

基于程序切片和服务构件的语义 Web 服务组合

王 伟,杨 庚,张迎周,孔华云

(南京邮电大学 计算机学院 宽带无线通信与传感网技术教育部
重点实验室,江苏 南京 210003)

摘 要:文中利用程序切片技术,根据用户的请求,将目标细分成一个个单一的需求,利用程序切片对依赖图进行函数抽取,将服务中实际需要使用的功能抽取出来,形成更小型的原子服务,作为服务构件并纳入构件库,进行服务组合。同时通过使用 Wordnet 词典,对这些关联服务的信息进行语义上的匹配,进行服务的组合,生成可行的组合方案,然后实现组合服务的绑定并执行服务。通过引入服务构件的思想,提高服务组合的复用性、扩展性和针对性,提高了服务的使用效率,并通过 Wordnet 增强了服务的语义性,使服务组合更加智能化。

关键词:Web 服务组合;程序切片; Wordnet; 服务构件

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2011)11-0141-04

Semantic Web Service Composition Based on Program Slicing and Web Component

WANG Wei, YANG Geng, ZHANG Ying-zhou, KONG Hua-yun

(Key Lab of Broadband Wireless Communications and Sensor Network Technology, College of
Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: Adopt program slicing to divide user's complicated goal into smaller single goals, then extract function from function dependence graph and get the accurate function really need. Regard these smaller Web services as Web components and put them in the Web component library to compose. Meantime, based on the Wordnet, match the information of the coordinated Web services on the ground of semantics. At last, generate the Web service composition plan, realize the composition and execute it. Therefore, improve the ability of reusing, specializing, extending and intelligence of Web service composition by using program slicing and Wordnet.

Key words: Web service composition; program slicing; Wordnet; Web component

0 引 言

随着互联网技术的迅速发展,Web 服务作为一种崭新的分布式计算模型已经在电子商务、企业应用集成等领域扮演着越来越重要的角色。Web 服务具有良好的封装性、松耦合性和跨平台性,因此逐渐成为实现面向服务体系结构(SOC)的重要技术。近年来,随着用户需求的不断增多,单一的 Web 服务已经远远不能满足用户的需要,因此将各个单一的 Web 服务进行有序、有效的整合来满足用户日趋多样化的需求已经成为了学术界和工业界共同关注的问题^[1]。

目前,主流的 Web 服务方法包括基于工作流的服务组合、基于形式化描述方法的服务组合、基于人工智能的服务组合^[2]。三类方法各有利弊:基于工作流的服务组合方法主要是在工业界中发展,以 BPEL 为代表,具有很强的流程操作性,但其执行流程是固定的,缺乏灵活性和自适应性;基于形式化描述的方法多为学术界所讨论,有较为完善的理论基础,但应用能力较弱;在语义和本体库概念下,基于人工智能的服务组合方法发展迅速,W3C 先后推荐了 RDF, DAML+OIL 和 OWL-S^[3]作为本体描述语言标准,但是由于每个客户端的本体库不一定相同,所以这些方法受到了一定的限制,而且服务越多,描述就越复杂越庞大。文中提出的基于程序切片和 Wordnet 的语义 Web 服务组合方法,通过将用户的需求进行分解,然后利用程序切片技术从服务中抽取实际所需的功能函数,再利用 Wordnet 进行语义的匹配,进行服务的输入输出组合,

收稿日期:2011-04-18; **修回日期:**2011-07-25

基金项目:国家(973)规划课题(2011CB302903);国家自然科学基金资助项目(60873231);江苏省自然科学基金(BK2009426)

作者简介:王 伟(1987-),男,硕士研究生,研究方向为 Web 服务组合。

从而简化了服务组合的复杂性,减轻了组合的负担,减少了组合服务执行时间。

1 基本概念

1.1 程序切片和 Haskell 语言

程序切片^[4]是由 M. Weiser 提出的一种重要的程序分析理解方法。给定一个源程序 P , 一个程序点 p 和一个变量集 V , 当源程序执行到程序点 p 时, 源程序 P 中所有对 V 中变量有影响的程序片段称为切片。程序切片可分为静态切片和动态切片两种, 两者的区别在于后者是根据特定输入情况而定的; 程序切片也分为前向切片和后向切片两种, 前向切片是指指定一个变量, 受到该变量影响的所有语句, 而后向切片指指定一个变量, 对该变量造成影响的其他语句。目前主流的切片算法总共有四种, 基于数据流方程的切片算法、基于信息关系的切片算法、基于程序依赖图的图形可达算法、基于波动图的切片算法, 文中采用的是基于程序依赖图的图形可达算法的前向切片技术^[5]。

基于依赖图的函数抽取^[6], 首先对 Web 服务源码进行语法分析, 得到显示服务中所有函数之间依赖关系的函数依赖图, 之后根据用户的需求选取抽取点, 将跟这点有依赖关系的函数都抽取出来, 形成函数依赖子图, 然后从服务源代码中将相关的函数的代码抽取出来形成切片代码。

图 1 就是功能函数的抽取过程^[7]。

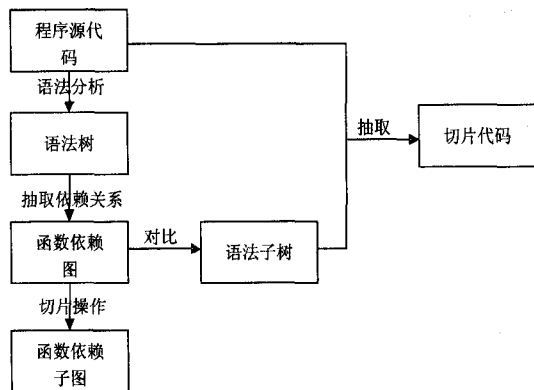


图 1 功能函数抽取过程

以上功能的实现都是用 Haskell 语言实现的。Haskell 语言是以逻辑学家 Haskell B. Curry 的名字命名, 是一种基于 λ 演算的多态类型的、惰性的纯函数式编程语言, 具有表达力强的语法以及丰富的内置数据类型。Haskell 具有以下特点:

第一, 和命令式的编程语言代码相比, Haskell 代码就是一个大型的表达式, Haskell 语言代码更加简单, 更加容易理解, 更加容易维护。

第二, Haskell 采用多态机制, 比如 C 语言的排序算法只能对其定义类型的数组进行排序, 但 Haskell 语

言的排序算法无须规定特定的类型, 一切可以进行大小比较的数据类型(如整型、字符型)数组都可使用同一个排序算法进行排序。

第三, Haskell 语言无内核转储, 避免了某些类型转换带来的问题。

第四, Haskell 语言减轻了程序员手工分配内存的负担, 隐式化地分配、初始化内存, 最后利用垃圾回收器恢复内存。

第五, Haskell 的惰性使得代码的组合更加容易, 模块性更强。

第六, Haskell 语言中的高阶函数为抽象提供了技术基础。在 Haskell 语言中, 函数不但是参数, 而且是返回值, 这使得程序的结构化和模块化更加容易实现。

文中就是利用 Haskell 语言辅助实现 Web 服务的原型系统开发^[8]。

1.2 WordNet

传统字典一般都是按字母顺序组织词条信息, 但忽略了词典中同义信息的组织问题。由 Princeton 大学研制成功的一个联机英语词汇检索系统 WordNet^[9], 作为语言学本体库, 同时又是一部语义词典, 在自然语言处理研究方面有着独特的用处。

它与其他标准词典最显著的区别在于: 它将词汇分成五个大类: 名称、动词、形容词、副词和虚词。它是根据词义而不是词形来组织词汇信息, 这就在进行服务的匹配时提供了语义匹配的可能, 而不是简单地进行字符串匹配, 使得选取的服务更加智能, 更加全面, 可以说, WordNet 就是一部语义词典。WordNet 中除了具有中心角色的同义关系外, 还有反义关系、上下位关系和部分关系。反义关系为 WordNet 中的形容词和副词提供了一种中心组织原则。与同义词和反义词都是词形之间的词汇关系不同, 上位关系(hypernymy)/下位关系(hyponymy)是词义之间的语义关系。例如: {樟树}是{树}的下位词, {树}又是{植物}的下位词。

下面介绍 WordNet 名词体系中一个重要知识——概念链^[10]。如果有一同义词集合(即概念)没有上位同义词集合(即上位概念), 则称之为独立起始概念(Unique Beginner)。在 WordNet 名词体系中, 共有 25 个独立起始概念。其他名词通过上位/下位关系与这 25 个独立起始概念构成 25 个独立的层次结构。也就是说, 标识着某个起始概念特点的属性将它的所有下位概念所继承, 而这个起始概念就可以看作为是该语义领域内的所有概念(同义词集合)的一个原始语义元素。每一个同义词集合(synonymy set)均有惟一的一个编号, 这个编号就称为词典编撰 ID(Lexicographer ID)。

概念链一般的定义是这样一种结构: $\delta = (C,$

<), 其中 C 代表的是概念集合, $<$ 代表概念间的下位/上位关系。也就是说概念链 δ 是由 C 概念集合中的概念通过概念间的上位/下位关系连接而成。在 WordNet 名词体系中, 定义概念链 (Concept Chain) 如下:

$$\delta: = (C, <) < \text{UBCi},$$

其中 UBCi 表示 WordNet 名词体系的一个独立起始概念, C 代表的是概念集合, $<$ 代表概念间的下位/上位关系。也就是说概念链 δ 是以一个独立起始概念 UBCi 为链首, 通过概念间的上位/下位关系与 C 概念集合连接而成。同时 C 概念集合中的概念也是通过概念间的上下位关系进行连接。

例如, 对于一个单词 “football”, 它有很多的同义词, 每个同义词都有一个相应的词典编撰 ID (Lexicographer ID), 如其中一个含义 ball 的 ID 为 (2681909)。在 WordNet 中: 词 “football” 有两个义项, 即它在两个同义词集合中出现, 它所代表的概念通过下位/上位关系联系起来, 从而构成概念链。概念链的首端对应的就是 WordNet 中的独立起始概念。比如: 概念链 ch1 可以表示为: (3255461) < (2681909) < (3289024) < (3174243) < (3443493) < (19244) < (2645) < (16236) < (1740)。其中 (3255461) 作为概念链的末端代表的是词 “football” 的一个义项, 而 (1740) 是 WordNet 中的独立起始概念, 成为概念链的首端。概念 “game equipment” (3289024) 是概念 “ball” (2681909) 的上层概念, 表达的语义更抽象。

1.3 基于 WordNet 概念树的语义相似性计算

为了能够组合服务, 需要对各个子服务之间的接口信息进行匹配, 而这种匹配是语义匹配, 是利用 WordNet 进行语义上的分析和对照, 因此需要计算出接口信息之间的匹配程度, 只有当相似度达到一定的数值, 即门限值的时候, 才认为这两个接口信息是匹配的, 可以组合, 因此对语义相似性的计算尤其重要。文中采用的是由 Nuno Seco^[11] 等提出的一种基于信息论模型的语义相似性计算方法, 其计算公式如下:

$$\text{sim}_{\text{jen}}(c1, c2) = 1 - \left(\frac{ic_{\text{wn}}(c1) + ic_{\text{wn}}(c2) - 2 \times \text{sim}_{\text{res}}(c1, c2)}{2} \right) \quad (1)$$

其中, $\text{sim}_{\text{res}}(c1, c2)$ 采用的是 Resnik^[12] 等提出的语义公式 $\text{sim}_{\text{res}}(c1, c2) = \max_{c \in s(c1, c2)} IC_{\text{res}}(c)$, 式中 c 为层次结构树中的概念, $IC_{\text{res}}(c) = -\log p(c)$, 用来得到各概念信息量的值。 $p(c)$ 为概念 c 在文本集中出现的概念, 即在文本集中遇到这个概念 c 的一个实例的概念。对于概念树中只有一个根节点的概念, 其出现概率为 1, 则信息量为 0。而一个概念的信息量是通过公式 (2) 表示的:

$$ic_{\text{wn}}(c) = \frac{\log(\frac{\text{hypo}(c) + 1}{\max_{\text{wn}}})}{\log(\frac{1}{\max_{\text{wn}}})} = 1 - \frac{\log(\text{hypo}(c) + 1)}{\log(\max_{\text{wn}})} \quad (2)$$

其中, 函数 $\text{hypo}()$ 返回的是输入的参数即一个概念的所有上/下位关系概念的数量, \max_{wn} 为一个常量代表所在 WordNet 分类体系中概念数最大的那棵树的节点即概念总数。另外需要注意的是, 由于 WordNet 中存在多重继承的情况, 因此, 对于 $\text{hypo}()$ 函数, 一个直接上/下位关系只需考虑一次即可。公式 (1) 的返回数值在 0 ~ 1 之间, 数值越高, 匹配程度越高。

这样, 就可以通过公式 (1) 计算出两个词语之间是否具有相似性, 然后进行匹配, 这样就大大增加了语言的智能性, 它不仅仅是简单的字符串匹配而已, 而是可以对各个原子服务的属性进行语义上的分析, 然后再判断是是否符合用户的需求。

1.4 服务构件

为了能够将服务中各种操作的功能有效地结合在一起, 满足用户多样化的需求, 需要使用一个灵活的、可选择的、可复用的 Web 服务组合方法, 文中引入服务构件^[13]的概念, 将 Web 服务的组合过程转变为服务构件的复用和细化过程。服务构件本身就是一个 Web 服务, 它是由已发布的 Web 服务组成。Web 服务构件的结构如图 2 所示。

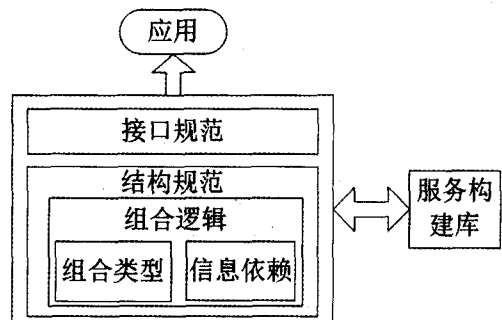


图 2 服务构件基本结构

服务构件主要包括两个部分, 接口规范和结构规范。接口规范用来与外界的程序进行互动, 服务构件本身是具有封装性的, 外界可以利用服务构件提供的这个公共接口来访问服务构件。结构规范包括三部分, 组合逻辑、端口类型和提供商。组合逻辑描述组合服务的结构, 包括组合类型和信息依赖。外界不需要知道服务组合逻辑, 只需通过接口访问服务构件。

2 服务组合的基本框架模型

文中将服务组合的整个过程分成 5 个阶段: 抽取阶段、预处理阶段、定义阶段、转换阶段和执行阶段。

在抽取阶段, 根据用户的具体需求, 将其逐个分解

成一个小小的任务,然后根据这些原子目标从服务中抽取能够完成其功能的功能函数,并生成函数依赖子图,形成新的程序切片。下面就是进行原子服务之间的一致性和兼容性检查。

如果两个操作的先行后续条件相同,且两个操作的输入输出都分别兼容,那么这两个操作兼容。如果一个服务 S_1 的活动中的操作在另一个服务 S_2 活动中都存在兼容的操作,那么服务 S_1 的活动和服务 S_2 的活动兼容。如果服务 S_1 的活动和服务 S_2 的活动兼容,服务 S_2 的内容大于服务 S_1 的内容,那么 S_2 和 S_1 兼容。如果服务 S_1 中某个操作的输出和服务 S_2 中某个操作的输入兼容,那么服务 S_1 顺从服务 S_2 。为了实现组合服务的复用性,文中定义了一个服务构件库。如果构件库中存在这样的服务构件,则直接使用此构件;如果构件库中不存在这样的服务构件,则从服务发现开始重新进行组合服务。构件库的存在避免了多余的重复的服务组合。

3 案例分析

利用 Haskell 语言开发五个服务,分别是查询天气服务、预订票务服务、预订酒店服务、景点订票服务和租车服务。对五个候选服务进行程序切片,将候选服务中满足用户请求的功能抽取出来,并得到这部分功能的函数依赖子图。图 3 显示了预订票务服务的函数依赖图。

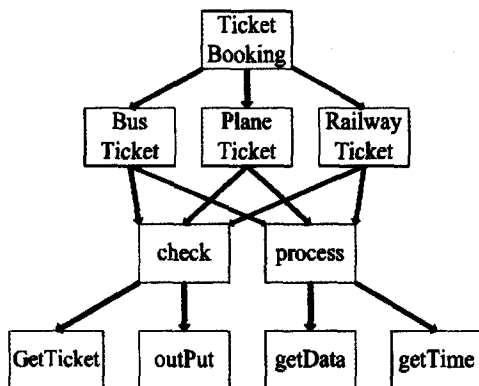


图 3 订票服务的函数依赖图

根据用户的需求,只需要原服务中的订飞机票的功能,不需要考虑乘坐公交或者铁路,于是抽取订订票服务中满足客户需求的飞机票预订功能,对预订票务服务进行切片,得到图 4 所示结果。其余四个服务将进行同样的操作,这里不再赘述。

得到所需要的原子服务之后,对抽取的结果进行一致性和兼容性检查,并利用 Wordnet 考察服务之间语义的匹配性,然后根据组合策略定义组合服务构件。

这样,就可以根据服务构件生成服务组合的关系图,接着就可以生成相应的 WSDL 文档并返回组合服

务的结果给用户。

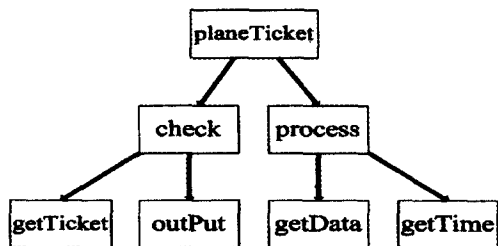


图 4 飞机票预订的函数及切片代码

可以看出,利用程序切片技术对原服务进行功能函数的抽取,从而可以提供更精准的原子服务操作,简化了服务组合的复杂性,同时利用 Wordnet 技术对各个原子服务之间的匹配进行了语义上的比较,增加了智能性,同时由于可以从构件库中重复使用已有的 Web 服务,因此提高了组合效率。

4 结束语

对于现在越来越复杂的用户需求,将目标任务进行一步步的分解,分解成一个个小的功能需求,然后就可以利用程序切片技术,对大型服务进行抽取,将其中实际使用的功能函数抽取出来,形成新的更加简单的原子服务,这些服务虽然功能没有原来大,但是对用户是恰好需要的,因而更具针对性。同时在组合的过程中,利用 Wordnet 技术对服务的输入输出进行了语义上的匹配,增加了可读性和智能性,使得服务的选择更加人性化。而进一步的研究工作是加强服务组合非功能属性的描述和服务组合的动态性。

参考文献:

- [1] Curbera F, Khalaf R, Mukhi N, et al. The Next Step In Web Services[J]. Communications of the ACM, 2003, 46(10): 29-34.
- [2] 岳昆, 王晓玲, 周傲英. Web 服务核心支撑技术: 研究综述[J]. 软件学报, 2004, 15(3): 428-442.
- [3] 吴善明, 沈建京, 韩强. 基于领域本体和 OWL-S 的 Web 服务组合方法[J]. 计算机工程, 2009, 35(21): 256-263.
- [4] Zhang Y Z, Xu B W, Shi L, et al. Modular monadic program slicing[C]//In: 28th Annual International Computer Software and Applications Conference, COMPSAC. Hong Kong, China: IEEE CS Press, 2004: 66-71.
- [5] Fu W, Zhang Y Z. Program Slicing Based Web-Service Generation and Composition[C]//IEEE International Workshop on Service-Oriented System Engineering (SOSE'10). [s. l.]: [s. n.], 2010: 189-192.
- [6] Zhang Y Z, Zhang W F. Haskell: A Modern Purely Functional Programming Language[J]. Journal of Nanjing University of Posts and Telecommunications, 2007, 27(4): 13-18.
- [7] 张迎周, 符炜, 郭铭铭, 等. 基于单子技术的服务计算形

(下转第 148 页)

法得出的高斯混合模型所转换出来的语音,具有较小的失真测度值,证明使用遗传算法对高斯混合模型的参数进行估计,能够一定程度上改善语音质量。

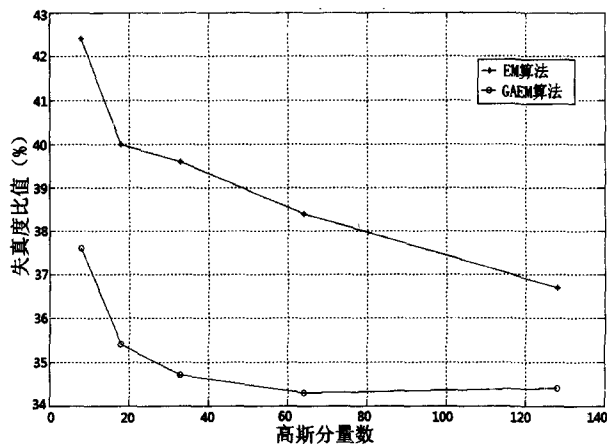


图 2 男声转女声失真度比值

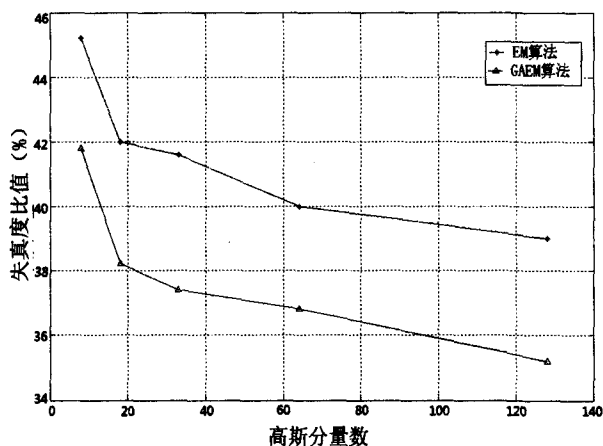


图 3 女声转男声失真度比值

4 结束语

采用遗传算法的全局搜索特性与 EM 算法相结合,实验证明文中的优化算法对高斯混合模型的参数估计有所改进。在提高转换后语音的质量方面,文中的算法是可行的。代价是增加了算法复杂度和运算时

间,实验中上述算法对模型参数进行估计时,训练时间常会成倍增加,与获得的质量改善相比,这是一个需要权衡考虑并加以进一步研究的问题。

参考文献:

- [1] Reynolds D A. An overview of automatic speaker recognition technology[J]. IEEE Trans on Speech and Audio Processing, 2002, 10(4): 472-475.
- [2] Kain. High resolution voice transformation[D]. Rockford: Rockford College, 1995: 47-52.
- [3] Zhang Kai, Zhu Lixin, Zhao Yizheng. Research on modified GMM based voice conversion method[J]. Technical Acoustics, 2008, 27: 392-397.
- [4] 张凯, 朱立新, 赵义正. 基于重训练高斯混合模型的语音转换方法[J]. 声学技术, 2010, 29(1): 52-55.
- [5] 吴庆棋, 林江云. 基于聚类优化 GMM 提高说话人识别性能的研究[J]. 计算机技术与发展, 2009, 19(4): 35-40.
- [6] 赵义正. 改进 GMM 谱包络转换性能的语音转换算法研究[J]. 科学技术与工程, 2010, 10(17): 4172-4174.
- [7] 王爱平, 张功营, 刘方. EM 算法研究与应用[J]. 计算机技术与发展, 2009, 19(9): 108-110.
- [8] 游源, 齐欢, 胡祥恩. 树状模型中 EM 算法的矩阵形式[J]. 计算机工程与应用, 2011, 47(5): 32-34.
- [9] Tang K S, Man K F, Kwong S, et al. Genetic algorithm and their applications[J]. IEEE Signal Processing Magazine, 1996, 13(6): 22-37.
- [10] 邹贻梅, 肖基毅, 龚向坚. 基于 Maximum Likelihood 与 HMM 的文本挖掘[J]. 计算机技术与发展, 2007, 17(12): 110-112.
- [11] Hong Q Y, Kwong S. A genetic classification method for speaker recognition[J]. Engineering Applications of Artificial Intelligence, 2005, 18(1): 13-19.
- [12] 田生文, 王伊蕾, 李阿丽. 一种应用复杂网络特征的 K-means 初始化方法[J]. 计算机工程与应用, 2010, 46(6): 127-129.
- [13] Seco N, Veale T, Hayes J. An Intrinsic Information Content Metric for Semantic Similarity in WordNet[C]//European Conference on Artificial Intelligence. [s. l.]: [s. n.], 2004.
- [14] Resnik P. Using Information Content to Evaluate Semantic Similarity in a Taxonomy[C]//Proceedings of IJCAI-95. Montreal, Canada: [s. n.], 1995: 448-453.
- [15] Yang Jian, Papazoglou M P. Web Component: A Substrate for Web Service Reuse and Composition[C]//Lecture Notes in Computer Science. [s. l.]: Springer, 2002: 21-36.

(上接第 144 页)

- [1] 式化方法研究[C]//全国服务计算学术会议. 出版地不详: 出版者不详, 2010: 9-17.
- [2] Zhang Y Z, Xu B W. An approach to dynamic program slicing based on modular monadic semantics[J]. Chinese Journal of Computers, 2006, 29(4): 526-534.
- [3] Miller G, Beckwith R, Fellbaum C, et al. Introduction to wordnet: an on-line lexical database[J]. International Journal of Lexicography, 1999, 3(4): 235-244.
- [4] Hirst G, St-Onge D. Lexical chains as representations of context for the detection and correction of malapropisms[M]//WordNet: An Electronic Lexical Database. [s. l.]: MIT Press, 1998: 305-332.