

## 基于遗传规划的服务组合模型及实现

邓磊, 吴健, 马满福, 胡正国

(西北工业大学 计算机学院, 陕西 西安 710072)

(denglei@mail.nwpu.edu.cn)

**摘要:**将服务组合的问题视为一个 Agent 动作规划的问题,给出了一个基于动作规划的服务组合模型。进而结合遗传规划的可变分层结构,提出了一个适应于上述模型的服务匹配算法。仿真实验表明,在具有关联特征的服务选择空间中,该算法比其他算法具有更好的寻优能力和更快的速度。

**关键词:**动作;遗传规划;服务匹配

**中图分类号:**TP393 **文献标识码:**A

## Service composition model and implementation based on genetic programming

DENG Lei, WU Jian, MA Man-fu, HU Zheng-guo

(College of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an Shaanxi 710072, China)

**Abstract:** Based on the feature abstraction in semantic service, the elementary service can be abstracted as action rule unit (ARU). So, a services composition model based on action planning was proposed. In order to satisfy the functional and nonfunctional needs, a service matching algorithm based on genetic programming (GP-SMA), was introduced by integrating with the variable and layered structure of genetic programming. The action rules among these ARU guide the planning process. Simulation results show that this algorithm exhibits some good performances such as find-best ability and efficiency in the selection space of associated service.

**Key words:** action; genetic programming; service matching

### 0 引言

在一个动态、开放的计算环境中,为了向用户提供随时随地的透明服务,需要系统能够根据用户的移动性、终端设备的资源受限性、任务及环境的多样性和多变性等,将多个服务进行组合,来构造个性化的应用程序。此时,如何将单个的、孤立的服务动态地组合起来,形成更加复杂的服务的协议和标准,就成为面向服务体系结构下实现按需计算的关键。服务组合的本质是将一系列服务组件按某些规则组合起来,使它们相互协同执行。服务组合模型可以反映上层应用业务模型的变化,并通过功能的抽象描述和底层的服务实体相结合,以适应底层信息基础设施的变迁。

当前,基于业务逻辑、模式匹配、工作流等的服务组合能够较好地解决的服务组合的可行性问题,但组合的动态适应度不高。我们希望借助已有的服务语义抽象方法,找到能够同时满足服务组合的功能性和非功能性需求的途径。

在一些基于 Agent 的服务组合方法中,服务被视为 Agent 的行为<sup>[1]</sup>,这样在服务智能组合过程中将能够充分利用 Agent 的自治性、主动性和推理性等特性。作为一种抽象方法,当把服务看作为动作时(action),则可以用参数、前提和结果等来描述服务,使得服务的组合问题可以通过动作规划等形式化方法来解决。

### 1 服务组合模型

#### 1.1 动作规则单元

动作规则单元  $u$  是对服务元素基于语义特征的抽象。每

个动作规则单元可以被描述成一个四元组  $u = \{S_{pre}, S_{goal}, A, C\}$ ,其中, $S_{pre}$  是初始状态, $S_{goal}$  是目标状态, $A$  是可以执行的动作集合,关于执行代价、与其他规划单元的交互代价等特征则通过属性信息集合  $C$  标识。

#### 1.2 服务组合模型

称  $x$  是基于动作规则单元集合  $U$  上进行的的服务组合。相似的,它也是一个四元组  $x = \{u_A, i, u_B, o\}$ ,其中  $u_A, u_B \in U, i \in P_{in}(u_A), o \in P_{out}(u_B)$ ,  $P_{in}(u_A)$  是  $u_A$  的输入端口集合,  $P_{out}(u_B)$  是  $u_B$  的输出端口集合。这种服务的组合要求满足  $i$  和  $o$  的兼容性匹配关系,因而服务组合  $x$  的必要条件为:  $\forall i \in P_{in}(u_A), o \in P_{out}(u_B), (-, i, -, i) \in x \Rightarrow (i, o) \in R$ 。

在服务组合的匹配过程中,通常存在许多的约束条件。作为一般性的描述,此处用约束函数  $f_{constr}: U \rightarrow 2^N, N = \{1, 2, \dots, n\}$  来表示这种约束关系。

#### 1.3 服务组合规划过程

对于一个确定的任务需求,结合上述模型,提出的服务组合规划过程如下:

- 1) 由服务元素  $S$  集合抽象得到动作规则单元集合  $U$ ;
- 2) 抽象特征集合  $P_{in}$  和  $P_{out}$ ,  $P_{in}$  是动作规则单元的输入端口集合,  $P_{out}$  是输出端口集合;
- 3) 兼容性匹配关系  $R$  的定义为:  $R \subseteq P_{in} \times P_{out}$ ;
- 4) 由任务分解的目标特征  $M_{out} \subseteq P_{out}$ ;
- 5) 由任务请求获知的已知条件  $M_{in} \subseteq P_{in}$ ;
- 6) 对动作规则单元的约束条件  $f_{constr}: U \rightarrow 2^N$ 。要求寻找到:

收稿日期:2006-01-10;修订日期:2006-04-06 基金项目:国家自然科学基金资助项目(60373078)

作者简介:邓磊(1973-),男,安徽定远人,讲师,博士研究生,主要研究方向:语义网、Web 计算; 吴健(1961-),男,江苏盐城人,教授,主要研究方向:嵌入式系统、软件理论; 马满福(1968-),男,甘肃甘谷人,副教授,博士研究生,主要研究方向:网络计算、计算机系统结构; 胡正国(1939-),男,陕西西安人,教授,博士生导师,主要研究方向:软件工程、软件理论。

- 1) 一组动作规则单元  $U_x \subseteq U$ ,  $M_{in} \subseteq P_{in}(U_x)$ ,  $M_{out} \subseteq P_{out}(U_x)$ ;
- 2) 该组动作规则单元满足约束条件  $f_{constr}$ ;
- 3) 这些动作规则单元间的组合  $x$  遵循兼容性匹配关系  $R$ ;
- 4) 对于每个动作规则单元  $\forall u_i \in U_x$ , 其输入端口  $P_{in}(u_i) - P_{in}(u_i) \cap M_{in}$  都能与相连的多个动作规则单元的输出端口  $\{o_1, o_2, \dots, o_m\} \subseteq \{P_{out}(u.), P_{out}(u.), \dots, P_{out}(u.)\}$  满足兼容性匹配关系  $R$ ;
- 5) 一条满足兼容性匹配关系  $R$  的可达路径。

## 2 基于遗传规划的服务匹配算法

在上述模型中,由于服务的搜索空间很大,并且基于  $U$  上的组合状态数更大,因此在动作规则单元的组合过程中,其组合规划算法将决定任务求解方案的优劣。

基于需求驱动的随机规划算法 (Requirement-Driven Random Planning, RDRP), 不考虑服务之间的关联关系, 因而搜索空间太大、命中率低。而基于服务聚集的启发式规划算法 (Service-Cluster Based Heuristic Planning, SCBHP), 虽然考虑了服务的关联, 建立了根据目标特征索引对应的服务聚集, 并在此空间中进行服务匹配, 但是只有在更好的解位于当前解附近的前提下它才能向优解搜索, 容易陷入局部最优。

20 世纪 90 年代初, Koza 提出的遗传规划方法 (Genetic Programming, GP) 对群体中表示独立的计算机程序的个体进行操作, 采用了更为灵活的可变分层结构反映求解问题特征<sup>[2]</sup>, 并按照遗传算法的思想来设计进化算子。考虑到服务匹配在某种程度上主要是由各种搜索构成的, 因此, 我们提出的基于遗传规划的服务匹配算法 (Service Matching Algorithm Based on Genetic Programming, GP-SMA) 便是基于这样一种思想来设计。

### 2.1 GP-SMA 算法

- 1) 读入关联服务的索引结构;
- 2) 读入任务参数  $M_{in}$ ,  $M_{out}$ ;
- 3) 读入 GP 参数: 种群规模 ( $pS$ )、最大进化代数 ( $max\_G$ )、遗传算子  $GO$  集合、遗传算子概率因子向量  $P_{GO}$ ;
- 4) 根据  $M_{in}$ ,  $M_{out}$  构造具有  $pS$  个体的初始群体;
- 5) 计算每个个体染色体的适应度函数, 保存最好的个体; 旋转赌轮选择  $pS$  个体作为父代; 对父代按照遗传算子执行进化操作以产生新一代群体; 将新一代群体代替当前群体; 重复上述过程, 直至达到  $max\_G$ ;
- 6) 返回最好个体的染色体以指导基于服务组合的应用运行。

上述算法的主要思路是将 GP 方法应用于基于关联服务的有向图中, 每个个体的染色体用变长的层次结构来表示一个  $M_{in} \Rightarrow M_{out}$  的服务组合规划方案。算法通过对群体的多个个体执行进化操作以促进群体的进化, 并利用适应度函数作为启发信息来指导算法朝规划的优化方向发展。

### 2.2 个体的编码方式

根据任务特征 ( $M_{in}$ ,  $M_{out}$ ), 每个服务组合规划方案的起始状态  $ps_0$  由  $U_{in}$  中的一些动作元素组成, 而其终止状态  $ps_t$  则由  $U_{out}$  中的多个动作元素组成。  $U_{in}$ ,  $U_{out}$  由算法通过关联动作的索引结构求取。其过程及要求如下:

$$\begin{aligned}
 U_{in} &: \{u_{in} \mid u_{in} \in U_{in}, M_{in} \cap I(U_{in}) \neq \emptyset\} \\
 U_{out} &: \{u_{out} \mid u_{out} \in U_{out}, M_{out} \cap O(U_{out}) \neq \emptyset\} \\
 ps_0 &: ps_0 \subseteq U_{in} \wedge M_{in} = P_{in}(ps_0) - R(ps_0) \\
 ps_t &: ps_t \subseteq U_{out} \wedge M_{out} = P_{out}(ps_t)
 \end{aligned}$$

其中,  $R(ps_0)$  表示某个  $u_i \in ps_0$  的部分初始状态被另一个  $u_j \in ps_0$  的目标状态的兼容性匹配所满足。此时, 一个服务组合的规划方案则是从起始状态  $ps_0$  到终止状态  $ps_t$ 、由若干动作元素组成的服务兼容性匹配过程。

### 2.3 适应度函数

适应度函数是引导 GP-SMA 算法向最优解逼近的关键因素。由于个体的选择是基于各自染色体的适应度, 染色体的适应度越高, 被选择的机会就越多。

GP-SMA 算法使用了两个适应度函数  $S_1$ ,  $S_2$ 。其中,  $S_1$  评价染色体是否满足任务需求,  $S_2$  则评价染色体所对应的规划方案的优劣。在进行适应度评价的时候, 将具有种群规模  $pS$  个体的总群体分为两个子群体, 两个子群体分别使用两个适应度函数选择个体, 以得到两个新的子群体, 将这两个新的子群体合并从而得到新的总群体。再利用遗传算子对新的总群体作进化操作。

$$\begin{aligned}
 S_1(i) &= \frac{\#(M_{out} \cap P_{out}(i))}{\#(M_{out})} \times 100\% \\
 S_2(i) &= \left( \frac{\sum_{k=1}^{pS} (req(k) - Min(P) + \varepsilon)}{pS \times (req(i) - Min(P) + \varepsilon)} \right)^2
 \end{aligned}$$

其中,  $\#(M_{out} \cap O(i))$  是染色体  $i$  所对应的规划方案  $i$  所满足的需求特征的数目,  $\#(M_{out})$  是总的需求特征数,  $req(i)$  是染色体  $i$  所对应的规划方案的代价,  $Min(P)$  是当前群体中代价的最小值,  $\varepsilon$  是一个小的整型常数。

### 2.4 进化操作

在服务环境下, 服务资源的关联结构为服务组合的规划过程提供显示知识, 因而 GP-SMA 可以利用这些显式知识来指导群体的演化。我们用动作空间描述表 (Action Space Description Table, ASDT) 来表示当前服务组合规划过程的服务信息搜索空间。ASDT 是系统服务资源的子集, 采用语义关联和相似性关联进行服务的组织和索引。ASDT 的初始状态为空。GP-SMA 基于显式知识指导的群体演化过程如下:

- 1) 初始化 ASDT;
- 2) 按照 ASDT 产生/更新新个体;
- 3) 基于新个体的增量式学习, 即根据新个体的情况将一些新的  $u_x$  添加到 ASDT 中;
- 4) 根据知识约束修整 ASDT;
- 5) 将遗传算子作用于当前群体;
- 6) 重复以上过程, 直至满足终止条件。

当  $S_1$  未达到 100% 时, 需要根据个体  $p$  的  $P_{out}(p)$  来添加一个  $u_x$ , 要求该新添加  $u_x$  的  $P_{in}(u_x)$  能够被  $M_{in} \cup P_{out}(p)$  所满足。

ASDT 以服务元素分组来规整特征标识, 从而为交叉及变异算子作准备。对于一个特征标识集合  $P_{ax} \subseteq M_{in} \cup P_{out}(p)$ , 在 ASDT 中将可能出现多个  $u$  分组  $\{o_{x1}, o_{x2}, \dots, o_{xn}\}$ , 使得:  $P_{ax} = P_{in}(o_{x1}) = P_{in}(o_{x2}) = \dots = P_{in}(o_{xn})$ 。此时,  $P_{ax}$  将作为一个可能的交叉变异点。

GP-SMA 通过 ASDT 对群体实施进化操作, 即个体根据 ASDT 实施的交叉 (或变异) 点来实现交叉 (或变异) 操作。在此, 以  $P_{GO} = \{i_a, i_c, i_m\}$  表示遗传算子的概率因子向量。相应地,  $GO = \{ao, co, mo\}$  则表示由添加算子、交叉算子、变异算子组成的进化操作向量。

## 3 仿真实验

### 3.1 仿真设计

对上述模型和算法, 我们将每个服务元素作为一个模块

$w, w \in U$ , 每个模块  $w$  的  $P_{in}(w)$  特征标识集和  $P_{out}(w)$  特征标识集通过随机的方式产生。其中,  $P_{in}(w)$  中的特征标识个数随机从  $[1, 7]$  中产生,  $P_{out}(w)$  中的特征标识个数随机从  $[1, 5]$  中产生, 每个特征标识则随机从  $[1, N_f]$  中生成,  $N_f$  为所有特征标识的个数。在仿真实验中, 设定  $N_f = 300$ 。要求在单个模块  $x$  中不存在相同的特征标识。

我们通过仿真实验来比较 GP-SMA 和目前存在的两个服务组合规划算法。其中, RDRP 算法不使用任何索引手段, 而 SCBHP 和 GP-SMA 要求所有服务元素都按照它们之间的关联关系组成一个基于  $P_{in}(w)$  特征标识和  $P_{out}(w)$  特征标识的索引体系。为了简化实验模型, 服务运行的性能只考虑动作元素的计算量开销。GP-SMA 的参数也简单地采用多次试探性的方法, 其中,  $p_s = 30, p_c = 0.2, p_m = 0.05$ 。

### 3.2 结果分析

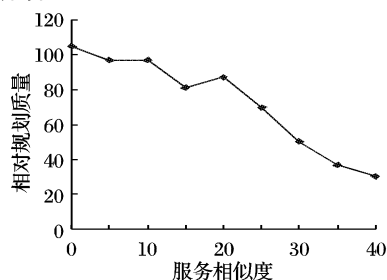


图1 不同相似度的规划质量对比

可见, 服务组合规划的质量设定为规划方案中所有模块的计算量的和。图1为 SCBHP 和 GP-SMA 质量随相似度增加的对比, 即  $q_{GP-SMA}/q_{SCBHP}$ 。由图1可见, 在大尺度的选择空间

中, GP-SMA 算法比 SCBHP 算法具有更好的寻优能力。

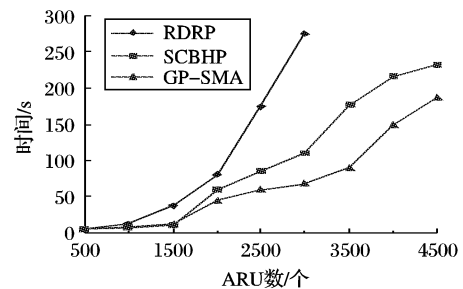


图2 不同动作元素数量规划时间对比

在相似度和服务数量不同的情况下, 服务相似度越高, 组合优化每一步的选择空间越大, 组合规划的时间越长, 而 GP-SMA 算法求到的第一个可行解的时间短。

### 参考文献:

- [1] NARAYANAN S, MCLLRATH S. Simulation, Verification and Automated Composition of Web Service[A]. Proceedings of the 11th International World Wide Web Conference[C]. Hawaii, USA, 2002. 47-54.
- [2] KOZA JR. Genetic Programming: On the Programming of Computers by Means of Natural Selection[M]. Cambridge, Massachusetts: MIT Press, 1992.
- [3] 周明, 孙树栋. 遗传算法原理及应用[M]. 北京: 国防工业出版社, 1999.
- [4] SALTSTOWICZ RP, SCHMIDHUBER J. Probabilistic incremental program evolution[J]. Evolutionary Computation, 1997, 5(2): 123-141.
- [5] BENATALLAH B, DUMAS M, SHENG QZ. The SELF-SERV Environment for Web Service Composition[J]. IEEE Internet Computing, 2003, 8(2): 40-48.

(上接第 1681 页)

```
integer, title: string)
director ( directorid: integer, director. parentid: integer, director.
parentCODE: integer, director. name. isroot: boolean, director. name.
firstname. isroot: Boolean, director. name. firstname: string, director.
name. lastname. isroot: Boolean, director. name. lastname: string,
director. address. isroot: Boolean, director. address: string, director.
directorid: string)
```

### 2.5 基本嵌入技术与共享嵌入技术的定性评价

评价的标准是查询处理效率, 具体表现为长度为  $N$  路径表达式的 SQL 连接平均数。因为路径表达是 XML 数据查询语言的核心, 用从 Robin Cover 的 SGML/XML Web<sup>[8]</sup> 页上得到的 37 个 DTD 来定性评价我们的转换法则, 制定了如下度量: 1) 长度为  $N$  的路径表达式产生的 SQL 查询的平均数; 2) 用于长度为  $N$  的路径表达式的每一个 SQL 查询的连接平均数; 3) 为了处理长度为  $N$  的路径表达式所需要的总的连接平均数。

使用基本嵌入技术, 37 个 DTD 中有 5 个因为用用虚存而转换没有完成。造成这种情况的原因是: 如果 DTD 含有大量关联紧密的成分, 基本嵌入技术将产生数目繁多的关系。而使用共享嵌入技术为 5 个 DTD 中的 4 个消除了大量的连接, 对于另外 1 个 DTD 产生和基本嵌入技术几乎相同数量的连接。这说明, 我们的改进技术的适用范围是有限的, 需要进行进一步的改进。

## 3 结语

我们研究了使用传统关系模型来处理符合一定模式的 XML 文档实时查询的方法。这种方法的潜在优势很多: 可以复用成熟的技术, 利用现有的高性能系统, 并实现对 XML 文档或关系数据的无缝实时查询。通过实验证实, 除了某些类型的复杂递归查询, 用关系数据库可以处理大多数基于 XML

文档的查询。

我们的工作表明, 如果关系数据库系统具有下列扩展, 可以更有效地处理 XML 实时查询: 集合操作的支持、无类型/可变类型引用、信息查询引擎、灵活的比较运算符。传统的关系模型不支持此类比较, 如果不同的 DTD 能够支持不同的类型, 就能够有效提高实时查询处理能力。

### 参考文献:

- [1] BOSAK J, BRAY T, CONNOLLY D, et al. W3C XML Specification DTD[EB/OL]. <http://www.w3.org/XML/1998/06/xmlspec-report-19980910.htm>, 1998.
- [2] CHRISTOPHIDES V, ABITEBOUL S, CLUET S, et al. From Structured Documents to Novel Query Facilities[A]. Proceedings of the ACM SIGMOD Conference[C]. Minneapolis, Minnesota, 1994.
- [3] FERNANDEZ DM, FLORESCU D, LEVY A, et al. XML - QL: A Query Language for XML[EB/OL]. <http://www.w3.org/TR/NOTE-xml-ql>, 1998.
- [4] ABITEBOUL S, QUASS D, MCHUGH J, et al. The Lorel Query Language for Semistructured Data[J]. International Journal on Digital Libraries, 1997, 1(1): 68-88.
- [5] FERNANDEZ DM, SUCIU D. Storing Semistructured Data with STORED[A]. Proceedings of the ACM SIGMOD Conference[C]. Philadelphia, Pennsylvania, 1999.
- [6] Oracle Corporation. XML Support in Oracle 8 and beyond[R/OL]. <http://www.oracle.com/xml/documents>, 2006.
- [7] BRAY T, PAOLI J, SPERBERG - MCQUEEN CM. Extensible Markup Language (XML) 1.0[S/OL]. <http://www.w3.org/TR/REC-xml>, 2004.
- [8] COVER R. The SGML/XML Web Page[EB/OL]. <http://www.oasis-open.org/cover/xml.html>, 2006.