

[19] 中华人民共和国国家知识产权局



[12] 发明专利申请公布说明书

[21] 申请号 200810054066.3

[43] 公开日 2009 年 12 月 16 日

[51] Int. Cl.
H04L 29/08 (2006.01)
H04L 29/06 (2006.01)

[11] 公开号 CN 101605141A

[22] 申请日 2008.8.5

[21] 申请号 200810054066.3

[71] 申请人 天津大学

地址 300072 天津市南开区卫津路 92 号

[72] 发明人 冯志勇 陈世展 王 辉 陈 炬
韩 冷

[74] 专利代理机构 天津市北洋有限责任专利代理
事务所
代理人 李素兰

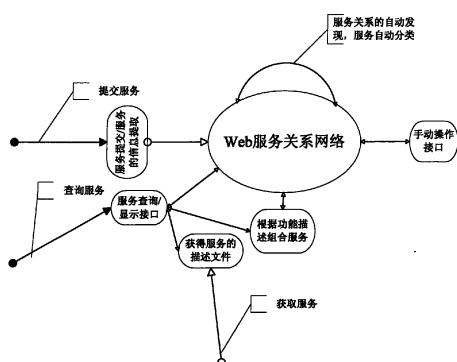
权利要求书 2 页 说明书 9 页 附图 6 页

[54] 发明名称

基于语义的 Web 服务关系网络系统

[57] 摘要

本发明公开了一种基于语义的 Web 服务关系网络系统，Web 服务关系网络用于基于语义的服务发现、(半)自动服务组合。该 Web 服务关系网络所要处理的服务来源于该网络的提交服务和该网络的获取服务，提交服务被动式接收服务注册请求，提取信息送到 Web 服务关系网络；获取服务则包括主动利用网络爬虫技术获得服务的描述文件并通过服务注册接口加入到 Web 服务关系网络；根据用户的功能描述所自动获得的复合服务，复合服务通过服务注册送到 Web 服务关系网络。本发明更加方便 Web 服务的发布、基于语义的服务发现、(半)自动组合，并且对多种服务描述语言能够扩充兼容；将可用的 Web 服务组织成服务生态系统，借助于服务之间的关系改进服务的自动组合、查找、维护等操作。



1. 一种基于语义的 Web 服务关系网络系统，将服务作为该网络的节点，与服务之间的关系组成的三维立体网络，具体又分为两层：抽象服务层和具体服务层，分别包括具体服务和抽象服务，其特征在于，该系统包括 Web 服务关系网络、该网络的提交服务、该网络的获取服务、该网络的根据功能描述所自动组合得到复合服务以及服务发现/显示接口，其中：

Web 服务关系网络用于自动发现、动态组合，该 Web 服务关系网络所要处理的服务来源于该网络的注册服务和发现服务，其中，注册服务包括服务提交/服务的信息提取，该提取信息送到 Web 服务关系网络；发现服务则通过爬虫获得服务的描述文件并注册到 Web 服务关系网络；以及根据功能描述自动进行服务组合，这些复合服务也是通过服务注册接口送到 Web 服务关系网络；该 Web 服务关系网络提供的查询/显示服务功能，通过服务发现接口来完成。

2. 如权利要求 1 所述的基于语义的 Web 服务关系网络系统，其特征在于，所述具体服务是抽象服务的实例，基于语义的 Web 服务关系网络中的抽象服务与各具体服务之间通过 Instance-of 连接。

3. 如权利要求 1 所述的基于语义的 Web 服务关系网络系统，其特征在于，所述具体服务之间，由以下定义的关系相连：等价型服务关系、替换型服务关系、相似型服务关系、组合型服务关系、调用型服务关系、时间型服务关系。

4. 如权利要求 1 所述的基于语义的 Web 服务关系网络系统，其特征在于，所述各节点的结构按照对应选取的 OWL-S 文件和 WSDL 文件中 Web 服务属性描述部分的常用子集进行定义，同时各对应的节点提供 URI 属性指向原先的 OWL-S 文件或者 WSDL 文件。

5. 如权利要求 4 所述的基于语义的 Web 服务关系网络系统，其特征在于，所述 WSDL 文件利用 WSDL2SN 解析器进行解析后，生成 Web 服务关系网络节点，进而组成 Web 服务关系网络系统，所述 WSDL2SN 的解析器实现流程，包括以下步骤：

首先，通过 WSDL4J API 把 WSDL 文件读到 definition 中；

从 definition 中读取得到类型 types 定义；

在 types 定义中构建命名空间等属性信息，并将该属性转换成 JDOM 型；

得到完成转换的类型的 schema 定义。

对于 WSDL 文件中的参数为自定义的复杂类型的情况，需要使用上面得到的 schema 去进行复杂类型的解析，直至将自定义类型分解，得到一系列的简单类型，因此，在上述流程之后还进一步包括以下步骤：

首先对 service 元素进行解析，得到绑定信息；

根据绑定信息找到对应的 portType 信息；

对 portType 中所包含的所有操作 operation 进行解析，得到每个操作的基本属性

和参数信息，每个输入和输出结果都分别是一个 message；

根据参数信息找到相应的 message 定义；

message 的具体结构从已经构建好的 schema 中得到，直至将 message 分解为简单类型；

对绑定类型进行判断，进一步分为 RPC 型和编码型；

当绑定类型为 RPC 型时，构建 RPC 类型复杂参数；

判断所构建的 RPC 类型复杂参数是否为复杂类型；

得到完整的服务信息；

当绑定类型为编码型时，构建文档类型复杂参数；

判断所构建的文档类型复杂参数是否为复杂类型；

得到完整的服务信息。

6. 如权利要求 4 所述的基于语义的 Web 服务关系网络系统，其特征在于，所述 OWL-S 文件利用 OWL-S 解析器进行解析，具体包括以下步骤：

首先通过 OWL-S 的 URI，读取 OWL-S 文件；

导入 OWL-S 引用的其他本体；

检查该文件描述的本体是否有效及符合规范；

如果上述本体的检查结果是有效及符合规范，则解析出组成服务网络中服务节点所需要的相关内容；

映射到符合服务网络定义的规范；

将结果持久化到服务网络中，并通知相应的维护程序；

将更新通知服务网络维护进程；

如果上述本体的检查结果是无效或不符合规范，则输出相关信息，并且结束此次操作。

7. 如权利要求 1 所述的基于语义的 Web 服务关系网络系统，其特征在于，对所述 Web 服务关系网络中的服务关系采用的服务关系挖掘算法，以及该算法进一步包括的操作接口关系的计算流程，具体步骤如下：

首先，对两个服务进行预处理，提取出服务的功能描述标签（tag）和服务的操作接口；然后分别计算两组 tag 和两组服务操作接口的关系；再按照一定权值加权得到服务关系；根据所得的服务关系更新服务网络，算法结束。

在服务操作接口的计算流程中，首先，对两个操作接口进行预处理；判断这两个操作接口的接口名和描述信息是否为反义，如果是，则算法流程结束；如果不是，则进行输入/输出参数关系的计算；然后得到操作接口的关系类型和相似度，算法结束。

8. 如权利要求 1 所述的基于语义的 Web 服务关系网络系统，其特征在于，所述系统还包括手动操作接口。

基于语义的 Web 服务关系网络系统

技术领域

本发明涉及描述和发现服务的基于因特网的网络系统，特别是涉及基于 Web 应用的服务信息组织方式，例如注册、管理、访问机制。

背景技术

目前，Web 服务注册中心往往采用类 UDDI 标准，使用 XML 文档来描述企业及其提供的 Web 服务，并维护 Web 服务的全球目录。这种基于 UDDI 或其变体的 Web 服务注册模型因缺少对 Web 服务之间关系的语义描述，存在如下弊端：

- 1) 注册表中仅仅存放 Web 服务的名称、领域等基本信息，只能基于关键字进行服务的发现、匹配，导致 web 服务的查全率和查准率不高；
- 2) 缺乏对 Web 服务语义属性的描述信息，造成在面向业务流程集成的应用中，无法通过 Web 服务的语义属性特征实现服务查询；
- 3) 缺乏对 Web 服务间关系的描述信息，无法提供更具灵活性和可用性的服务分类机制；
- 4) 侧重于功能性要求的服务发现算法，无法满足 QoS 的要求；
- 5) 缺乏对 Web 服务自动组合的进一步支持。

目前，许多研究致力于修正现有 UDDI 注册模型的缺陷，比如利用本体技术来提高查全率，利用哈希表来提高查找效率，但是都没有从根本上解决当前 Web 服务注册模型所面临的上述主要问题。

发明内容

鉴于上述现有技术中存在的问题，本发明提出了一种基于语义的 web 服务关系网络系统，构建基于语义和关系的 Web 服务网络，本发明通过对现有的 Web 服务注册中心中的数据、以及对各种离散的 Web 服务信息进行重新的归纳、整理，转换成富有语义的本体形式；利用挖掘算法对 Web 服务间的关系进行挖掘，将 Web 服务注册中心组织成一个以 Web 服务为节点、以服务间的关系为边的网络系统，从而实现动态、自主、基于语义的服务发现和（半）自动的服务组合。

本发明提出了一种基于语义的 Web 服务关系网络系统，将 Web 服务作为该网络的节

点，与服务之间的关系组成的三维立体网络，具体又分为两层：抽象服务层和具体服务层，分别包括具体服务和抽象服务，其特征在于，该系统包括 Web 服务关系网络、该网络的服务提交系统和服务发现系统、该网络的根据功能描述（半）自动产生组合服务系统以及服务查询/显示接口，其中：

Web 服务关系网络用于服务自动发现、服务（半）自动组合，该 Web 服务关系网络所要处理的服务来源于该网络的提交服务和该网络的获取服务，其中，提交服务包括服务提交/服务的信息提取，该提取信息送到 Web 服务关系网络；获取服务则将获得服务的描述文件通过服务查询/显示接口送到 Web 服务关系网络；以及根据功能描述所获得的组合服务，这些组合服务也是通过服务查询/显示接口送到 Web 服务关系网络；该 Web 服务关系网络提供的查询服务功能，通过服务查询/显示接口来完成。

所述具体服务是抽象服务的实例，基于语义的 Web 服务关系网络中的抽象服务和具体服务之间通过 Instance-of 连接。

所述具体服务之间，由以下定义的关系相连，该定义的关系包括：等价型服务关系、替换型服务关系、相似型服务关系、组合型服务关系、调用型服务关系、时间型服务关系、位置型服务关系。

所述各节点的结构按照对应选取的 OWL-S 文件和 WSDL 文件中 web 服务属性描述部分的常用子集进行定义，同时各对应的节点提供 URI 属性指向原先的 OWL-S 文件或者 WSDL 文件。

所述 WSDL 文件利用 WSDL2SN 解析器进行解析后，生成 Web 服务关系网络节点，进而组成 Web 服务关系网络系统，所述 WSDL2SN 的解析器实现流程，包括以下步骤：

首先，通过 WSDL4J API 把 WSDL 文件读到 definition 中；

从 definition 中读取得到类型 types 定义；

在 types 定义中构建命名空间等属性信息，并将该属性转换成 JDOM 型；

得到完成转换的类型的 schema 定义。

对于 WSDL 文件中的参数为自定义的复杂类型的情况，需要使用上面得到的 schema 去进行复杂类型的解析，直至将自定义类型分解，得到一系列的简单类型，因此，在上述流程之后还进一步包括以下步骤：

首先对 service 元素进行解析，得到绑定信息；

根据绑定信息找到对应的 portType 信息；

对 portType 中所包含的所有操作（operation）进行解析，得到每个操作的基本属性和参数信息，每个输入和输出结果都分别是一个 message；

根据参数信息找到相应的 message 定义；

message 的具体结构从已经构建好的 schema 中得到，直至将 message 分解为简单类型；

对绑定类型进行判断，进一步分为 RPC 型和编码型；

当绑定类型为 RPC 型时，构建 RPC 类型复杂参数；

判断所构建的 RPC 类型复杂参数是否为复杂类型；

得到完整的服务信息；

当绑定类型为编码型时，构建文档类型复杂参数；

判断所构建的文档类型复杂参数是否为复杂类型；

得到完整的服务信息。

所述 OWL-S 文件利用 OWL-S 解析器进行解析，具体包括以下步骤：

首先通过 OWL-S 的 URI，读取 OWL-S 文件；

导入 OWL-S 引用的其他本体；

检查该文件描述的本体是否有效及符合规范；

如果上述本体的检查结果是有效及符合规范，则解析出组成服务网络中服务节点所需要的的相关内容；

映射到符合服务网络定义的规范；

将结果持久化到服务网络中，并通知相应的维护程序；

将更新通知服务网络维护进程；

如果上述本体的检查结果是无效及不符合规范，则输出相关信息，并且此次操作结束。

对所述 Web 服务关系网络中的服务关系采用的相应服务关系挖掘算法，具体步骤如下：

首先，对两个服务进行预处理，提取出服务的功能描述标签（tag）和服务的操作接口；然后分别计算两组 tag 和两组服务操作接口的关系；再按照一定权值加权得到服务关系；根据服务关系更新服务网络，算法结束。

其中该算法进一步包括服务操作接口关系的计算流程，具体步骤如下：

首先，对两个操作接口进行预处理；判断这两个操作接口的接口名和描述信息是否为反义，如果是，则算法流程结束；如果不是，则进行输入/输出参数关系的计算；然后得到操作接口的关系类型和相似度，算法结束。

所述系统还包括手动操作接口。

与现有技术相比，本发明更加方便 Web 服务的发现、组合、查找匹配，并且对多种服务语义描述语言能够实现扩充兼容，使用更加方便；将可用 Web 服务组织成服务生态系统（Services ecosystem），借助于服务之间的关系改进服务的组合、查找、维护等操作。

附图说明

图 1 为本发明的基于语义的 Web 服务关系网络形成的示意图；

图 2 为本发明的基于语义的 Web 服务关系网络系统示意图；

-
- 图 3 为对各种 Web 服务模式转换器不同描述方式的转换流程图；
图 4 为适用于 WSDL 文件的 WSDL2SN 解析器的实现流程图；
图 5 为适用于 OWL-S 文件的 OWL-S2SN 解析器的实现流程图；
图 6 为本发明的基于语义的 web 服务关系网络系统的整体框架图；
图 7 为本发明的服务关系挖掘算法的操作接口关系计算流程图；
图 8 为本发明的服务关系挖掘算法流程图。

具体实施方式

如图 1 所示为本发明的基于语义的 Web 服务关系网络形成的示意图。服务网络系统首先从传统的无语义的 Web 服务注册中心取得许多独立的 Web 服务，通过对每个服务结构，语义的分析，挖掘出各个服务之间存在的各种关系，形成具体服务网络层，同时也将各个 Web 服务的功能进行泛化，抽象出共性，从而形成抽象服务网络层。由此构成的整体就是本发明的基于语义的 Web 服务关系网络。

如图 2 所示，为本发明的基于语义的 Web 服务关系网络系统示意图。该服务关系网络系统将服务作为该网络的节点，与服务之间的关系组成的三维立体网络。具体又分为两层：抽象服务层和具体服务层，分别包括具体服务和抽象服务。该服务关系网络中的服务关系主要包括以下七种类型：

- 1、等价型服务关系：该类型服务的接口属性（包括输入/输出接口）完全一致，所实现功能完全一致，可以相互替换；并且，这种关系的成立与服务的具体实现无关；
- 2、替换型服务关系：该类型服务之间存在的是种有向的替换关系，例如服务 A 实现的功能，服务 B 全部能够实现，则服务 B 可替换服务 A（反之不一定成立）；
- 3、相似型服务关系：例如，（单向）服务 S 与服务 T 存在功能上的部分重叠（如图 2 中所示：相似度 0.6），则称之为服务 B 相似于服务 A，服务 A 相似于服务 B，但两者相似程度上可能有差异。（服务 A 对服务 B 的相似度，可能与服务 B 对服务 A 的相似度不同）；
- 4、组合型服务关系：服务 P 的实现由服务 A₁, A₂, A₃…等组成，则服务 A_i 到服务 P 之间存在组合关系。关于组合关系，首先应考虑到组合的顺序，即服务的组成结构如何，调用顺序如何，需要考虑将这些关于组成的控制信息存放在何处，以及需要考虑如何获得这些组成的控制信息。这些调用顺序可以有以下几种：顺序调用，循环调用，分支调用等等。OWL-S 标准支持以上几种控制结构，比如：Sequence, Split, Split+Join, Unordered, Choice, If-Then-Else, Iterate 及 Repeat-Until 等；
- 5、调用型服务关系：服务 B₁ 调用了服务 E₁，则服务 B₁ 到服务 E₁ 存在调用关系，调用关系常常跟组合关系一起存在；
- 6、时间型服务关系：指不同服务在发生时间方面的先后次序关系。比如，服务 P

必须先与服务 Q 执行。

该服务关系网络系统中的具体服务是抽象服务的实例，之间通过 Instance-of 连接。具体服务之间，还可以由上面定义的关系相连。

其中，服务节点的属性参数包括：

- 接口属性：IOPE，即 Input, Output, Precondition, Effect。
- 功能描述（是标签或者分类）：采用关键字形式描述服务完成的功能以及服务的特点，并以此作为服务的分类依据，有相同关键字的服务成为一类。
- 服务的提供者信息：比如：姓名，联系方式（电话、E-mail）。当用户只能使用此服务，而此服务又不可用，那么用户可以联系服务的提供者，通过协商使服务可用。
- URI：服务的地址。
- D-URI：描述文件所在的 URI。
- 组合服务属性：如为原子服务则用“atomic”做属性值，若为组合服务则用 cmp 做为属性值。
- 权限信息：username/password。对于付费服务，可能会需要。
- 服务创建时间：（每次的修改时间）。
- 服务质量：包括稳定性、可靠性、服务成本及信誉度。稳定性（Stability）用于描述同一 Web 服务在不同的调用时刻所需的响应时间的差异；响应时间（Response Time）指服务从请求者发出执行请求开始到收到应答消息所经过的时间；可靠性（Reliability）表示能够维护服务和服务质量的程度；服务成本（Service Cost）描述服务消费者需要为使用服务而付出的费用；信誉度（Grade）用于描述服务使用者对他们所调用的 Web 服务的评价。

在本发明的基于语义的 Web 服务关系网络系统中，各节点的结构按照对应选取的 OWL-S 文件和 WSDL 文件中 web service 属性描述部分的常用子集进行定义，同时各对应的节点提供 URI 属性指向原先的 OWL-S 文件或者 WSDL 文件。采用这种做法即可以保证 web service 结构的简洁，清晰，同时也保证了信息的完整性，容易实现。

本发明主要采用对服务的本体描述，来表示 Web 服务关系网络。

由于本发明的服务关系网络与现有技术的各种 Web 服务描述方式不一致，因此定义了一种较为简略又不丢失原描述文件所包含的信息的节点结构，以保证该服务关系网络系统对现有技术中的各种不同 Web 服务描述方式的兼容。该处理需要首先对不同的描述方式进行解析，解析流程如图 3 所示。该解析流程包括以下步骤：

对于 WSDL 文件，利用 WSDL2SN 解析器进行解析后，生成 Web 服务关系网络节点，进而组成 Web 服务关系网络系统；

对于 OWL-S 文件，经过 OWL-S2SN 解析器的解析处理后，生成 Web 服务关系网络节

点，进而组成 Web 服务关系网络系统。

WSDL2SN 的解析器实现流程，该流程如图 4 所示步骤 401~步骤 404，该流程包括以下步骤：

首先，通过 WSDL4J API 把 WSDL 文件读到 definition 中，步骤 401；从 definition 中读取得到类型 types 定义（DOM 型），步骤 402；在 types 定义中构建命名空间等属性信息，并将该属性转换成 JDOM 型，步骤 403；得到完成转换的类型的 schema 定义，步骤 404；该 schema 定义在面对自定义的参数类型进行解析时需要用到。

对于 WSDL 文件中的参数为自定义的复杂类型的情况，需要使用上面得到的 Schema 去进行复杂类型的解析，直至将自定义类型分解，得到一系列的简单类型。这部分流程如图 4 所示步骤 405~步骤 415，该流程包括以下步骤：

步骤 405：首先对 service 元素进行解析，得到绑定信息，即：解析至<wsdl:service>标签，找到每个<wsdl:port>标签中的 binding 后面的绑定信息；

步骤 406：根据绑定信息找到对应的 portType 信息，即：到<wsdl:binding>标签中去找 portType 属性，这就是该服务所提供的主要的接口名称。

步骤 407：对 portType 中所包含的所有操作（operation）进行解析，得到每个操作的基本属性和参数信息，即：到<wsdl:portType>标签中去找每一个操作，<wsdl:operation>每一个操作又分别对应有自己的输入和输出，每个输入和输出结果都分别是一个 message；

步骤 408：根据参数信息找到相应的 message 定义，即：到<wsdl:message>中去找每个 message 的基本组成，它可能由简单类型和复杂的自定义类型组成。如果是简单类型则解析过程终止；

步骤 409：Message 的具体结构可以从已经构建好的 schema 中得到，直至将 message 分解为简单类型，即：如果 message 的组成是复杂的自定义类型，则需要递归地到类型定义中去找对这种复杂类型的定义，直至构成类型完全为简单类型；

步骤 410：对绑定类型进行判断，进一步分为 RPC 型和编码型；

步骤 411：当绑定类型为 RPC 型时，构建 RPC 类型复杂参数；

步骤 412：判断所构建的 RPC 类型复杂参数是否为复杂类型；

步骤 415：得到完整的服务信息；

步骤 413：当绑定类型为编码型时，构建文档类型复杂参数；

步骤 414：判断所构建的文档类型复杂参数是否为复杂类型；

步骤 415：得到完整的服务信息。

如图 5 所示为 OWL-S2SN 解析器的实现流程图，具体包括以下步骤：

步骤 501：首先通过 OWL-S 的 URI，读取 OWL-S 文件，即：获得被解析的 OWL-S 文

件的 URI，读取该文件，并读取其引用的其他本体文件；

步骤 502：导入 OWL-S 引用的其他本体；

步骤 503：检查该文件描述的本体是否有效及符合规范，即是否存在语义上的矛盾，以及是否符合 OWL-S 规范；

步骤 504：如果上述本体的检查结果是有效及符合规范，则解析出组成服务网络中服务节点所需要的相关内容；

步骤 505：映射到符合服务网络定义的规范，即：将解析的结果格式化成符合服务网络规范定义的格式；

步骤 506：将结果持久化到服务网络中，并通知相应的维护程序；

步骤 507：将更新通知服务网络维护进程；

步骤 508：如果上述本体的检查结果是无效或不符合规范，则输出相关信息，并且结束此次操作。

如图 6 所示，为本发明的基于语义的 Web 服务关系网络系统的整体框架图，其中 Web 服务关系网络用于服务自动发现、服务（半）自动组合，该 Web 服务关系网络的所要处理的服务来源于用户提交的服务；Web 服务网络自动产生的组合服务；Web 服务关系网络系统通过机器人程序从 internet 自动搜索，提取出的公开的服务。其中，提交服务包括服务提交/服务的信息提取，该提取信息送到 Web 服务关系网络；Web 服务网络将根据用户的需求（半）自动的产生新的组合服务，并产生出该 Web 服务的描述文件，在将该服务提交给用户的同时也作为一个案例存放在 Web 服务网络中；获取服务则将获得服务的描述文件通过服务添加接口送到 Web 服务关系网络；该 Web 服务关系网络提供的服务查询功能，通过服务查询/显示接口来完成。

该 Web 服务关系网络还包括手动操作接口，用于对 Web 服务关系网络所提供的查询服务、获取服务以及组合服务与 Web 服务关系网络的手动操作。

为了实现本发明的目的，本发明将采用语义网技术构建 Web 服务本体以及设计相应的关系挖掘算法。对于服务关系挖掘来讲，要做的是在普遍的服务世界中计算服务的逻辑关系，需要详细的数值来度量服务之间相似的程度，而服务之间的关系很大程度上依赖于服务操作接口之间的关系。

如图 7 所示，为服务关系挖掘算法的操作接口关系的计算流程，该流程包括以下步骤：

首先，对两个操作接口进行预处理；判断这两个操作接口的接口名和描述信息是否为反义，如果是，则算法流程结束；如果不是，则进行输入/输出参数相似度计算，然后接口相似度建立，得到关系类型和相似度，算法结束。

下面具体介绍本算法中的一些处理步骤的具体含义：

“预处理”模块是对操作接口进行的一些预处理工作，包括对操作接口名的分词和

分析，对描述信息中关键字进行提取，以及对输入输出参数列表的提取。

操作接口名和操作接口的描述性信息只起参考作用，如果不是相反的意思即可不再考虑。

“参数相似度计算”模块用来计算两组参数的相似程度。输入参数列表和输出参数列表的相似度计算方法是一样的，后文统称为参数列表相似度计算。只是在最终计算操作接口关系的时候，输入和输出所占的比重是不一样的，通常情况下，人们更关心输出的匹配情况，因而输出所占的比重要大于输入。

一般情况下，服务的参数列表不只是简单的数据类型，而是具有语义信息的概念，在本文中，参数列表相似度的计算即可以看作是两组概念的相似度计算。

首先要说明一下单独的两个概念之间相似度的计算方法。概念相似度是指本体中两个概念间相似程度的度量。需要说明的是，只有同一个本体中的两个概念才有相似度，不同本体中的概念认为相似度为零（未来可以考虑不同本体中的概念相似度）。本体中概念的相似程度与概念之间的语义距离有关，通过相关的相似度函数即可得到单个概念间的相似度。

两组概念的相似度可以可用加权二部图的最优匹配方法解决。以两组概念分别为两组顶点，操作间的相似度为边的权值（为方便起见，为相似度设置一个阈值，低于这个阈值则认为是零，即不存在这条边）。本发明参照经典的 kuhn-munkres 算法来确定参数列表之间的匹配。

在“服务关系建立”模块中，根据输入、输出概念列表的相似度值和相似属性来最终确定操作接口之间的相似关系。

如图 8 所示，为本发明的操作服务关系计算流程。该流程具体包括以下步骤：

首先，对两个服务进行预处理，对两个服务进行预处理，提取出服务的功能描述标签（tag）和服务的操作接口；然后分别计算两组 tag 和两组服务操作接口的关系；再按照一定权值加权得到服务关系；根据所得的服务关系更新服务网络，算法结束。

下面具体介绍本算法中的一些处理步骤的具体含义：

“预处理”模块是对服务操作接口进行的一些预处理工作，提取出服务的功能描述标签（tag）和服务的操作接口。

在“接口关系计算”模块中，将两组操作接口作为输入，然后两两配对，应用图 7 所示的算法进行计算，得到两组操作接口中两两配对的相似度值和关系类型，再通过寻找二部图的最佳匹配。以两组操作分别为两组顶点，操作间的相似度为边的权值（为方便起见，为相似度设置一个阈值，低于这个阈值则认为是零，即不存在这条边）。从而确定两组操作的相似度和关系类型。这里依旧参照 kuhn-munkres 算法来确定操作接口间的匹配。

在“词组关系计算”模块中，将两组 tag 作为输入，然后两两配对，基于 WordNet

语义词典计算词汇的相似度和关系类型，得到两组 tag 中两两配对的相似度值和关系类型，再通过寻找二部图的最佳匹配。以两组 tag 操作分别为两组顶点，词汇间的相似度为边的权值（为方便起见，为相似度设置一个阈值，低于这个阈值则认为是零，即不存在这条边）。从而确定两组操作的相似度和关系类型。

“服务关系建立”模块通过操作接口关系和 tag 关系，按照一定的权值来确实服务关系，如果服务关系存在，则通过“更新服务网络”模块来更新服务网络，算法结束；否则算法直接结束。

以上内容仅为本发明的实施例，其目的并非用于对本发明所提出的系统的限制，本发明的保护范围以权利要求为准。在不脱离本发明的精神和范围的情况下，本领域技术人员在不偏离本发明的范围和精神的情况下，对其进行的关于形式和细节的种种显而易见的修改或变化均应落在本发明的保护范围之内。

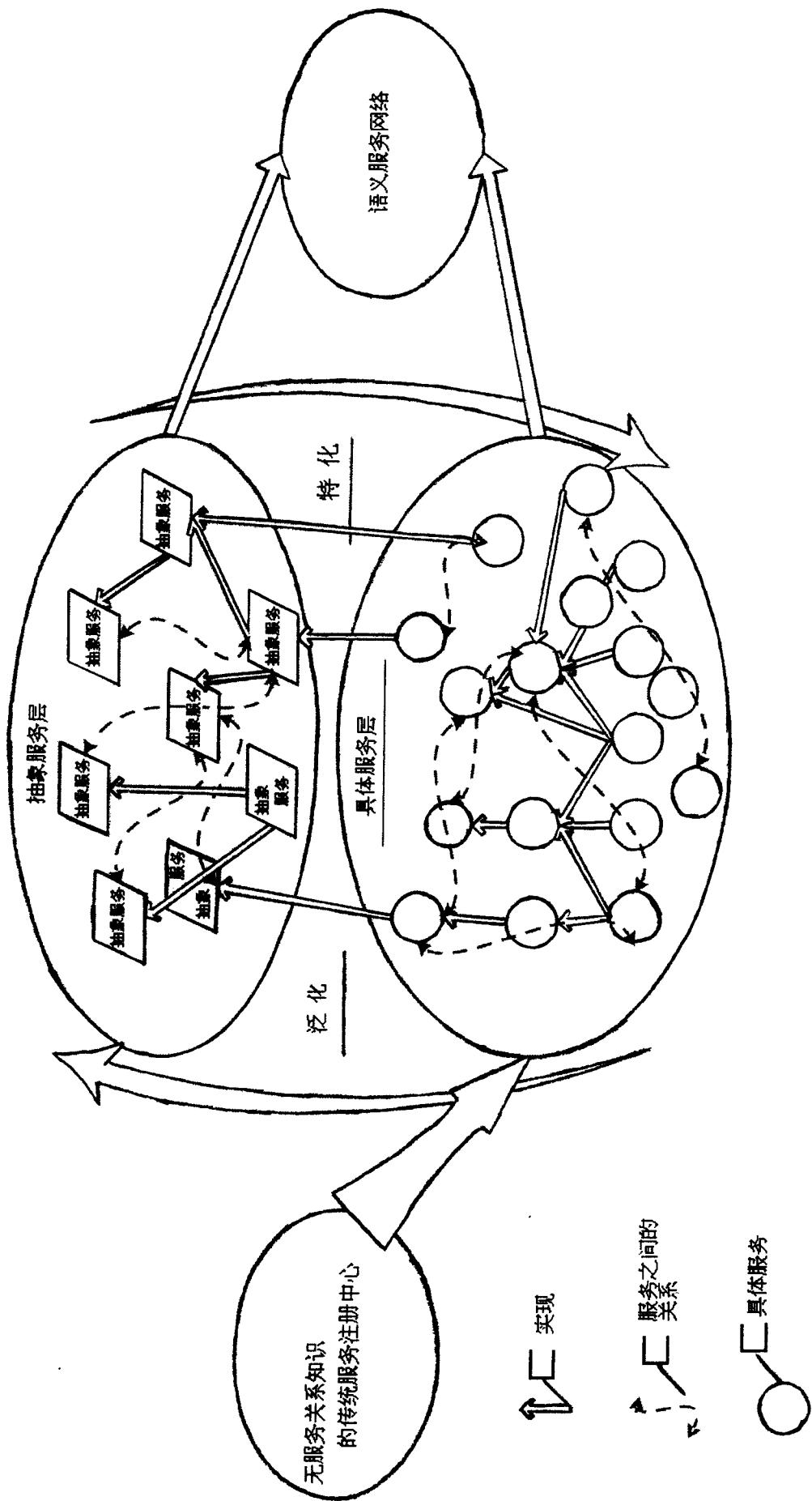


图 1

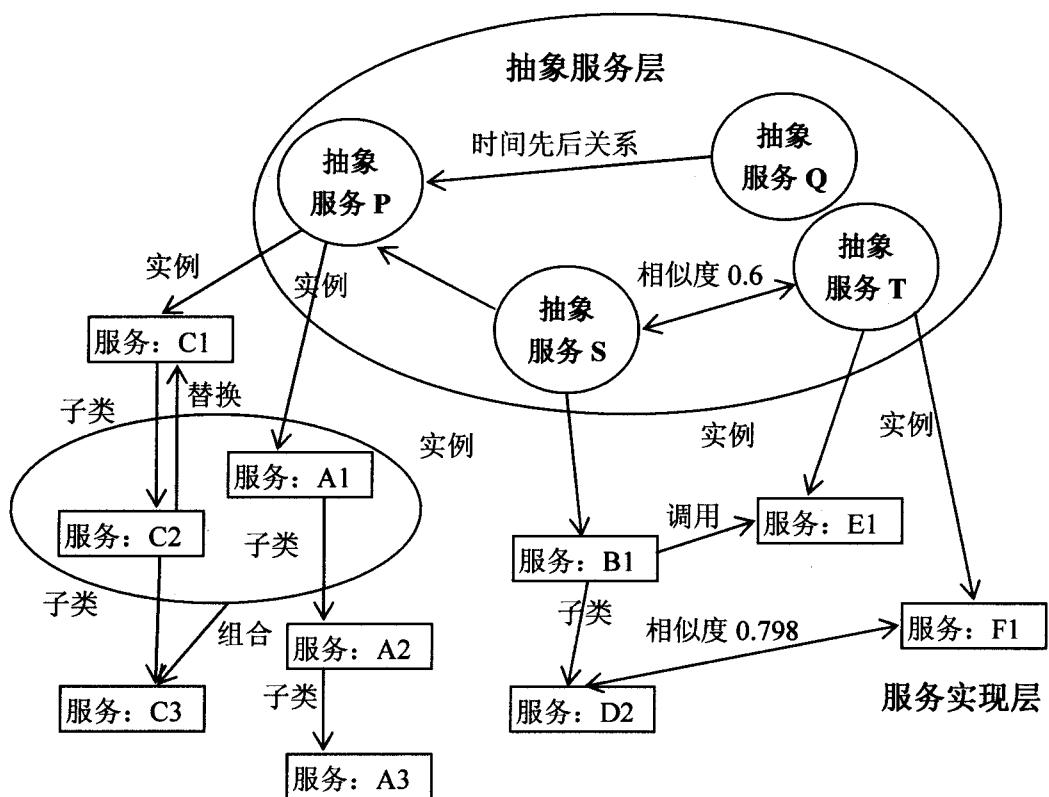


图 2

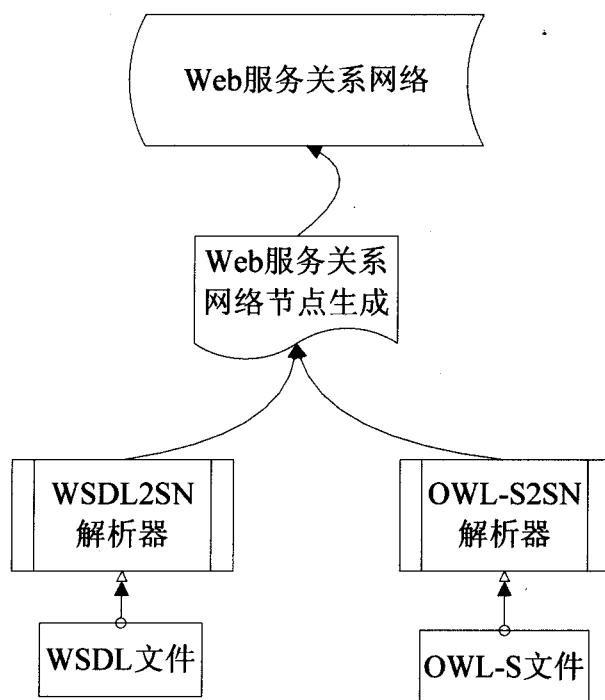


图 3

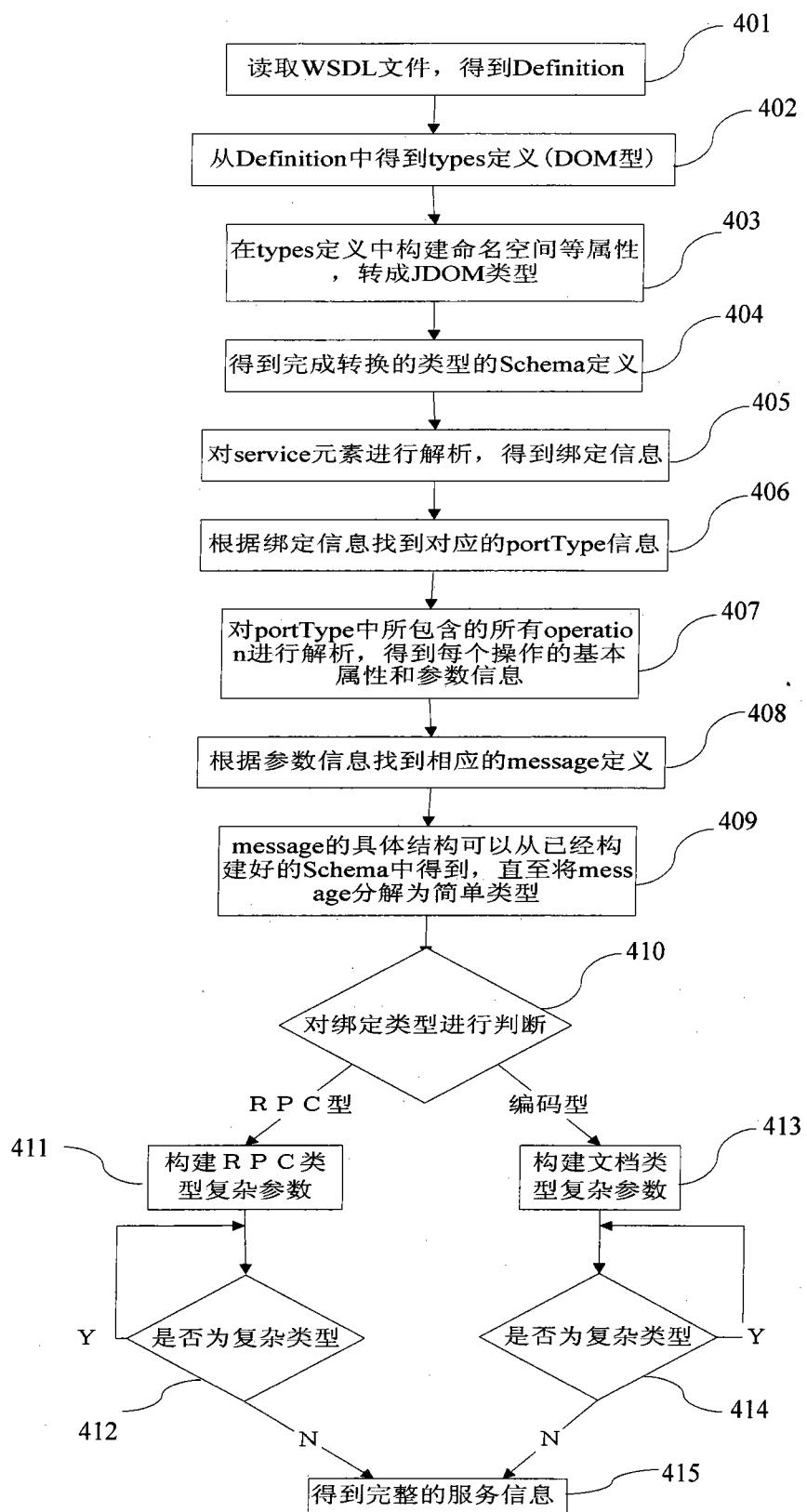


图 4

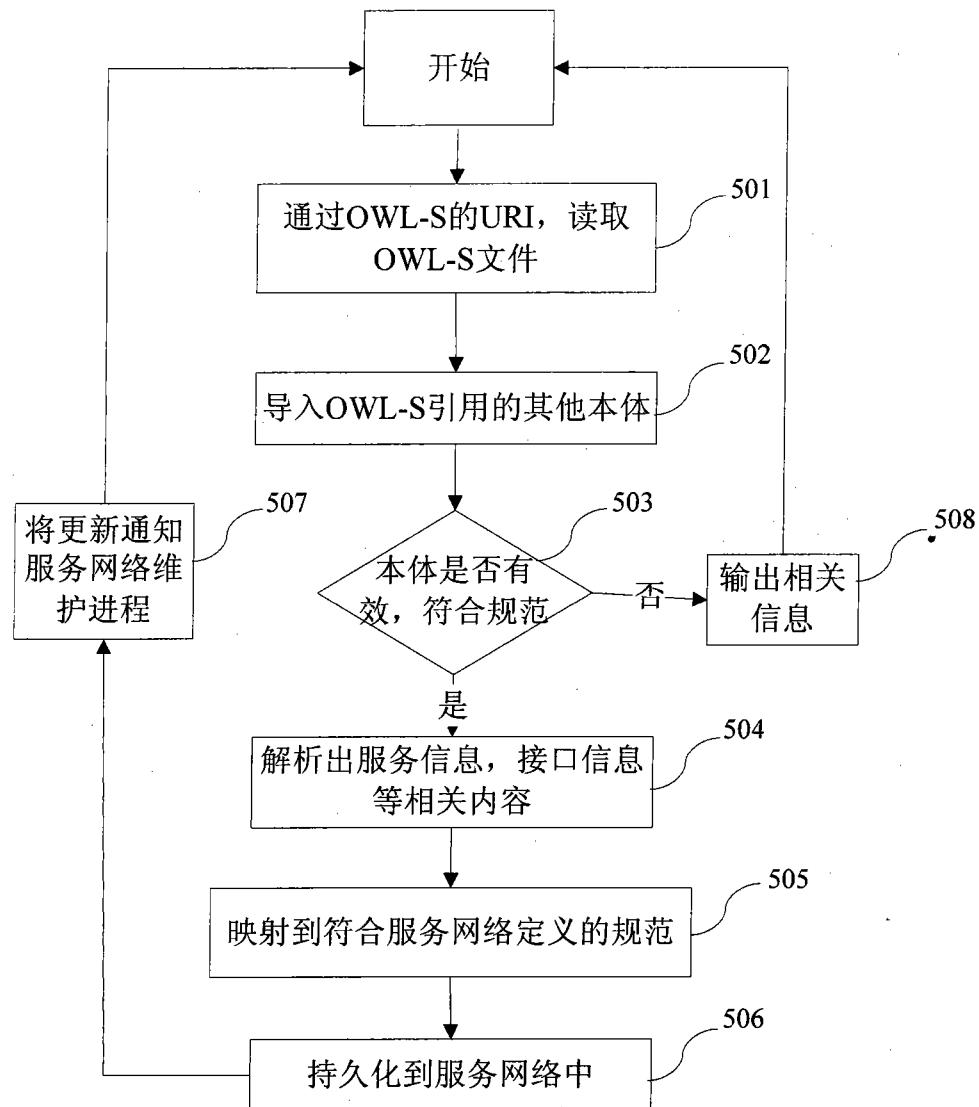


图 5

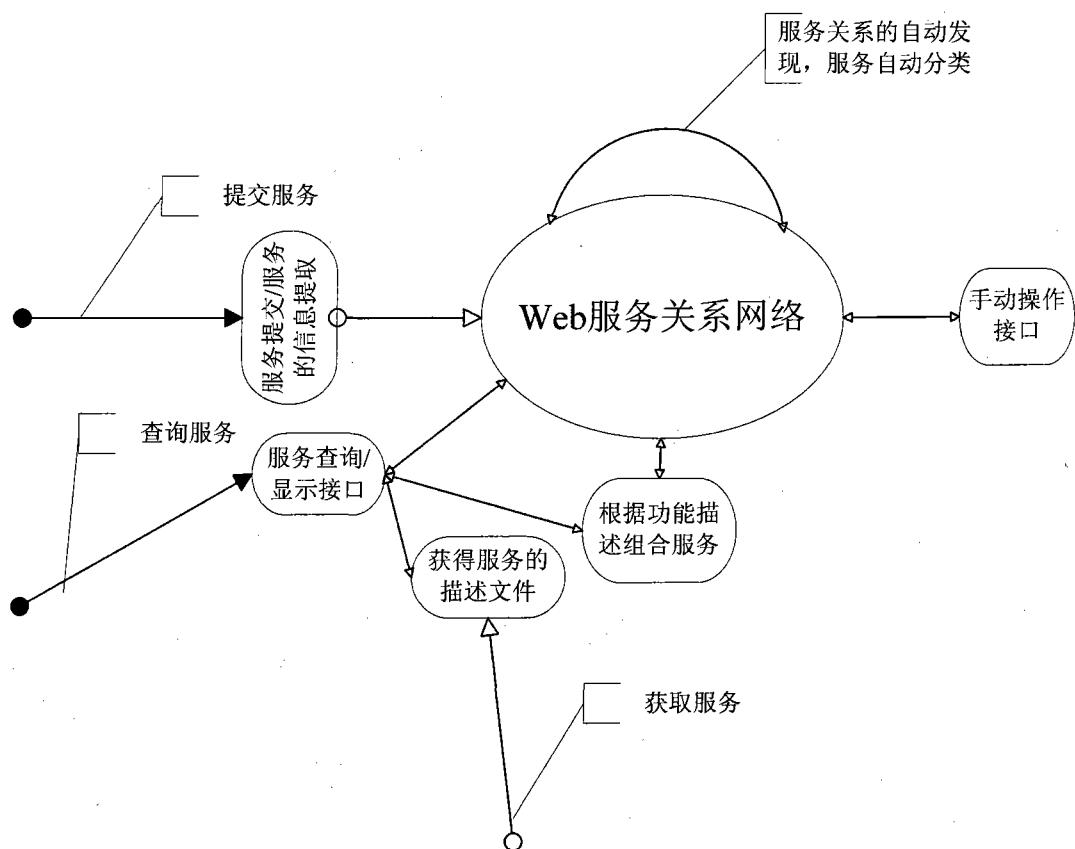


图 6

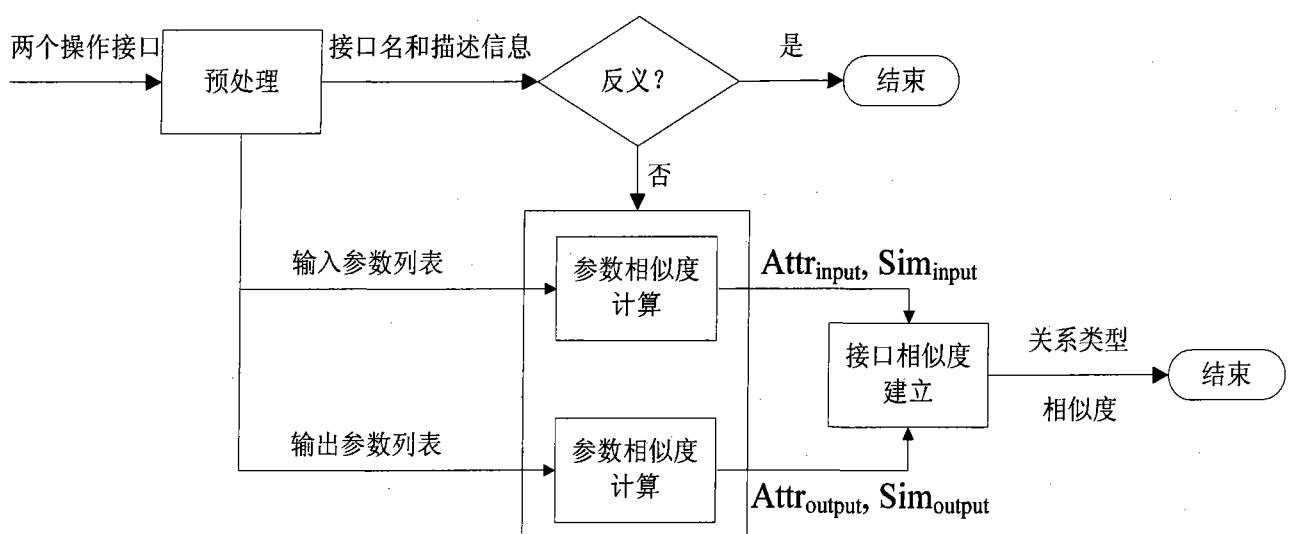


图 7

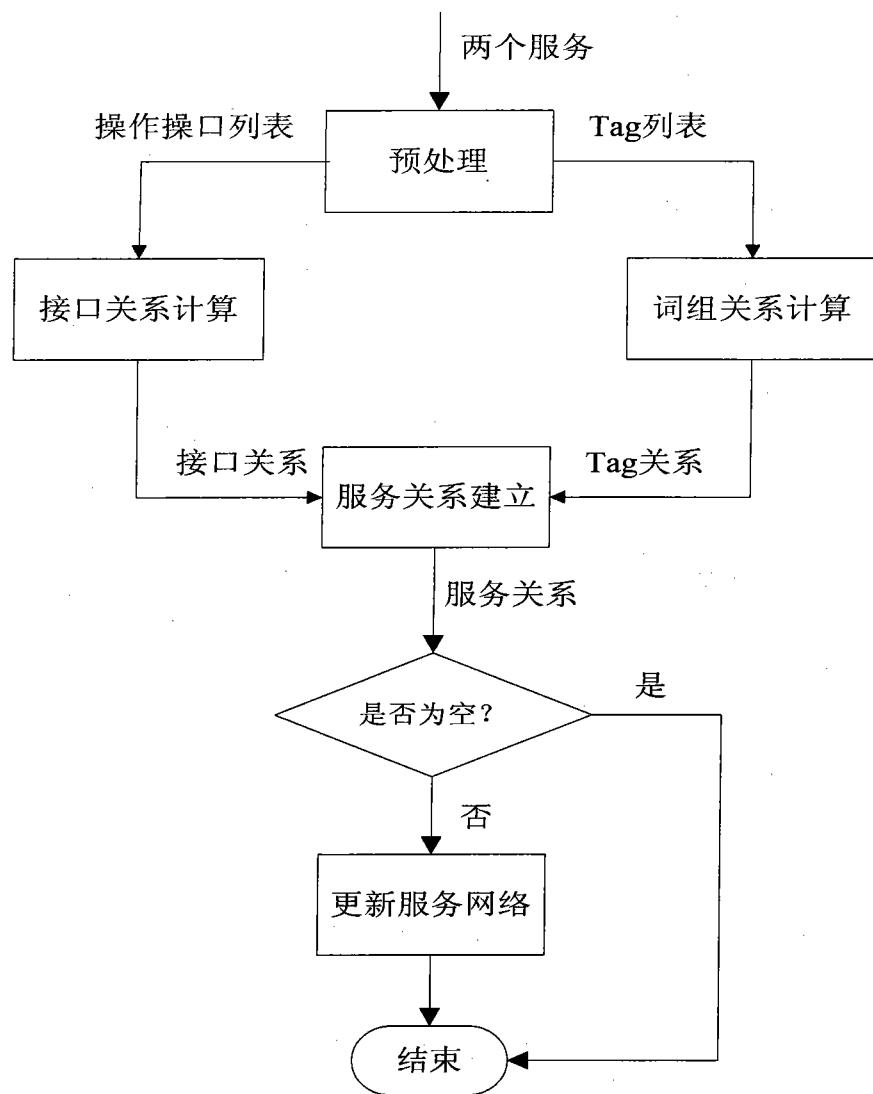


图 8