

基于可达路径保障的服务组合方法^①

张鹏^② 丁嘉宁 李焱 杨嵘 刘庆云

(信息内容安全技术国家工程实验室 北京 100093)

(中国科学院信息工程研究所 北京 100093)

(国家计算机网络应急技术处理协调中心 北京 100029)

摘要 针对服务组合逻辑的不确定性需求,本文提出了基于可达路径保障的服务组合方法(FPRSCS),该方法通过一种基于服务约束关系的可达保障机制帮助用户快速逼近目标进行问题求解。试验表明,FPRSCS方法和语义匹配方法相比,可使到达目标的流程案例的比例提高73.1%。

关键词 服务组合,可达路径,临机逻辑,预置逻辑

0 引言

在科研协作过程中,科学实验因其具有创新性和不确定性的特点,导致科研工作者难以预先定义完备的实验流程,只能在实验过程中根据前一阶段的实验结果进行分析判断,确定下一步的实验内容,也常常需要根据结果来反复修改实验参数观察不同的结果,直到达到预期的试验效果。然而,当前科研协作领域的相关研究工作如e-Science[1], Science Grid[2]等项目主要致力于支持科学活动的网络基础设施构建上,对于在环境和用户需求的不确定性情况下,如何支持科研工作者快速、灵活的自行组装网络资源、搭建个性化网络协同应用以及引入人的动态决策的流程开环控制问题没有给与足够的关注。在这方面,HAN等人提出通过用户自主编程和系统智能辅助相结合来组合服务,在不断探索中采用逐渐逼近最终解的方式来进行问题求解[3]。其中流程一致性验证是关键。流程一致性验证可分为语法一致性和语义一致性两类。语法一致性与所应用的场景无

关,它是一致性的最基本条件。典型的语法上的错误包括死锁、参数不匹配等[4]。而语义一致性与所应用的场景有关,需要关于应用场景的领域知识才能判断语义一致性[5],在这方面,有人提出了3种基于语义的行为匹配关系进行Web服务进行运行时验证方法[6]。此外,邓水光等人QoS最优的语义Web服务自动组合方法,该方法基于语义以最优组合QoS为约束条件构造所有可能的最优解[7]。同时,常亮等人提出基于动态描述逻辑的语义Web服务组合,将语义Web服务组合问题建模为基于动态描述逻辑的规划问题[8]。

1 案例分析

下面以图1所示的生物信息领域的场景—家蚕基因组组装试验为例来进行具体分析。

某分子生物学科研人员对新获取的样本进行测序后,需要对得到的一组测序数据做基因组拼接试验,以分析此样本的生物学特性。由于以前从未进行过此类物种的基因研究,没有可以直接使用的分析流程,科研人员自行设计了试验过程。

^① 863计划(2011AA010703),国家自然科学基金(61402464)中国博士后基金(2013M541076)资助

^② 通讯作者,张鹏,1984年生,男,博士,主要研究领域为服务计算,流数据处理,云计算; E-mail:pengzhang@iie.ac.cn

在试验过程中，科研人员经常需要通过观察前一阶段的实验结果来决定下一步做什么。例如，在图 2 中当 2-25 轮的组装结果进行连接后，科研人员发现许多克隆重叠群 (contig) 都不能连接上，分析其原因可能是日本和中国的家蚕存在的个体差异所导致的，故增加“个体差异分析”这一步骤。在试验过程中，科研人员经常需要对试验过程进行修改，例如增加、删除或者更新活动节点，用户的变更操作可能导致出现语义冲突。尤其是

科研人员在探索过程中，可能偏离预期的目标，他们希望系统能从领域专家成功（或者失败）的试验过程中抽取出某些经验和知识，提供智能建议，以帮助他们逐步逼近目标进行问题求解。为此，本文针对服务组合逻辑的不确定性，提出了基于可达路径保障的服务组方法—FPRSCS，该方法通过基于服务约束关系的保障机制帮助用户快速逼近目标进行问题求解。

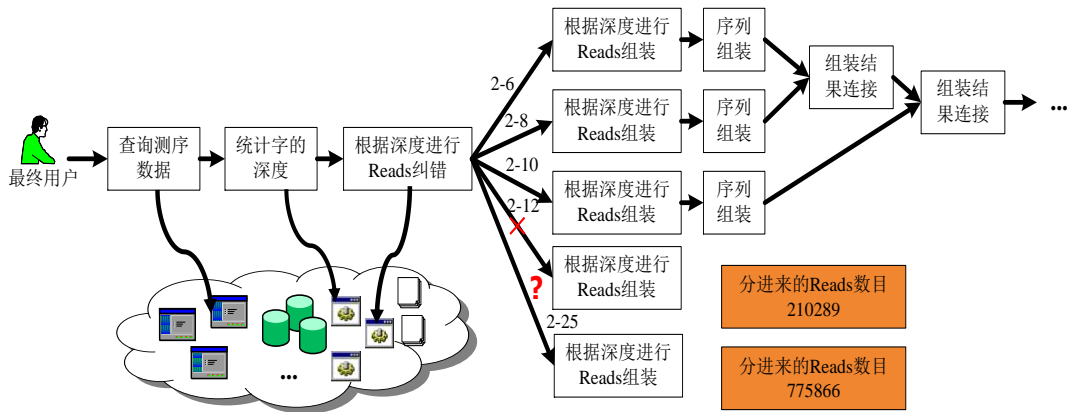


图 1.家蚕基因组组装流程图（前半部分）

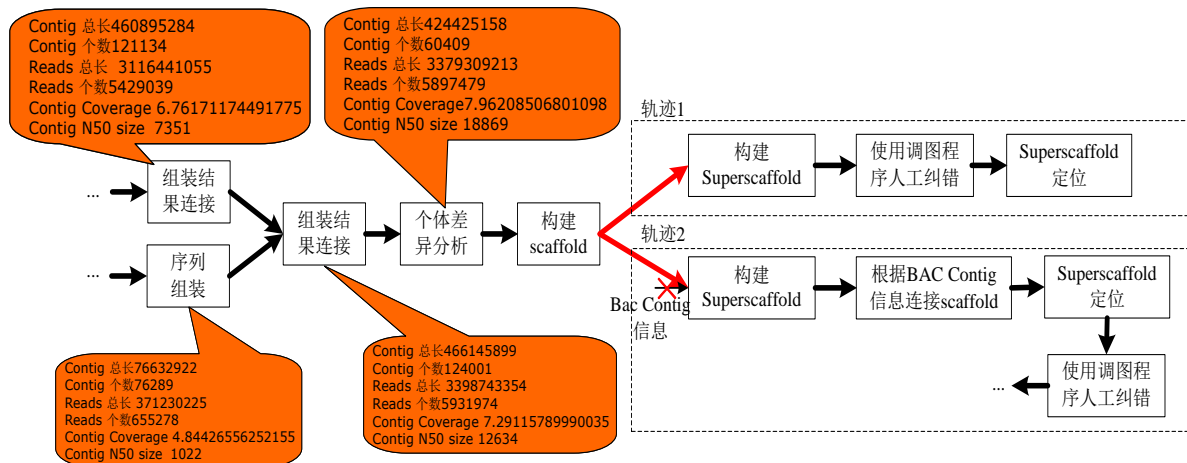


图 2.家蚕基因组组装流程图（后半部分）

2 FPRSCS 方法的基本原理

针对网络化科研协作项目中的实际问题，我们在 VINCA 方法学[3][9]的基础上，研究一种可达路径保障的服务组方法(FPRSCS)，其基本思路如图 3 所示。该方法支持服务组合逻辑不确定

情况下用户服务组合需求的表达，灵活便捷以增量的方式构造应用，同时以服务约束关系为基础，为用户提供一致性验证、保障到达目标的支持。主要包括两个层：用户探索层和系统支撑层。各类箭头对应可达路径保障的服务组合需要的主要

操作，其中空心箭头表示用户的“编程”操作，实心箭头表示编程环境的支撑操作。

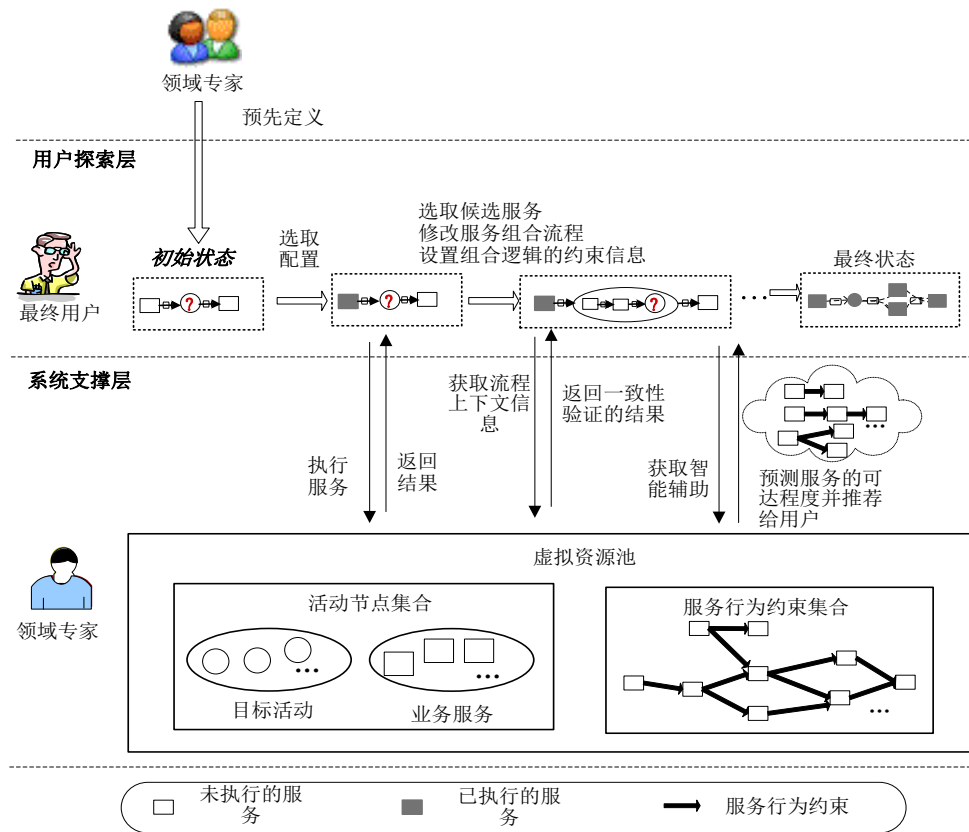


图 3. FPRSCS 方法的基本思路

1) 用户探索层：支持最终用户参与灵活的服务组合，应对服务组合过程中的参数、活动以及活动间逻辑关系的不确定性问题。主要包括以下两方面工作：

- 提出了抽象服务组合模板，支持用户事先定义骨架流程以及个性化的约束信息表达确定的服务组合需求；通过使用目标活动节点来表达不确定的需求。最终用户可以使用领域专家预先定义好的服务组合模板，也可以自己定义新的模板。

- 通过对目标活动节点的扩展和重编排来支持即时的流程构建和动态调整。在节点扩展方面，提供对目标活动节点的变更、控制操作，支持用户在目标活动节点内以“边构造边执行”的方式进行流程的构建。在节点重编排方面，通过引入“调试点”及其添加和删除操作，支持用户

对目标活动节点内已执行的部分流程，以增量的方式进行参数调整和流程结构调整，从而减少数据的冗余，提高流程构建的效率。

2) 系统支撑层：提供对用户的灵活探索过程在目标路径可达方面的保障。包括以下三方面工作：

- 对服务行为约束关系进行分析建模，有序化表示服务间的约束关系，并将其作为一种领域知识，为可达保障提供基础支撑。

- 实时检测用户构建的不完备流程和已有约束间的一致性，发现存在潜在语义冲突的约束集合，为用户提供编程指导，保证用户编程的正确性。

- 动态预测单步候选服务到达目标的程度，对其进行排序并推荐给用户，以帮助其快速逼近目标进行问题求解。

FPRSCS 在传统的工作流模型[10]的基础上通过引入目标活动节点对其进行扩充，其中包括预置逻辑的构建和临机逻辑的构建两个阶段。预置逻辑是指领域专家事先根据用户需求制定包含不确定因素的服务组合的骨架流程（即抽象服务组合模板），最终用户在目标活动节点内定义的关键点和关键边约束以及领域业务约束。临机逻辑是指部分不确定的服务组合逻辑在运行阶段由最终用户进行即时构建。为了帮助用户快速逼近目标进行问题求解，临机逻辑和预置逻辑需要保证一致，下面给出相关定义。

3 临机逻辑和预置逻辑一致性保障

首先给出服务间一元和二元约束关系的表示。一元约束关系主要包括两个部分：发生模式和模式的有效范围。二元约束关系主要包括两个部分：发生模式和顺序模式。发生模式：“always exist directly”：表示业务服务 a 总是紧邻业务服务 b 出现。“always exist”：一元约束关系中表示业务服务 a 总在一定范围内出现；二元约束关系中表示业务服务 a 总是在业务服务 b 的一定范围内出现，但不一定紧邻出现。“sometimes exist directly”：表示业务服务 a 可能紧邻业务服务 b 出现。“sometimes exist”：一元约束关系中表示业务服务 a 可能在一定范围内出现；二元约束关系中表示业务服务 a 可能在业务服务 b 的一定范围内出现，但不一定紧邻出现。“always absent”：表示业务服务 a 在一定范围内不出现。

顺序模式：“before”：表示发生模式一直保持到业务服务 b 出现。“after”：表示发生模式一直到业务服务 b 出现后才会发生。

模式的范围：“globally”：表示发生模式在系统执行的整个过程中都必须为真。

定义 1(临机逻辑)：临机逻辑是指运行阶段某一时刻的服务组合流程实例，包括节点和变迁关系等信息。记为 π 。

定义 2(预置逻辑)：预置逻辑是指用户目标约束(包括领域专家制定的抽象服务组合模板和最终用户在目标活动节点内定义的关键点和关键边约束)以及领域业务约束两部分，可以看作是服务行为约束模型的一个子集。

定义 3(临机逻辑和用户目标约束的一致性)：如果临机逻辑 π 和用户需求目标的强制性约束 C_{Mg} 是无冲突的，则称 π 是满足目标约束一致性的，记为 $f(\pi, C_{Mg}) = true$ 。假设 $C_M = C_P \cup C_E$ ，其中 C_P 表示关键点的约束， C_E 表示关键边的约束。 S 表示 π 中的结点集合， T 表示 π 中边的集合，则临机逻辑 π 同时满足以下条件时，称 π 是用户目标约束一致的： $p_i = \text{always-exist-globally}(a_m)$ ， $\exists b_j \in S, \forall p_i \in C_P$ 使得 $b_j = a_m$ 。 $\forall x_i \in C_E, \exists t_j \in T$ ，使得 $t_j = x_i$

定义 4(临机逻辑和领域业务约束的一致性)：如果临机逻辑 π 和领域业务约束中强制性的约束 C_{Mb} 是无冲突的，则称 π 是满足领域业务约束一致性的，记为 $f(\pi, C_{Mb}) = true$ 。若 S 表示流程 π 中的节点集合， T 表示流程 π 中边的集合，则 $\forall s_i \in S$ ，以 s_i 为源或者目标节点的强制性约束集合记为 C_{Mb} 。则临机逻辑 π 满足以下条件时，称 π 是领域业务约束一致的：对于 $\forall a, b \in S$ ，需要满足下列原子约束：

如果在 C_{Mb} 中 a always-exist-directly-before b ，则 $(a, b) \in T$ ，且从 a 到边 b 的长度为 1，记为 $|(a, b)| = 1$ 。

如果在 C_{Mb} 中 a always-exist-directly-after b 则 $(b, a) \in T$ ，且从 b 到 a 边的长度为 1，记为 $|(b, a)| = 1$ 。

如果在 C_{Mb} 中 a always-exist-before b 则 $(a, b) \in T$ ，且从 a 到 b 边的长度大于 1，记为 $|(a, b)| \geq 1$ 。

如果在 C_{Mb} 中 a always-exist-after b 则

$(b,a) \in T$, 且从 b 到 a 边的长度大于 1, 记为 $|b,a| \geq 1$ 。

如果在 C_{Mb} 中 a always-absent-before b 并且 $S' \subseteq S$, S' 为 b 结点的前驱节点集合, 则 $a \notin S'$ 。

如果在 C_{Mb} 中 a always-absent-after b 并且 $S' \subseteq S$, S' 为 b 结点的后继节点集合, 则 $a \notin S'$ 。

下面借鉴曲线拟合以及流程相似性度量方面的思想, 给出临机逻辑和预置逻辑一致性验证结果的量化定义。

定义 5(用户目标约束的拟合度): 表示临机逻辑 π 和强制性用户目标约束 C_g 的一致程度, 可表示为:

$$f(\pi, C_g) = t_{11} \times \frac{X_M}{|P_M|} + t_{12} \times \frac{Y_M}{|\prec_M|}$$

其中 $|P_M|$ 表示必须满足的关键点的数量, 而 X_M 表示当前流程轨迹中从开始节点可达的路径中包含的关键点的数量。 $|\prec_M|$ 表示必须满足的关键边的数量, 而 Y_M 表示当前流程轨迹中从开始节点可达的路径中包含的关键边的数量。 t_{11} 和 t_{12} 是权值。

满足约束: $t_{11} > 0, t_{12} > 0, t_{11} + t_{12} = 1$ 。 如果 $f(\pi, C_g) = 1$, 则临机逻辑 π 和用户目标约束 C_g 是一致的。

定义 6(领域业务约束的拟合度): 表示临机逻辑 π 和强制性领域业务约束 C_b 的一致程度, 可以表示为:

$$f(\pi, C_b) = t_{21} \times \frac{M_1}{|N_1|} + t_{22} \times \frac{M_2}{|N_2|} + t_{23} \times \frac{M_3}{|N_3|}$$

其中 $|N_1|$ 表示 C_b 中 always-exist-directly-after 和 always-exist-after 类型的服务行为约束的数量, 而 M_1 表示临机逻辑中满足 N_1 约束的数量。 $|N_2|$ 表示 C_b 中 always-exist-directly-before 和 always-exist-before 类型的服务行为约束的数量, 而 M_2 表示临机逻辑中满足 N_2 约束的数量。 $|N_3|$ 表示 C_b 中 always-absent-before 和 always-absent-after 类型的服务行为约束的数量, 而 M_3 表示临机逻辑中满足 N_3 约束的数量。 t_{21} , t_{22} 和 t_{23} 是权值, 满足约束: $t_{23} > t_{22} > t_{21} > 0, t_{21} + t_{22} + t_{23} = 1$ 。

定义 7(目标距离): 目标距离 d 描述了临机逻辑和预置逻辑一致程度, 形式化表示为 $d(\pi, C_g \cup C_b) = 1 - (w_1 \times f(\pi, C_g) + w_2 \times f(\pi, C_b))$

其中权重 w_1 和 w_2 满足约束: $w_1 > 0, w_2 > 0, w_1 + w_2 = 1$ 为了计算临机逻辑向预置逻辑的逼近程度, 在定义 7 中目标距离定义的基础上, 给出了可达路径等相关定义。

定义 8(可达路径): 对于临机逻辑中任意目标活动节点 g , 假设 g 的直接前驱节点为 a , g 的直接后继节点为 b , 则若能找到一条从 a 到 b , 且满足 g 上用户目标约束以及领域业务约束的路径, 就称之为 g 的可达路径。

定义 9(可达路径集): 同一个目标活动节点的多个可达路径的集合。

4 基于可达路径集的路径推荐算法

基于上述定义, 下面给出了基于可达路径集的推荐算法。该算法主要包括以下两个步骤:

步骤 1: 对于某个给定的服务节点, 先定位可达路径中距离其较近且能使其快速到达目标的前

驱节点和后继节点，称之为前驱到达节点和后继到达节点。为了定位给定节点在可达路径中的前驱到达节点和后继到达节点，我们利用 *Dijkstra* 算法。该算法用于求从给定当前节点到可达路径上距离其较近且能使其快速到达目标的前驱节点和后继节点。其最差情况下的时间复杂度为 $O(n^2)$ ，最好情况下的时间复杂度为 $O(1)$ 。

步骤 2: 基于前驱到达节点和后继到达节点预测用户单步探索过程中的候选服务的推荐程度。并据此对结果进行排序，推荐给用户。候选服务的推荐程度预测主要包括以下几种的情况：

1: 假设当前节点 x_0 的前驱到达节点为 S_{pre} ，后继到达节点为 S_{post} 。从当前可达路径的初始节点 s_0 到 S_{pre} 的部分流程表示为 pt_{pre} ，从 s_0 到 S_{post} 的部分流程表示为 pt_{post} ，则得到目标距离值 $d(pt_{pre}, g)$ 和 $d(pt_{post}, g)$ 。

2: 假设 S_{pre} 到 x_0 的路径权值为 x ， x_0 到 S_{post} 的路径的权值为 y ， x_0 所在的当前流程轨迹的开始节点到 x_0 的路径权值为 z ， S_{pre} 到可达路径的结束节点的路径权值为 w ，那么：

2.1: 若 S_{pre} 和 S_{post} 都存在，则可根据如下公式得到 x_0 处的目标函数值：

$$d(pt_{post}, g) + (d(pt_{pre}, g) - d(pt_{post}, g)) \times \frac{y}{x + y}$$

2.2: 若 S_{pre} 不存在， S_{post} 存在，则可根据如下公式得到 x_0 处的目标函数值：

$$d(pt_{post}, g) + (1 - d(pt_{post}, g)) \times \frac{y}{z}$$

2.3: 若 S_{post} 不存在， S_{pre} 存在，则可根据如下公式得到 x_0 处的目标函数值：

$$d(pt_{pre}, g) - d(pt_{pre}, g) \times \frac{x}{w}$$

2.4: 若 S_{pre} 和 S_{post} 都不存在，则 x_0 处的目标函数值为 1。

5 评价与分析

本文通过 PROPOLS 语言[11]来检查不完备的服务组合和 PROPOLS 约束语句之间的一致性，并且在 PROPOLS 的基础上，通过 FPRSCS 帮助用户快速逼近目标进行问题求解。为了证明 FPRSCS 的效果，我们进行如下实验。

5.1 实验环境

实验平台采用 Intel Pentium IV CPU 2.4GHz，内存 2G 的 PC 机；操作系统为 Windows XP；软件开发包 JDK1.5；开发工具为 Eclipse。

5.2 实验数据

由于目前并不存在标准的服务组合测试平台和数据集。实验中我们从目前收集到生物信息领域序列比对、基因组装、进化树分析三个方面的 26 个流程案例信息进行分析，如附录 A 所示。这些流程案例涵盖了 MyGrid、DDBJ、巴塞罗那 CNS 生命科学部门，马拉加生物信息技术实验室，北京华大基因几个不同机构的流程案例。

5.3 实验评测

为了评估每步推荐时，有和无可达保障对用户编排质量的影响，我们对上述 26 个流程案例进行了试验。针对每个案例，在不考虑用户自定义目标约束的情况下将我们的方法(FPRSCS 方法)和基于语义匹配进行单步推荐的方法进行对比。不妨假设两种方法在领域本体构造、语义服务注册方面完全相同。语义匹配方法在每步根据服务语义匹配来给用户推荐服务，FPRSCS 方法根据服务行为约束集合来推荐服务。不妨假设对于两种方

法用户每步都选取推荐中优先级最高的服务。

下面采用目标距离作为指标，来评估随着应用程序步长的不断变化生成的服务组合流程和已知的流程案例之间的吻合情况。当前的应用程序中每增加一个可达的服务节点，我们就认为步长加 1。为了计算目标距离，先将已知流程中的点和边转换成关键点和关键边，然后利用定义 7 中目标距离的公式来计算。如果经过若干步后，若目标距离能达到 0，则流程到达目标。

实验中，选取了 5 家机构的 26 个生物信息方面的流程作为实验样本，其中巴塞罗那 CNS 生命科学部门(8 个)、MyGrid(6 个)、DDBJ(4 个)、马

拉加生物信息技术实验室(5 个)，北京华大基因(3 个)。使用上述两种方法所获取的实验结果如表 1 所示。从表中可以看出，采用 FPRSCS 方法比语义匹配方法到达目标的流程案例的比例提高了 73.1%。在 26 个流程案例中使用语义匹配方法仅有 5 个案例可以到达用户想要的目标，其它 21 个都不能到达目标，而采用 FPRSCS 方法有 24 个流程可以到达用户想要的目标，其它 2 个不能到达目标。通过分析我们发现这 2 个案例在实验的服务行为约束集中没有可达的路径，无法生成相应的可达路径。

表 1 语义匹配方法和 FPRSCS 方法对用户编排质量的影响

	流程案例的总个数	到达目标的流程案例个数	没有到达目标的流程案例个数	到达目标的流程案例个数占总流程案例个数的比值
语义匹配方法	26	5	21	19.2%
FPRSCS 方法	26	24	2	92.3%

我们对语义匹配方法无法到达目标的 21 个流程案例进行了分析，发现主要存在四类典型的原因。**原因 1:** 当后继服务中无语义匹配的服务时，只考虑语义匹配将导致探索过程提前中止。对于附录中的流程案例 25，在 Retrieve biological sequence 之后选取语义匹配度最高的 Split ace，接着选取语义匹配度最高的 Consed ace(步长为 2)，然而 Consed ace 后面没有与之存在语义匹配的服务，所以提前中止。**原因 2:** 只考虑语义匹配无法发现服务之间的控制依赖关系。对于附录中的流程案例 26，目标是建立进化树分析试验，在 Retrieve similiar sequences 之后选取语义匹配度最高的 Generate bootstrap sample(步长为 4)，然而其后没有与之存在语义匹配的服务，所以中止。而我们通过流程案例挖掘可以发现从 Generate bootstrap sample 到 Compute distance matrix from

nucleotide sequences 的控制依赖关系。**原因 3:** 只考虑语义匹配导致背离用户的目标。对于附录中的流程案例 9，在 Blastn(步长为 1)之后选取语义匹配度最高的 Blast，其后不断选取语义匹配度高的服务添加进来，但是却一直处于和目标偏离的一个稳定状态。**原因 4:** 只考虑语义匹配导致循环查找。对于附录中的流程案例 24，在 Assemble repeat reads(步长为 2)之后选取语义匹配度最高的 Merge ace，而和 Merge ace 的语义匹配度最高的是 Assemble unique reads，而和 Assemble unique reads 的语义匹配度最高的是 Retrieve unique reads 服务，该服务同时又是 Assemble repeat reads 的前驱节点，造成循环查找。

6 结论

针对服务组合逻辑的不确定性，本文提出了一种基于可达路径保障的服务组合方法 FPRSCS，

支持灵活的服务组合。在路径推荐方面，FPRSCS 通过基于服务关系约束的保障机制帮助用户快速

逼近目标进行问题求解。试验表明了 FPRSCS 的效果。

参考文献

1. U. Schwiegelshohn, R. M. Badia, M. Bubak et al, "Perspectives on grid computing", Future Generation Computer System, 2010, 26(8): 1104-1115.
2. K Gross, S Hayashi, S Teige et al, "Open Science Grid (OSG) Ticket Synchronization: Keeping Your Home Field Advantage In A Distributed Environment" Journal of Physics: Conference Series, 2012, 396(6): 062009
3. Y. Han, J. Wang, P. Zhang. "Business-oriented service modeling: A case study", Simulation Modelling Practice and Theory, 2009,17(8): 1413—1429
4. J. Chen, Y. Yang, "Temporal dependency-based checkpoint selection for dynamic verification of temporal constraints in scientific workflow systems", ACM Transactions on Software Engineering and Methodology, 2011,20(3):9
5. D. Borrego, R. Eshuis, M. T. Gomez-Lopez, R. M. Gasca, "Diagnosing Correctness of Semantic Workflow Models", Journal of Data and Knowledge Engineering, 2013
6. 邓水光, 黄龙涛, 吴斌等. 一种 QoS 最优的语义 Web 服务自动组合方法. 计算机学报, 2013, 36(5): 1015-1030.
7. 张亚红. 基于运行时验证的Web服务选择方法研究. 硕士学位论文, 新疆大学, 2013.
8. 常亮, 刘进, 古天龙, 等. 基于动态描述逻辑的语义 Web 服务组合. 计算机学报, 2013, 36(12): 2468-2478.
9. P. Zhang, J. Wang, X Zhao, Business Services Instantiation on the Basis of the Generalization-Specification Patterns, Chinese Journal of Computers, 2008, 31(8):1320-1330
10. <http://www.wfmc.org/xpdl.html>
11. J. Yu, Y. Han, J. Han, et al, "Synthesizing Service Composition Models on the Basis of Temporal Business Rules", Journal of Computer Science and Technology, 2008. 23(6): 885-894.

A service composition strategy based on feasible path recommendation

Zhang Peng, Ding Jianing, Li Yan, Yang Rong, Liu Qingyun

(National Engineering Laboratory for Information Security Technologies, Beijing 100090)

(Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100090)

(National Computer network Emergency Response technical Team, Beijing 100029)

Abstract

A feasible path recommendation-based service composition strategy (FPRSCS) is proposed to adapt to the uncertainty and innovativeness of service composition. This strategy utilizes a feasible mechanism based on services constrained relation to help users to solve problems rapidly. The experiments indicate that the percentage of feasible cases was raised up to 73.1% by FPRSCS compared with semantic matching strategy.

Key words: service composition, feasible path, temporary logic, preset logic

附录 A：实验样本——生物信息方面的流程案例

序号	工作流名称	提供商	地址
1	blast-px	巴塞罗那 CNS 生命科学部 门(bsc.es)	<a href="http://inb.bsc.es/services/document
ation/moreinfo_wf.php">http://inb.bsc.es/services/document ation/moreinfo_wf.php

2	bl2seq	bsc.es	
3	phyl2	bsc.es	
4	clustalw-phylip	bsc.es	
5	fastaAA	bsc.es	
6	fastaDNA	bsc.es	
7	wublast2seq	bsc.es	
8	blast-n	bsc.es	
9	Phylogeny Bootstrap Analysis	MyGrid 的 MyExperiment 工具(MyExperiment)	http://www.myexperiment.org/
10	Genome annotation pipeline demonstrator workflow for Nucleic Acids Research	MyExperiment	
11	BlastagainstPDBatDDBJ	MyGrid	http://workflows.mygrid.org.uk/repository/myGrid/
12	blast_simplifier	MyGrid	
13	BlastComparer	MyGrid	
14	Williams-PartA	MyGrid	
15	Homology workflow	DDBJ	http://xml.nig.ac.jp/workflow/index.html
16	Blast-ClustalW workflow	DDBJ	
17	BLAST workflow	DDBJ	
18	E.coli bacteria genome workflow	DDBJ	
19	Homology_search_and_Phylogenetic_tree	马拉加生物信息技术实验室(uma.es)	http://chirimoyo.ac.uma.es/gnv5/modules/help/Workflows/workflow_help.cgi
20	Clustalw_Phylip	uma.es	
21	DNA_Clustalw_Phylip	uma.es	
22	AAseq_Clustalw_Phylip	uma.es	
23	FastaAA	uma.es	
24	Bombyx.mori genome sequence assembly experiment	北京华大基因研究所(bgi)	实际调研
25	Rice genome reassembly	bgi	
26	Phylogenetic tree Analysis	bgi	