & h_i < 1 \end{cases} \quad (8)$$

根据式(6)~式(8), 设计修订情景信息的匹配度权值的修正函数为:

$$x_i = x_i + C * k_i^3, 0 < C < 0.01 \quad (9)$$

$$x_{i+1} = x_{i+1} - C * k_i^3, 0 < C < 0.01 \quad (10)$$

修正函数的工作流程如图 1 所示。

根据修正函数分别对 Profile, PerCha, LeaPre, TimSpa 和 TotalMatchDegree 的学习服务利用率进行更新。

## 5 实验及分析

本文实验是在 P8600 @ 2.40GHz 处理器, 3G RAM, 250G 硬盘的联想计算机环境下展开的。共注册 15000 个发布服务, 分别在基于 UDDI 的匹配算法、基于 OWL-S 的匹配算法和 eLSDACA 算法 3 种环境下进行了实验。

### 5.1 系统体系结构

为了实现 eLSDACA 算法, 设计了一个学习服务发现系统, 如图 2 所示。

该系统使用了一个情景代理来处理学习者的情景信息。情景代理包括 3 个模块: 情景感知模块、情景评估模块、本体更新模块。情景感知模块用来搜集学习者的情景信息, 将搜集到的信息传递给情景评估模块。情景评估模块负责对收到的情景信息进行分类评估, 通过修正函数计算出新的匹配度

权值向量。服务更新模块根据 CA 评估器的计算结果,动态更新发布服务的属性匹配度权值。

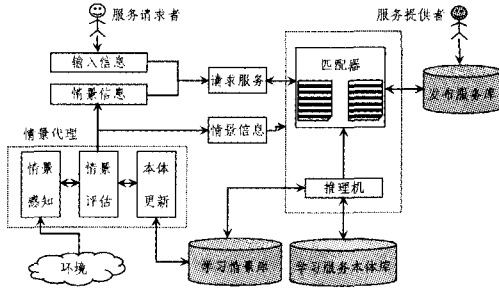


图2 学习服务发现体系结构

## 5.2 服务发现的准确度

为评估 eLSDACA 算法在学习服务发现中的准确度,本文注册了 15000 个学习服务,并且开发了 100 个学习者。在其进行学习服务发现时,随机赋予了一定的学习情景,进行了 4 组仿真实验,每组实验都分别使用 UDDI 发现算法、OWL-S 发现算法和 eLSDACA 发现算法进行服务发现。表 2 是采用 3 种算法进行服务发现的准确度对比。

表2 学习服务发现准确度评估表

| 发现服务的次数 | UDDI  |       | OWL-S |       | eLSDACA |       |
|---------|-------|-------|-------|-------|---------|-------|
|         | 查准率   | 查全率   | 查准率   | 查全率   | 查准率     | 查全率   |
| 1       | 53.6% | 48.1% | 80.3% | 87.8% | 85.9%   | 88.1% |
| 10      | 47.2% | 56.5% | 81.4% | 86.9% | 90.7%   | 91.9% |
| 50      | 49.3% | 48.7% | 79.7% | 87.1% | 91.2%   | 93.1% |
| 100     | 54.5% | 47.6% | 82.6% | 86.2% | 93.1%   | 94.7% |

从实验结果可以看出,在所有 4 组测试中,基于 UDDI 发现算法的查全率和查准率都不高,这是因为基于 UDDI 发现算法的匹配是基于关键词的静态匹配。无论是查准率还是查全率,基于 OWL-S 发现算法和 eLSDACA 发现算法都要比基于 UDDI 发现算法高出很多,这是因为前两个算法引入了基于本体的语义信息。

从这 4 组测试结果还可以看出,随着学习者发现服务次数的不断增多,基于 UDDI 发现算法和基于 OWL-S 发现算法的查准率和查全率并没有明显提高,甚至还有下降。而 eLSDACA 发现算法的查准率和查全率都随着发现服务次数的增加不断上升,最好时可以达到 90% 以上。这是因为 eLSDACA 发现算法引入了学习者的情景,在学习情景支持下,服务发现的过程更接近实际,从而提高了服务发现的查准率和查全率。

## 5.3 服务发现的效率

服务发现的效率是指服务发现的时间开销,也就是服务发现的速度。3 种匹配算法环境下的服务发现的时间开销如图 3 所示。

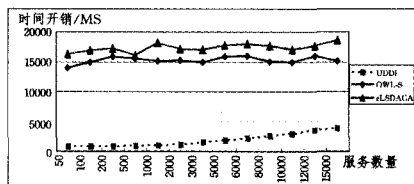


图3 学习服务匹配的开销

从图 3 可以看出,UDDI 发现算法的时间开销随着发布服务数量的增多平稳上升,但一直维持在比较低的水平,这是因为 UDDI 是基于关键字的静态匹配,匹配效率较高;而传统

的 OWL-S 发现算法和 eLSDACA 发现算法的时间开销一直维持中等水平,这是因为这两种服务发现算法要经过推理机进行推理。而总体看来,eLSDACA 发现算法的时间开销要高于传统的 OWL-S 算法,这是因为 eLSDACA 发现算法的匹配过程需要感知学习者的学习情景,并对感知到的学习情景进行相应的处理,故增加了部分时间开销。

**结束语** 针对网络环境下学习服务的特点,本文提出了一个基于情景感知的学习服务发现算法。该算法在学习者输入少量信息,甚至不输入信息的情况下,可根据学习情景发现满足学习者需求的学习服务。这种方法改进了学习服务匹配的综合匹配度的计算过程,提高了学习服务发现查准率,有助于学习者个性化自主学习,能提高学习者学习质量。尽管在学习服务发现的过程中增加了部分时间开销,使得效率有所下降,但其在学习者容忍范围之内。

学习情景本体的信息并不是很全,对学习情景信息的利用也不是很充分,今后可以从这两个角度进一步完善本算法,以达到更加实用的目的。

## 参考文献

- [1] Gao Shu, Rana O F, Avis N J. Ontology-based semantic match-making approach [J]. *Advances in Engineering Software*, 2007, 38(1):59-67
- [2] Klien E, Lutz M, Kuhn W. Ontology-based discovery of geographic information services-an application in disaster management [J]. *Computers, Environment and Urban Systems*, 2006, 30(1):102-123
- [3] 陆汝钫. 人工智能[M]. 北京, 科学出版社, 1996
- [4] 张正, 左春, 王裕国. 基于语义扩展的 Web 服务发现方法[J]. *通信学报*, 2007, 28(1):57-63
- [5] Vega-Gorgojo G, Miguel L. Bote-Lorenzo, Eduardo Góo-mez-Sánchez. A semantic approach to discovering learning services in grid-based collaborative systems[J]. *Future Generation Computer Systems*, 2006, 22(6):709-719
- [6] 吴健, 吴朝晖, 李莹, 等. 基于本体论和词汇语义相似度的 Web 服务发现[J]. *计算机学报*, 2005, 28(4):595-602
- [7] 胡建强, 邹鹏, 王怀民. Web 服务描述语言 QWSDL 和服务匹配模型研究[J]. *计算机学报*, 2005, 28(4):505-513
- [8] The OWL Services Coalition. OWL-S: semantic markup for Web services [EB/OL]. <http://www.w3.org/Submission/OWL-S/>, 2007-07-21
- [9] Selvi S T, Balachandar R A, Vijayakumar K. Semantic discovery of grid services using functionality based matchmaking algorithm[C]// *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence Web Intelligence*. Hong Kong: IEEE Press, 2006:170-173
- [10] Cardoso J, Sheth A, Miller J, et al. Quality of service for workflows and Web service processes[J]. *Journal of Web Semantics*, 2004, 1(3):281-308
- [11] Dey A K. Understanding and using context[J]. *Personal and Ubiquitous Computing*, 2001, 5(1): 4-7
- [12] 顾君忠. 情景感知计算[J]. *华东师范大学学报:自然科学版*, 2009(5):1-21
- [13] Jackson S L, Parker C P, Dipboye R L. A comparison of competing models underlying responses to the Myers-Briggs type indicator [J]. *Journal of Career Assessment*, 1996, 4(1):99-115

输入查询命令  $S_1$  和  $S_2$ , 结果分别得到 0 个 goto 语句和 3 个 continue 语句。还比如规则 15.3: switch 语句的最后子句应该是 default 子句。因为提取的信息粒度不够, 本文的工具不能判断一个语句是不是 switch 语句的最后一个子句, 但它可以找出不包含 default 子句的 switch 语句, 这样的 switch 语句同样是不满足规则 15.3 的。相应的查询命令  $S$  是:

```
find x; statement satisfying
  all y; statement x. specificity= 'switch'
  &&(! y isIn x || ! y. specificity= 'default')
```

依然以 NppTags 项目作为测试对象运行工具, 输入  $S$ , 结果返回 64 个不满足规则 15.3 的 switch 语句。

本节给出了本文工具适用的两个实际的应用场景, 在一定程度上验证了本文代码查询技术的研究意义以及工具的有效性。

## 7 相关工作

代码查询技术普遍采用了抽取-查询-展示 (extract-abstract-present)<sup>[1,11]</sup> 的模式来实现。它包括 3 个步骤, 即抽取: 扫描源代码, 将其映射到某种中间结构, 比如图; 查询: 对中间结构进行一些处理和查询操作, 以获得查询结果; 展示: 展示查询结果。在一种代码查询技术提出的同时, 通常会引入一个与领域相关的查询语言来描述查询的需求。

Holt<sup>[2,9]</sup> 在 20 世纪 90 年代实现了 Grok 工具, 并首先提出了基于 Tarski 关系代数<sup>[10]</sup> 的代码查询语言形式。它主要用来做基于 C++ 的软件架构分析, 所关注的程序中的信息粒度较大、信息量有限, 没有处理有关变量、语句层次的程序信息。

同样基于 Tarski 关系代数的还有 2007 年 Storm 发布的 JRelCal<sup>[12]</sup>, 它专门用作 Java 程序的外接 lib。JRelCal 本身不提供抽取过程的支持, 只是以 RSF 格式文件作为中间结构进行查询操作。

2007 年, Moor 等人提出了一种通用的面向对象的查询语言. QL<sup>[13]</sup>, 并基于此开发了 SemmlCode<sup>[3]</sup>——一个 Eclipse 平台的插件, 用于 Java 源码的分析。QL 的语言形式是基于 Codd 关系代数<sup>[14]</sup> 的。

本文提出的以程序元素为目标、面向 C/C++ 的查询技术是基于—阶谓词逻辑的查询语言形式的。相比于其他查询语言, 其语法结构相对简明易懂。同时, 它支持处理的程序信息覆盖面广、粒度较小, 包括变量、语句、方法、类的属性信息以及它们之间的各种关系信息。虽然目前处理的用户查询命令是一条语句, 但该语言对用户需要查询的目标仍具有相当可观的表述能力, 且语言本身还有很好的扩展性。

**结束语** 代码查询技术在程序分析和测试研究中有广泛的应用。本文提出并实现了一种基于程序分析的面向 C/C++ 的代码查询技术, 其能应用于代码审查、程序自动插桩等常用的软件工程工作场景。它通过代码静态分析获得程序元素信息并保存为中间结构, 作为代码查询过程的目标集合; 查询过程以程序元素为目标, 以谓词逻辑表达式的形式来描述查询

条件。基于此技术, 本文实现了一个基于 C/C++ 语言的原型工具 C2Parser。通过将其应用于几个实际场景, 验证了工具的有效性和研究意义。

下一步的研究工作将在以下几个方面展开: 第一, 扩充目标语言。只需要利用 JDT 实现对 Java 程序元素信息的收集, 就可以将本文的工作迁移到 Java 语言上来。第二, 扩充查询语言。可以引入更多种类的元素, 如表达式类型; 引入更丰富的元素属性和元素间关系; 支持输入语句更丰富的表达, 比如支持对查询结果的二次查询, 加入形如 find...in(find...) 的语法支持。

## 参考文献

- [1] Feijs L, Krikhaar R, van Ommerring R. A relational approach to support software architecture analysis[J]. *Software-Practice and Experience*, 1998, 28(4): 371-400
- [2] Holt R C. Structural manipulations of software architecture using Tarski relational algebra[C]//WCRE'98. IEEE Computer Society, 1998; 210
- [3] Verbaere M, Hajjiev E, de Moor O. Improve software quality with SemmlCode: an Eclipse plugin for semantic code search [C]//OOPSLA'07. New York, NY, USA: ACM, 2007; 880-881
- [4] Fagan M. Design and code inspection to reduce errors in program development[J]. *IBM Systems Journal*, 1976, 15(3): 182-211
- [5] Gilb T, Graham D. *Software Inspection*[M]. AddisonWesley, 1993
- [6] Marin M, Moonen L, van Deursen A. Soquet: Query-based documentation of crosscutting concerns[C]//ICSE'07. IEEE, 2007; 758-761
- [7] Huang J C. Program Instrumentation and Software Testing[J]. *Computer*, 1978, 11: 25-32
- [8] Chen S K, Fuchs W K, Chung J Y. Reversible debugging using program instrumentation[J]. *IEEE Transactions on Software Engineering*, 2001, 27(8): 715-727
- [9] Holt R C. Binary relational algebra applied to software architecture[R]. CSRI Technical Report 345. Computer Systems Research Institute, University of Toronto, March 1996
- [10] Tarski A. On the calculus of relations[J]. *The Journal of Symbolic Logic*, 1941, 6(3): 73-89
- [11] Tilley S R, Paul S, Smith D B. Towards a framework for program understanding[C]//IWPC'96. IEEE Computer Society, 1996; 19-28
- [12] Rademaker P. Binary relational querying for structural source code analysis[D]. Utrecht, Netherlands: Software Technology Group, Universiteit van Utrecht, 2008
- [13] de Moor O, Verbaere M, Hajjiev E. Keynote address. QL for source code analysis[C]//SCAM. 2007; 3-16
- [14] Codd E F. A relational model of data for large shared data banks [J]. *Commun. ACM*, 1970, 13(6): 377-387
- [15] Holt R C, Winter A, Wu J. Towards a common query language for reverse engineering [R]. 8/2002. Fachbereich Informatik, Universitat Koblenz Landau, June 2002

(上接第 135 页)

- [14] Choi O, Han Sang-yong, Abraham A. Semantic matchmaking services model for the intelligent Web services [C]//International Conference on Computational Science and Applications. UK: IEE Press, 2006

- [15] Zhu Zheng-zhou, Wu Zhong-fu, Zhou Shang-bo. An e-Learning Services Discovery Algorithm Based on User Satisfaction[C]//The 2008 IEEE International Conference on Networking, Architecture, and Storage. 2008; 342-348