

基于本体的构件过程描述和检索*

杨涛 彭鑫 赵文耘

复旦大学计算机科学与工程系软件工程实验室, 上海市邯郸路 220 号, 200433

{yangtao,pengxin,wyzhao}@fudan.edu.cn

摘要: 基于构件的软件开发已经成为软件工程研究领域和软件开发实践过程中越来越重要的内容。在构件进行描述和检索的过程中, 检索主要的依据是基于构件 IOPE 进行的。本文作者基于现有的基于本体的构件描述和检索研究基础, 结合时序逻辑的相关理论, 提出了一种使用时序逻辑的描述和验证框架, 来对基于本体的构件描述进行检索结果的过程属性进行精化和验证。该方法可以有效的解决目前构件的描述和检索过程中, 构件的过程信息的验证问题, 提高构件检索结果的精准度和质量, 为基于构件的软件开发和软件重用提供帮助。

关键词: 构件, 检索, 语义, 过程, 时序逻辑

Ontology-based Component Represent and Retrieval Using Process Property

Yang Tao Peng Xin Zhao Wenyun

Software Engineering Lab of Fudan University, No.220, Handan Road, Shanghai, 200433

{yangtao,pengxin,wyzhao}@fudan.edu.cn

Abstract: CBSD (Component-based software development) is becoming more and more important in software engineering research and software development practice. Software repositories play a key role in CBSD and software reuse. Existing techniques for component represent and retrieval mainly base on the IOPE matching of components which offer low represent precision and lack on the process description ability. The authors introduce a novel approach that verify and refine the retrieval results using the pre-described process property based on the semantic-based component represent and retrieval technique and the theory of temporal logic. Evaluations suggest that process-based represent and retrieval method provides substantially higher retrieval precision and better retrieval quality than existing approaches.

Key Words: Component, Retrieval, Semantic, Process, Temporal Logic

*本文受国家863计划(课题编号2006AA01Z189), 国家自然科学基金(课题编号60473061、60473062), 973项目(课题编号2006CB3003002) 资助。杨涛 硕士研究生, 主要研究方向为软件工程、领域工程、企业构件化开发技术。彭鑫 博士、讲师, 主要研究方向为软件体系结构、软件再工程和软件产品线。赵文耘 教授, 博导, 主要研究方向为软件工程、电子商务、企业应用集成。

1、介绍

基于构件的软件开发已经成为软件工程研究领域和软件开发实践过程中越来越重要的内容。能否成功的实施基于构件的软件开发过程和软件重用,构件库扮演了一个关键性的角色[5]。由于基于本体的方法可以描述构件的内在知识及其语义信息,提供检索的知识匹配和推理能力,该方法已经成为该领域最新的研究方向。如何利用本体知识来描述构件库中的构件,对用户的构件查询需求进行描述,以及高效、准确的从构件库中查找出满足用户需求的构件,正是该项研究的主要研究内容[5][11]。

在传统构件进行描述和检索的过程中,构件被看出一个完全的黑盒,用户无需知道构件内如何进行操作和执行的,构件检索主要的依据也仅是基于构件 IOPE (输入、输出、前置条件以及后置条件) 来进行。但随着目前构件开发和提供者越来越多,构件结构越来越复杂,仅使用 IOPE 逐渐不能满足构件检索的需求,我们需要其他信息来辅助我们进行构件检索,提高检索质量。这时,就需要我们在构件库中,对构件有更详细的描述和说明,让构件库和构件检索者更多的了解构件的具体信息,方便其检索和挑选构件。这样不但可以方便用户查询到适合的构件,也为相对增强了构件的可复用性,这也是基于灰盒的构件检索的基本思路。

举个例子,现在的构件库中有两个进行网上购物的可重用构件,一个是普通的先付款后发货的在线购物服务,另一个是需要通过其他中间机构如 PayPal, 先进行将客户在中间机构账户内的相关资金冻结,然后先发货,用户确认后再由中间结构付款给商家的在线购物服务,由于购物需要的信息完全一样,结果也都是客户成功购入商品,所以两者的 IOPE 都是完全相同的,唯一的区别就在于工作流程的不同。如果用户需要检索一个能进行先发货后付款的在线购物构件,如果仅使用 IOPE 的检索方法,就无法对上述两个构件进行区分。

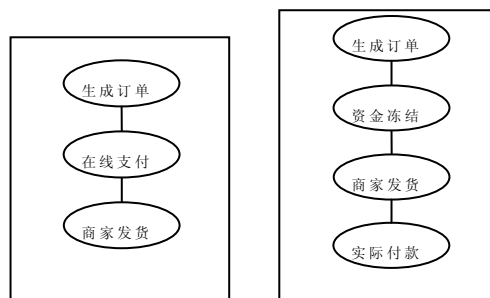


图 1: IOPE 无法区分的例子

构件的过程描述和检索就是基于这一目的提出的。构件的过程描述了构件工作的流程,我们可以使用构件库中构件的工作流程,来检查一个 IOPE 匹配的构件是否确实满足用户提出的需求。在过程描述和检索方面, Petri 网和工作流是目前用于描述构件工作过程和过程匹配中,应用最广泛的方法[3][10]。但由于 Petri 网和工作流本身不具有对本体的支持能力,加上使用这些技术来建立构件库的成本过高,在目前的构件库系统中,尚未得到实际的应用。

针对在构件检索中需要使用构件的执行过程信息来精化构件的检索结果,国内外也有一些学者在进行,主要有 Mark Klein 等的高精度构件过程检索方法[8]和李兴宇等的基于行为的服务匹配[12]等,但都有某些缺陷和不足。Mark Klein 等使用 Mit 的 Process Book[7]过程库,将过程本身看成可复用对象,使用自定义的 PQL 语言来检索和匹配库中的过程,但并没有说到如何将过程的检索应用到实际的构件检索中。李兴宇等[12]提出的基于行为的服务匹配方法将过程约束引入到服务检索中,通过对目标服务执行过程的描述进行匹配,而本文则更加关注于构件检索者所关心的某些时序属性

本文作者基于现有的基于本体的构件描述和检索研究基础,结合时序逻辑的相关理论,提出了一种使用时序逻辑的描述和验证框架,来对基于本体的构件描述进行检索结果的过程属性进行精化和验证。该方法可以有效的解决目前构件描述和检索过程中,构件的过程信息的验证问题,提高构件库检索的结果质量,为基于构件的软件开发和软件重用提供帮助。

2、背景知识

OWL-S

OWL-S(也称 DAML-S)[9]是一种用于描述 Web Service 的本体模型,它是由 Service Profile/ServiceModel/ServiceGrounding 3 个子本体所组成。

ServiceProfile 是服务基本信息,说明了本服务能做什么。它主要描述了哪些用于判断服务是否满足需求的信息的类型,并主要提供给服务搜索引擎服务。ServiceModel 是服务模型,主要描述了服务如何进行工作,以及当服务执行的时候,哪些行为会被执行。ServiceGrounding 是实现模型,描述了该服务如何被访问和执行。一般来说,服务的实现需要描述通信协议,消息格式,使用到的其他服务相关的细节,以及每一个过程模型中所提到的抽象类的实现等。

由于 Web Service 本身就是构件的一种特例,因此,基于构件的软件开发中的技术和成果同样也能适用于基于 Web Service 的软件开发中来。而 OWL-S 作为描述 Web Service 的最常用工具,目前在基于 Web Service 的应用系统构建中,已经得到了较广泛的应用。基于 OWL-S 的 Web Service 检索,也是这些应用中的一项重点研究工作[6]。

过程建模

一般来说,为了详细描述一个构件的执行过程,我们需要使用过程模型来进行建模和描述。在 OWL-S 中,我们可以通过服务模型中的过程模型来描述服务的执行过程。

OWL-S 将服务的执行分为 3 种过程,原子过程,一般过程和复合过程。复合过程就是通过过程控制结构,像顺序、分支等,被分解为其他过程的复合过程。为了详细描述复合过程,OWL-S 提供了 8 种过程控制结构,分别是 Sequence(顺序执行)/Split(并发执行)/Split + Join(并发执行并无需同步)/Unordered(无序执行)/Choice(选择执行)/If-Then-Else(条件执行)/Iterate(对某类中的所有元素迭代重复)/Repeat-Until(重复直到条件满足)。

构件检索

一般来说,构件检索的过程包括构件描述,检索请求描述,检索请求匹配,检索结果精化等步骤,在检索请求匹配这一步骤中,又包括构件 IO 匹配,构件 PE 匹配以及构件执行过程匹配三种主要匹配方法。目前的研究主要是基于 IOPE 的匹配进行的,特别是针对 IO 的匹配,已经有很多比较成熟和高效的检索匹配方法,例如: SWORD, SPARTACAS, OWLS-MX 等[8][2][6],而对于构件过程的匹配,还少有这方面的研究。

时序逻辑

时序逻辑是一阶逻辑的一个可判定真子集,用于描述系统中同时间相关的行为以及规则。线性时序逻辑是时序逻辑的一个有效的形式化表示方法,它通过形式化的语义,定义了时序逻辑的线性关系,并保持了时序逻辑的可判定性。

目前对线性时序逻辑进行逻辑判定的工具主要有 Spin, TLQSolver 等[4][11]。利用这些工具提供的一些描述语言和检测方法,可以对过程库以及基于过程库的线性时序逻辑行为序列进行可行性判定。

3、应用过程判定的构件检索方法

作者提出的一种使用时序逻辑的描述和验证框架,用于在构件检索的过程中应用过程属性,对检索结果进行精化,其主要步骤如下:

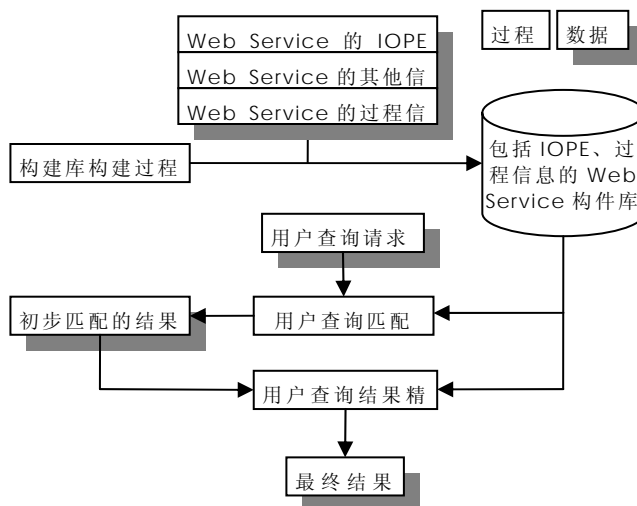


图 2：应用过程判定的构件检索框架

图中包括两部分主要的步骤,一部分是基于本体的构件库的构建步骤,另一部分是使用上述构件库,对用户的构件检索进行查询并返回匹配

的结果的过程。

在基于本体的构件库的构建步骤中，我们以 Web Service 为构件载体，使用 OWL-S 作为描述 Web Service 的语言，在服务基本信息、服务模型和实现模型三个描述框架中，除了描述此服务的一般 IOPE 信息外，并通过 Process Model 来描述其的过程信息。

在使用上述构件库，对用户的构件检索进行查询并返回匹配的结果的过程中，涉及到检索请求的描述方法、检索请求的匹配查询、检索结果的精化等步骤。检索请求的描述包括对待检索服务的 IOPE 描述，已经该服务的过程描述；检索请求的匹配查询主要过程是针对 IOPE 的请求，利用资源的描述和本体的推理机制，在构件库中初步查询出满足用户需求的构件结果集；检索结果的精化步骤是比较检索结果集和检索请求，从本体推理，过程验证的角度，过滤和筛选结果集，得到同用户查询匹配度更高的精化结果。

在上面的步骤中，本体的构件库的构建步骤、对待检索服务的 IOPE 描述、使用 IOPE 的构件查询等内容已经有大量文章进行研究，我们这里就不再一一展开，下面主要描述构件检索中的过程因子描述方法，以及使用时序匹配方法精化检索结果集的具体实现步骤。

3.1、构件检索中的过程因子描述方法

基于 LTL 的过程因子描述

为了实现在构件检索的过程中，针对检索条件的过程描述，我们提出了以线性时序逻辑为基础的过程因子描述方法。在这种描述方法中，我们将常用的过程检索条件整理归纳为如下。

首先，我们定义构件描述中的任何一个原子活动为线性时序逻辑中的一个原子，记为 p ，构件的过程模型就是由一系列活动 p_1, p_2, \dots, p_n 组成。对构件过程因子的检索，也是使用 p 的线性时序逻辑组合得到的执行序列表达式来进行描述。

下面，我们给出 LTL 逻辑算子的过程检索意义。

| 用 LTL 表达式 | 过程因子检索意义 |
|-----------|----------------|
| $\Box p$ | 构件执行过程中，过程 p |

| | |
|----------------------|-----------------------------------|
| | 一直在运行 |
| $\Diamond p$ | 构件执行过程中，过程 p 一定运行过一次 |
| $!p$ | 构件执行过程中，过程 p 未运行 |
| $p_1 \cup p_2$ | 构件执行过程中， p_1 过程的结束之后， p_2 开始 |
| $p_1 \text{ R } p_2$ | 构件执行过程中， p_1 过程的开始才引发 p_2 的结束 |

表 1：LTL 语法定义

对上述 LTL 逻辑算子组成的过程检索条件，我们还可以通过普通逻辑组合的 \neg , \wedge , \vee , \rightarrow 4 种关系，以及对过程的执行结果的判定来组成更复杂的逻辑表达式。

例如，现在我们的过程库中包括如下原子过程 {MakeOrder, PayOrder}，我们可以使用 LTL 来定义出如下的过程关系：

表示先发货后付款的过程关系：

| |
|--------------------------|
| $!PayOrder \cup Deliver$ |
|--------------------------|

3.2、构件检索中的时序匹配方法

在进行构件检索的过程时，当构件的 IOPE 检索条件都已经满足后，我们需要对使用上面介绍的线性时序逻辑的过程因子进行匹配和结果精化。目前主要有三种对时序条件进行匹配的方法，一是对时序因子进行基于已知条件的逻辑判断^[8]，二是现有构件进行遍历，检查时序条件的出现情况^[12]，三是使用现有的工具直接对过程库和条件进行检查^{[1][4]}。由于第一种方法在使用时序逻辑算子局部判断结果作为最终结果，存在漏判或错判的情况，第二种方法在复杂性方面存在状态空间爆炸的问题，经过权衡，作者选择第三种方法来进行时序条件的匹配。

Spin 是目前应用最为广泛的检索 LTL 的执行过程的工具，我们可以利用 spin 工具来帮助我们自动进行过程因子的时序匹配。Spin 使用 PROMELA 语言来描述过程库和待检查的 LTL 行为序列，其具体描述方法为可以参见^[4]。

我们进行时序匹配的基本步骤如下：

1) 将构件库中由于描述构件过程组合的原子过程知识，建模为 PROMELA 语言模型中的单

独原子过程，并通过全局变量标识过程的成功和失败。例如，我们可以使用下面的 PROMELA 来标识一个付款过程：

```
bool payorder_return;
proctype payorder(){
    if
        ::payorder_return=true;
        ::payorder_return=false;
    fi
}
```

2)完成对原子过程的描述之后，我们需要将构件库中实际构件的执行过程使用 PROMELA 对其过程的组合进行建模。例如，下面我们给出一个在线购物的组合过程，描述了先付款后发货，发货失败需要退款的过程组合：

```
proctype OnlinePaymentService(){
    run makeorder()->
    if
        ::(makeorder_return == false) ->
            goto endproc
        ::(makeorder_return == true) ->
            goto payproc
    fi;
    .....
endproc:
    skip;
}
```

3)完成组合过程建模之后，就需要为每一个构件生成可以执行的过程序列，为下一步验证做准备。过程序列只需要指定其 init 方法即可。

4)在用户进行查询的过程中，我们利用 spin 工具，将用户查询的 LTL 条件，同现有模型进行匹配。

上面我们描述了使用 spin 进行构件查询结果精化过程中的时序匹配流程。该流程的前 3 步均可在构件入库的时候，由构件库管理员使用构件作者提供的 OWL-S 过程模型中的相关信息预先完成，最后一步也是直接调用 spin 工具的命令行，来指定的执行序列进行 LTL 语句过程验证，因此，在用户的构件检索步骤中，无需人工

参与，完全可以实现自动化进行。

4、基于本体的过程描述和检索的测试应用

作者通过对 OWL-S 的示例构件[9]，以及 Process Book[7]中的有关在线销售的 Web Service，使用不同的 LTL 查询条件对其进行查询，用于检查本算法的查全率和查准率。

实验所用到的构件库一共包括 13 个在线销售的例子，它们具有相同的 IOPE,但在实现过程上有所区别，我们使用下面的查询 LTL 条件进行查询。

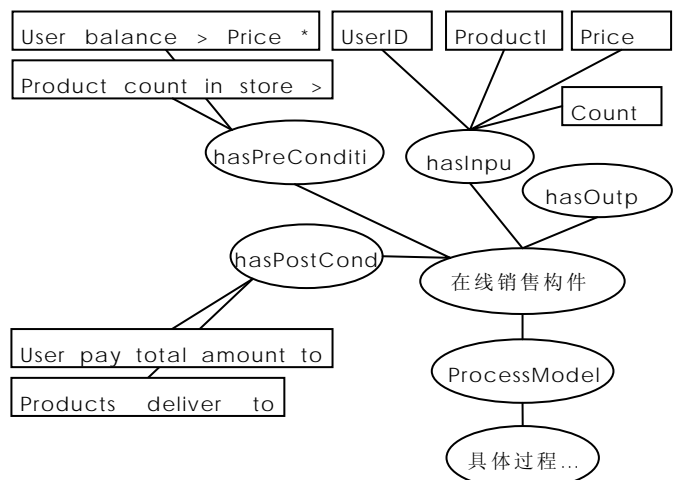


图 3: 待检索的在线销售构件的 IOPE 情况

| 查询 | 查询条件 | 查全率 | 查准率 |
|----|---|-----------------|---------------|
| Q1 | 仅 IOPE | (13/13) 100% | (4/13) 31% |
| Q2 | [Query1 &] (PayOrder R Deliver) | (6/6) 100% | (4/6) 67% |
| Q3 | [Query2 &] (◇Deliver V (!Deliver U !PayOrder)) | (3/4) 75% | (3/3) 100% |

表 2: 使用各种条件进行检索的查全率和查准率

Q1 查询的是 IOPE 符合在线销售的所有构件，Q2 查询的是除了 Q1 条件外，还需要先发货后付款的构件，Q3 查询的是除了 Q2 条件外，还需要处理发货失败的话，不能付款的构件。

通过实验数据我们看到，由于选择的构件仅是针对在线销售的，所以条件最宽的 Q1 的查全率最高，而通过 LTL 进行过程精化的查询，对那些没有完整描述过程的构件是无法查出的，所以

查全率有所下降。而查准率方面, 可以看到仅 IOPE 查询的查准率是无法满足精确的用户需求的, 其中包括大量明显不符合用户对过程需求的构件存在, 而 Q3 由于较精确的描述了构件, 因此查询结果得到的构件完全匹配用户的需求。

通过上面的实验, 我们可以看到, 使用基于本体的过程描述和检索, 在构件检索中能起到大提升查准率的效果, 并且可以通过对构件过程的详细描述, 来有效降低查全率的损失。

5、问题和未来研究

本文给出了一种使用时序逻辑的描述和验证框架, 来对基于本体的构件描述进行检索结果的过程属性进行精化和验证。实验证明, 相对于传统的 IOPE 的查询方式, 使用过程因子的查询精化可以得到更加符合用户需求的查询结果集。

但该方法在实际应用过程中, 还存在一定的问题。首先, 对于查询请求的描述, 要求使用较为抽象的线性时序逻辑来描述用户需要的构件过程, 对用户的能力提出了一定的要求。其次, 使用过程因子对检索结果集的进行精化的过程中, 需要预先给构件库中的构件使用 PROMELA 语言进行 2 次描述, 尚未达到实际应用的步骤。

我们将在以后研究中, 就下面 3 个问题进行进一步的展开和深入研究。

- 1、使用过程因子对检索结果进行精化过程中, 能否自动将 OWL-S 的过程模型转化为相应的 PROMELA 描述。
- 2、匹配过程的精确性、完整性以及性能的问题。
- 3、如何使用本体中的知识推理能力, 来增加过程因子检索和验证的能力。

6、引用

[1] Arie Gurfinkel, Marsha Chechik, Benet Devereux, Temporal logic query checking: a tool for model exploration, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING Vol. 29, No. 10; OCTOBER 2003, P898-914

[2] Brandon Morel, Perry Alexander, SPARTACAS: Automating Component Reuse and Adaptation, IEEE TRANSACTIONS

ON SOFTWARE ENGINEERING, VOL. 30, NO. 9, SEPTEMBER 2004

- [3] Francesco Calabrese, Giovanni Di Dio, Anna Rita Fasolino, Porfirio Tramontana, A Methodology for Searching Reusable Business Processes, Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services, 2006
- [4] Gerard J. Holzmann, The Model Checker SPIN, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 23, NO. 5, MAY 1997
- [5] Mark Klein, Abraham Bernstein, Toward High-Precision Service Retrieval, IEEE INTERNET COMPUTING, JANUARY • FEBRUARY 2004, P30 - 36
- [6] Matthias Klusch, Benedikt Fries, Katia Sycara, Automated Semantic Web Service Discovery with OWLS-MX, AAMAS 2006 May 8-12, 2006
- [7] MIT Process Handbook, <http://process.mit.edu/>
- [8] Shankar R. Ponnekanti, Armando Fox, SWORD: A Developer Toolkit for Web Service Composition, Proceedings of the Eleventh International World Wide Web Conference, 2002
- [9] OWL-S, <http://www.w3.org/Submission/OWL-S>
- [10] 段风琴, 李祥, Petri 网性质的线性时序逻辑描述与 Spin 检验, 计算机科学 2006, Vol.33, No.5
- [11] 彭鑫, 赵文耘, 肖君, 基于本体的构件描述和检索, 南京大学学报 (自然科学版), 2005 年, 第 10 月第 41 卷计算机专辑.470-47
- [12] 李兴宇, 殷琴, 胡昊, 吕建, SOBECA 中间件上支持行为的服务注册和查找机制, 计算机工程与科学, 2006, Vol 28, No.A2

