

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

H04L 29/06 (2006.01)

G06Q 30/00 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200510096291.X

[43] 公开日 2006 年 4 月 26 日

[11] 公开号 CN 1764183A

[22] 申请日 2005.11.3

[21] 申请号 200510096291.X

[71] 申请人 西安交通大学

地址 710049 陕西省西安市咸宁路 28 号

[72] 发明人 李增智 陈彦萍 晋勤学 杨怀洲

王 创 夏 虹

[74] 专利代理机构 西安通大专利代理有限公司

代理人 李郑建

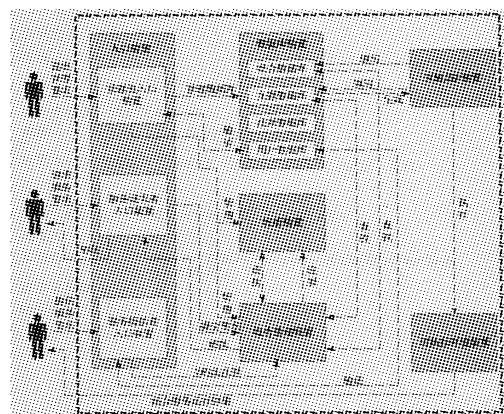
权利要求书 2 页 说明书 14 页 附图 4 页

[54] 发明名称

一种基于计算机网络的 QoS 驱动的提供组合 Web 服务的计算机系统

[57] 摘要

本发明公开了一种基于计算机网络的 QoS 驱动的提供组合 Web 服务的计算机系统，它包括组合管理装置、数据库装置、用户数据库、主题数据库、组合数据库、注册数据库、注册装置、执行环境装置、系统入口装置，该计算机系统能够尽可能的利用现有的分布在计算机网络中的 Web 服务资源，通过按照 QoS 需求进行服务组合的方式来满足计算机用户不可预料的对新服务的使用要求。该系统通过对现有的服务描述标准语言 WSDL 进行语法扩展得到新的服务描述语言 EWSDL，使用 EWSDL 可以对 Web 服务 QoS 属性进行描述，在组合时可以根据每个原子服务的 EWSDL 文件中的 QoS 参数信息按照用户的需要计算机网络上的现有 Web 服务找到合适的原子服务进行组合从而快速满足不同用户的对服务的功能和质的需求。



1. 一种基于计算机网络的 QoS 驱动的提供组合 Web 服务的计算机系统，其特征在于，该计算机系统至少包括：

一个组合管理装置，用于管理服务的查找、服务 QoS 的组合和管理；通过对每一个组合方案的决策得到最满足用户 QoS 需求的组合方案；

一个数据库装置，由用户数据库、主题数据库、注册数据库、组合数据库构成，其中：

一个用户数据库，用于存放服务提供者，服务请求者和系统管理员这三种用户的身份验证信息；

一个主题数据库，用于存放系统当前发布的主题的信息；

一个组合数据库，用于存放组合管理装置完成的一个主题的具体组合方案；

一个注册数据库，用于存放的服务开发商注册的信息；

一个 SCML2EP 装置，用于将 SCML 文件转换成可执行流程，并将其交给执行环境装置；

一个注册装置，用于存储那些注册到本系统的 Web 服务的 EWS DL 文件；

一个执行环境装置，用于执行组合服务流程，通过给 stubs 合适的 portypeID 来实现动态的服务组合；

一个系统入口装置，该系统入口装置针对不同的用户分为三种类型的入口：

①服务请求者入口，用于提供给服务请求者来查看系统发布的工作主题，通过本装置进入系统，服务请求者可以选择需要服务的主题，并和组合管理装置就服务的 QoS 签订合同，组合成功后或者在组合数据库中找到合适服务后，给服务请求者返回组合服务的 WSDL 文件；

②服务提供商入口，用于服务提供商查找组合主题，并按照主题的要求

来提供原子服务，通过这个入口进入系统，可以看到系统发布的 SCML 语言描述的组合服务的流程，SCML 语言中还包含了服务的功能描述和 QoS 描述；

③管理员入口，用于给系统管理员提供管理本系统的入口；

上述组合管理装置分别与系统入口装置、数据库装置、注册装置互连，系统入口装置还分别和数据库装置、注册装置连接，SCML2EP 装置与数据库装置连接。

2. 如权利要求 1 所述的计算机系统，其特征在于，所述的将 SCML 文件转换成可执行流程的主要任务是：

①生成可执行的流程；

②为流程中的每一个 PortType 生成其 stubs，用于在运行时绑定合适的原子服务；

③将 SCML 文件中的 PortType 填入到主题数据库中的 PortType 表；

④将 SCML 的 Operation 信息填入到主题数据库中的 Operation 表；

⑤将顺序信息或流程保存成图，填入到组合数据库中的 Sequence 表。

一种基于计算机网络的 QoS 驱动的提供组合 Web 服务的计算机系统

技术领域

本发明属于计算机网络技术领域,涉及一种利用计算机系统来快速提供复杂的、新的计算机网络服务的生成方法,确切的说,涉及一种按照用户对服务的功能属性和 QoS 属性要求,在运行时动态的组合现有的分布在计算机网络中的各种 Web 服务,从而提供给用户满足用户需求的 Web 服务组合的计算机系统。

背景技术

随着计算机网络中新的体系结构-面向服务的体系结构(SOA)的出现,面向服务的分析和设计将取代现有的面向对象的分析和设计思想。目前 SOA 有许多的实现方式,但 Web 服务成为目前公认的 SOA 的最好实现方式。现有的 Web 服务的体系结构其中主要包含三个角色,Web 服务用户,服务提供商,和服务注册中心。任何的应用程序都可以被包装成 Web 服务,并将其发布到计算机网络中的注册中心。

面向服务的目标是为了整合网络上的现有的各种服务资源,来满足用户的各种不可预料的需求,因此,当现有的分布在计算机网络中的单个 Web 服务都不能完全满足用户的需求(功能需求或者 QoS 需求)时,动态组合多个计算机网络中现有的多个 Web 服务来提供更复杂的增值服务就是一种解决方法。Web 服务本质上也是一种软件,所以 Web 服务组合具有软件复用的特点,但是它又有如下的特性:

- 服务组合专门处理计算机网络业务,这些业务处在一个分布式,松耦合的网络环境中,并且明确定义好一系列操作。

- Web 服务组合是一种与具体位置无关的“即插即用”式的动态服务重用，从而为系统动态的适应需求的变化提供技术保证。

- Web 服务组合不但可以重用服务提供者的服务处理能力，也可以透明的重用服务提供者的数据存储结构和数据内容，从而提高了软件开发和部署的效率。

下面我们通过一个在电子商务系统中使用动态服务组合的例子来更深入的理解动态 Web 服务组合带来的好处：

1. 对于一个电子商务系统 S 来说，它在设计的时候只能考虑怎么应对现有的安全问题，对未来可能出现的安全问题并不能预料，这样当新的安全问题 X 出现时，可以通过将问题 X 的解决方案封装为 Web 服务 W1 并将 W1 发布到网络的注册中心，S 通过查找注册中心找到 W1 并将其合成到 S 中得到 S'，其中 S' 就包含了问题 X 的解决方案，这样就可以应对该系统在设计时没有考虑到的安全问题，而不用重新开发 S；

2. 对于电子商务 S 来说，它可以在多种客户端设备上运行：如移动电话，PDAs，数字寻呼机，个人计算机等。这些设备的处理能力，存储能力，图形能力，和网络能力都有所不同。尽管电子商务 S 的处理流程相同，但在不同的客户端上处理电子商务 S，则需要不同的客户端处理服务。将这些不同的客户端处理服务作为单独的 Web 发布，并使用动态的 Web 服务组合技术可以方便的使 S 运行在不同的基础结构上。

由此可见，动态 Web 服务组合可以在不打断原有计算机系统的操作的前提下，快速的使原有系统适应各种改变-动态抗失配性，这对于那些高效的，任务紧急的，不能打断的实时系统来说是非常重要的。

但是对于 Web 服务组合装置来说，如果只是考虑到用户对服务的功能上的要求，而忽略了用户的 QoS（服务质量）要求就会导致提供的服务大都在运行时不能被用户接受。例如，对于上面说到的电子商务系统 S 来说，如果

在使用 S 的过程中不能保证该系统的可靠性和安全性(可靠性和安全性都属于 S 的 QoS 属性), 那么就没有用户愿意来使用这个系统。因此, 对于 Web 服务组合装置来说, 如何保证用户对服务运行时的 QoS 就成为一个迫切需要解决的问题。

要想按照用户的 QoS 需求进行服务的组合, 必需在发布的时候就能够提供更多地服务的描述信息, 而目前的 Web 服务的描述语言 WSDL (WSDL Web Service Description Language) 主要关注的是服务的功能属性, 即做什么, 怎么做, 在那里做。还不支持对 Web 服务的 QoS 属性的描述, 因此也就不支持按照用户对服务的 QoS 要求来进行 Web 服务的组合。这样要实现 QoS 驱动的 Web 服务组合, 扩展现有的 Web 服务描述语言就是首先要解决的问题, 现有的扩展 Web 服务描述语言-WSDL 的方法可以分为语法和语义两种。语法的方法就是通过对 WSDL 进行语法扩展即加入新的元素来描述更多的信息, 语义的方法就是开发一种富语义的新语言来描述 Web 服务, 如 OWL-S (Ontology Web Language-Web Service)。由于语义的方法过于复杂, 目前还没有支持对 Web 服务的 QoS 属性的描述的语法和语义问世。

发明内容

本发明的目的在于, 根据服务请求者对功能和 QoS 的要求, 提供一种基于计算机网络的 QoS 驱动的提供组合 Web 服务的计算机系统, 该计算机系统能够尽可能的利用现有的计算机网络中的 Web 服务资源, 来快速满足用户不可预料的服务需求。

本发明通过对 WSDL 进行语法扩展的方法对 Web 服务 QoS 属性进行描述。使得在组合时可以根据 QoS 需求找到合适的原子服务进行组合。

除了要对 Web 服务的 QoS 属性进行描述之外, 还要将多个原子服务按照 QoS 的要求组合成复杂的新服务, 因此还必须提供组合服务和构成组合服务的原子服务之间的 QoS 度量。

实现上述发明目的的技术解决方案是,一种基于计算机网络的 QoS 驱动的提供组合 Web 服务的计算机系统,其特征在于,该计算机系统至少包括:

一个组合管理装置,用于管理服务的查找,服务 QoS 的组合和管理;通过对每一个组合方案的多目标决策得到满足用户 QoS 需求的组合方案;

一个数据库装置,由用户数据库、主题数据库、注册数据库、组合数据库构成,其中:

- 用户数据库,用于存放服务提供者,服务请求者和系统管理员这三种用户的身份验证信息;

- 主题数据库,用于存放系统当前发布的主题的信息;

- 组合数据库,用于存放组合管理装置得到的具体组合方案;

- 注册数据库,用于存放的服务开发商注册的信息;

一个 SCML2EP 装置,用于将 SCML 文件转换成可执行流程,主要有下面的任务:

- 生成可执行的流程,并将其交给可执行环境 EE;

- 为流程中的每一个 portType 生成其 stubs,用于在运行时绑定合适的原子服务;

- 将 SCML 文件中的 PortType 填入到主题数据库中的 Porttype 表;

- 将 SCML 的 Operation 信息填入到主题数据库中的 Operation 表;

- 将顺序信息(将流程保存成图)填入到组合数据库中的 Sequence 表。

一个注册装置,用于存储那些注册到本系统的 Web 服务的 EWS DL 文件;

一个执行环境装置(EE 装置),用于执行组合服务流程,通过给 stubs 合适的 portypeID 来实现动态的服务组合;

一个系统入口装置,针对不同的用户有三种类型的入口:

- 服务请求者入口，用于提供给服务请求者来查看系统发布的工作主题，通过本装置进入系统，服务请求者可以选择需要服务的主题，并和组合管理装置就服务的 QoS 签订合同，组合成功或者在组合数据库中找到合适服务后，给服务请求者返回组合服务的 WSDL 文件。

- 服务提供商入口，用于服务提供商查找组合主题，并按照主题的要求来提供原子服务。通过这个入口进入系统，可以看到系统发布的 SCML (Service Composition Management Language) 语言描述的组合服务的流程，SCML 语言中还包含了服务的功能描述和 QoS 描述。

- 管理员入口，用于给系统管理员提供管理本系统的入口。

上述组合管理装置分别与系统入口装置、数据库装置、注册装置互连，系统入口装置还分别和数据库装置、注册装置连接，SCML2EP 装置与数据库装置连接。

为了提高软件的可重用程度，人们从基于过程的函数重用，面向对象的重用，面向组件的重用发展到目前的面向服务的重用。软件的重用方式也从紧耦合的过程调用与对象的继承等方式向松耦合的服务继承方式发展。软件的开发方式也从最早的基于代码的开发向基于组件的开发过渡，并向未来的服务自动化合成和装配的自动化软件生产方式发展。本发明正是在这个背景下，利用计算机系统，按照用户的需要组合计算机网络上的现有 Web 服务，能够满足不同用户的需求。尤其是对于那些高效的，任务紧急的，不能打断的实时系统来说，提供了一个非常重要的途径。

附图说明

图 1 是 EWSDL 中 operationInst 元素语法的图形表示；

图 2 是 SCML 中 tServiceProcess 元素的语法图；

图 3 是 SCML 中 tServiceConstraints 元素的语法图；

图 4 是本发明中的组合过程示意图；

图 5 是本系统的结构和执行流程图；

图 6 是本发明的组合管理装置的流程图；

以下结合附图对本发明作进一步的详细说明。

具体实施方式

为了更清楚的理解本发明，申请人首先给出下列定义及相关概念：

定义 1：原子服务和组合服务

由多个 Web 服务按照一定的组合逻辑组合出的新的 Web 服务称为组合服务，组合服务中包含的 Web 服务称为这个组合服务的原子服务。在本发明中，原子服务和组合服务本质上都是 Web 服务，它们的区别仅仅在于粒度的大小。

定义 2：服务类

令 A 是 Web 服务的集合，那么按照功能把这些服务分成叫做服务类的多个非空子集，使得 A 中的每一个元素都至少属于一个服务类，而同一个服务类中的服务的差异在于其 QoS 属性。注意，这些服务类的全体构成的集合只是 A 的一个覆盖，并不是一定是划分。

定义 3：候选关系

对于注册到本系统得注册装置的两个服务 $Service_a, Service_b$ 来说，如果，

$$1) \bigcup_{i=1}^l funtion(Service_a) \supseteq \bigcup_{i=1}^k funtion(Service_b), \quad l, k \in N, l \geq k, \text{ 或}$$

$$2) \bigcup_{i=1}^k funtion(Service_b) \supseteq \bigcup_{i=1}^l funtion(Service_a), \quad l, k \in N, k \geq l, \text{ 则}$$

$Service_a, Service_b$ 是候选关系，记作 $Service_a \overset{\wedge}{\longleftrightarrow} Service_b$ 。候选

关系是一种传递的，自反的，和对称的关系二元关系。因此候选关系是一种等价关系。

基于上述三个定义，本发明中的组合过程可以用图 4 来表示。其中组合服务流程可以划分为有限个服务类，而每一个服务类都有有限多个服务之间存在候选关系。

本发明的基本思想就是兼顾 Web 服务的功能信息和 QoS 信息，选择需要组合的原子服务，使得组合服务的功能和 QoS 属性能够满足用户的需求。本发明首先扩展了现有的 Web 服务描述语言 WSDL，得到了一种新的 Web 服务描述语言 EWSDL，EWSDL 在原有 WSDL 描述服务功能属性的基础上添加了描述服务 QoS 属性的结构。服务的 QoS 属性应该包括响应时间，可靠性，并发性，截至日期，价格，罚金，安全等级。将服务 QoS 属性具体化为参数，是为更好的对服务 QoS 属性进行度量。

按照上述技术方案，本发明提供了一种按照用户的 QoS 要求组合现有的服务，得到更复杂的服务的方法。具体包含以下内容：

1. Web 服务的描述模型 S

$S = (Func, QoS)$ ，这个 Web 服务模型将 Web 服务 S 描述成二元组的形式，其中 *Func* 描述了服务的功能属性的集合，QoS 描述了服务的 QoS 属性集合。现有的 WSDL 语言并不能描述服务的 QoS 属性，为了按照这个 Web 服务模型描述服务，本发明扩展了现有的 WSDL 语言得到了一种增强的 Web 服务描述语言（EWSDL-Enhanced Web Service Description Language），其中功能属性可以由现有的 Web 服务描述语言（WSDL- Web Service Description Language）中的 portType 元素表示，我们通过在 WSDL 中的 tport 元素中加入了 toperationInst 属性来描述服务的 QoS 属性。EWSDL 的语法的图形表示如图 1 所示。使用 EWSDL 的这个结构后，需要什么 QoS 只要需要都可以随时加入进行扩展。

2. 增强的 Web 服务描述语言 EWSDL

EWSDL 和原有的 WSDL 相比增加了 tOperationInst 元素，通过这个元素，QoS 属性可以按照需要定义，本发明中的 QoS 属性涉及如下内容：

第一，responseTime，表示原子服务的响应时间；通常，它和 price 之间不是独立的，而是相关的，响应时间是一个分段函数，根据该服务的 price 而不同。

第二，availability，表示原子服务的可靠性，即服务可以被其服务请求者正确使用的概率；

第三，concurrency，表示原子服务的并发性，即原子服务在运行时可以同时支持多少个并发事务；

第四，expireTime，表示原子服务的截至日期，在这个日期之前，都可以保证服务的可靠性；

第五，price，表示原子服务的价格，即服务使用者由于使用该服务而需向服务提供商提供的费用；

第六，fine，表示服务商提供商和服务请求者之间由于其中一方的违约而按照合同支付给另一方的罚金，作为违约方的惩罚和损失方的补偿。一般和价格成线性关系；

第七，securityLevel，表示原子服务的安全等级。

下面给出一个用 EWSDL 语言中 operationInst 的例子（只给出在现有的 WSDL 上添加的内容）：

```
<wsdl:operationInst name="approve" operationID="DBA609F4-28DA-48B3-BB4B-046811A23C48">
    <wsdl:constraintInst name="ResponseTime" value="3" probality="79"
constraintID="648CDC87-121F-47B7-A05F-B8186472F2D1"/>
    <wsdl:constraintInst name="Price" value="74" probality="83"
constraintID="DC984545-0AEB-494F-8DDD-B830C1D54CFC"/>
    <wsdl:constraintInst name="SecurityLevel" value="3" probality="54"
```

```

constraintID="DC984545-0AEB-494F-8DDD-B830C1D54CFC"/>
        <wsdl:constraintInst name="Stability" value="65" probality="34"
constraintID="DC984545-0AEB-494F-8DDD-B830C1D54CFC"/>
        <wsdl:constraintInst name="Avalilablity" value="49" probality="47"
constraintID="535D9654-BBB5-417A-B83E-492115B4D124"/>
        <wsdl:constraintInst name="Concurrence" value="8" probality="47"
constraintID="535D9654-BBB5-417A-B83E-492115B4D124"/>
</wsdl:operationInst>

```

3. 计算组合服务中 QoS 参数的方法

对一个由 m 个原子服务组成的组合服务 A 来说, 其 QoS 参数的值可以按照下面的方式计算得到:

第一, ResponseTime, 表示组合服务的响应时间。考虑到组合服务执行流程中包含并发执行的服务, 因此, 组合服务中的 ResponseTime 这个参数的值并不是所有的原子服务的 responseTime 之和, 也就是说 $ResonseTime \neq \sum_{i=1}^m responseTime_i$, 因此, 必须把组合服务的流程按照图的形式保存起来, 并从中找到一条关键路径, 即执行中响应时间最长的那条路。CPA 就是寻找关键路径的一个算法, 即:

$$ResponseTime(A) = CPA (Service_1, Service_2, \dots, Service_m);$$

第二, Availability, 表示组合服务的可靠性, 对于组合服务来说, 它的可靠性为每一个原子服务的可靠性之积; 即:

$$Availability(A) = \prod_{i=1}^m avalibility_i;$$

第三, Concurrency, 表示组合服务的并发性, 对于组合服务来说, 它的并发性由那个并发性最小的原子服务决定; 即:

$$Concurrency(A) = \min (concurrency_1, concurrency_2, \dots, concurrency_m),$$

第四, ExpireTime, 表示组合服务的截至日期, 对于组合服务来说它的截至日期由那个最近截至的原子服务决定, 即:

$\text{ExpireTime}(A) = \min(\text{expireTime}_1, \text{expireTime}_2, \dots, \text{expireTime}_m),$

第五, Price, 表示组合服务的价格, 组合服务的价格应给是所包含的所有原子服务的价格之和, 即;

$$\text{Price}(A) = \bigcup_{i=1}^m \text{price}_i;$$

第六, Fine, 表示组合服务的罚金, 一般价格越高罚金也越高, 即: $\text{Fine}(A) = C * \text{Price}(A)$, 其中 C 是由用户和提供商进行协商后得到的;

第七, SecurityLevel, 表示组合服务的安全等级, 它有那个安全等级最低的原子服务来决定, 即:

$\text{SecurityLevel}(A) = \min(\text{securityLevel}_1, \text{securityLevel}_2, \dots, \text{securityLevel}_m).$

4. 服务组合管理语言 SCML (Service Composition Management Language)

服务组合管理语言 SCML 不同于其他的 Web 服务流程描述语言 (例如 BPEL4WS, BPEL4WS 本质上是一种过程建模语言, 支持将一组现有的服务整合成一个新的服务, 缺点是服务组合运行时的静态性), 在 E-WsFrame 中, SCML 可以描述组合服务的各种需要的属性。SCML 的根元素 tService 由两个子元素组成。一个是 serviceProcess, 描述了服务的流程, 体现了服务的功能属性; 另一个是 serviceConstraints, 描述了对服务的约束, 体现了服务的 QoS 属性。而在我们常用的 Web 服务流程描述语言 BPEL4WS 中, 仅仅只包含了 SCML 中的 serviceProcess 元素中的部分信息, 这样的流程就紧紧包含了功能方面的内容却没有涉及 QoS 的内容。SCML 的 tServiceProcess 元素和 tServiceConstraints 元素的语法如图 2, 3 所示。

下面, 申请人给出一个 SCML 中 serviceConstraints 元素的实例:

```
<scml:serviceConstraints>
```

```
  <scml:constraint name="ResponseTime" valuetype="DECIMAL-MAX" composetype="CPA"
  constraintID="648CDC87-121F-47B7-A05F-B8186472F2D1">
```

```

    <scml:description>ResponseTime</scml:description>
  </scml:constraint>

  <scml:constraint    name="Price"    valuetype="DECIMAL-MAX"    composetype="CPA"
constraintID="535D9654-BBB5-417A-B83E-492115B4D124">
    <scml:description>Price</scml:description>
  </scml:constraint>

  <scml:constraint    name="SecurityLevel"    valuetype="DECIMAL-MAX"    composetype="MAX"
constraintID="DC984545-0AEB-494F-8DDD-B830C1D54CFC">
    <scml:description>SecurityLevel</scml:description>
  </scml:constraint>

  <scml:constraint    name="Stability"    valuetype="DECIMAL-MIN"    composetype="MIN"
constraintID="19013B0B-581F-4FEB-A8D7-4E26630BE21D">
    <scml:description>Stability</scml:description>
  </scml:constraint>

  <scml:constraint    name="Avalilability"    valuetype="DECIMAL-MIN"    composetype="PI"
constraintID="FDF0B248-CB24-4EF8-AB9F-97FB2E59B740">
    <scml:description>Avaliability</scml:description>
  </scml:constraint>

  <scml:constraint    name="Concurrence"    valuetype="DECIMAL-MIN"    composetype="MIN"
constraintID="657B814D-00D0-45C2-932F-63A0BDA6A565">
    <scml:description>Concurrence</scml:description>
  </scml:constraint>

```

5. 按照 QoS 组合服务的计算机系统

基于计算机网络的 QoS 驱动的提供组合 Web 服务的计算机系统，该计算机系统至少包括：

第一，组合管理装置，它的功能是管理中心提供服务的注册，查找 UDDI，基于服务 QoS 的组合，管理的功能。

第二，数据库装置由用户数据库、主题数据库、注册数据库、组合数据库构成，其中：

- 用户数据库里存放了三种用户（服务提供者，服务请求者和本系统的管理员）的信息。

- 主题数据库里存放的是系统当前运行的主题的信息。
- 组合数据库里存放的是对一个主题的具体合成方案。
- 注册数据库里存放的是服务开发商注册的信息。

第三，系统入口装置，包含以下几种入口：

- 服务请求者入口，提供给服务请求者来查看系统发布的工作主题；进入后，服务请求者选择需要的服务，并和组合管理装置就 QoS 签订合同。组合成功后给服务请求者返回组合服务的 WSDL 文件。

- 服务提供商入口，提供给服务提供商，用来按照发布的主题的要求来提供服务。通过这个界面进入，可以看到系统发布的组合服务的流程，SCML（Service Composition Management Language）语言描述。SCML 语言中包含了服务的功能描述和 QoS 描述。

- 管理员入口，提供给系统管理员来管理本系统。

第四，SCML2EP 装置，是将 SCML 文件转换成可执行流程，主要有下面的任务：

- 生成可执行的流程，（可以是 BPEL4WS 语言描述的流程，也可以是其它语言描述的流程）并将其交给可执行环境装置（EE 装置）；

- 为流程中的每一个 PortType 生成其 stubs，用于在运行时绑定合适的原子服务；

- 将 SCML 文件中的 PortType 填入到主题数据库中的 PortType 表；
- 将 SCML 的 Operation 信息填入到主题数据库中的 Operation 表；
- 将顺序信息（将流程保存成图）填入到组合数据库中的 Sequence 表。

第五，执行环境装置（EE 装置）。用于执行组合服务流程，通过对每一个组合方案的决策得到最满足用户 QoS 需求的组合方案，通过给 stubs 合适

的 portTypeID 来实现动态的服务组合。

图 5 是发明人给出的本系统的主要流程。

参照图 5，它是本发明的结构和执行流程图。其中在系统提供组合服务前需要的准备工作步骤如下：首先，将组合服务主题的 SCML 文件发布到系统的入口装置上，服务提供者可以按需（SCML 描述）提供原子服务；其次，通过 SCML2EP 装置将 SCML 文件转换成可执行的流程，最后，生成该组合服务的 WSDL 文件（注意，此时 WSDL 还不能执行，只有在运行时才动态绑定原子服务）存储到主题数据库和组合数据库中。

这时，服务请求者就可以使用服务请求入口装置来使用组合服务，首先，在系统入口装置上找到自己需要的主题，也找到了该主题的 WSDL 文件（这个 WSDL 是对组合服务流程的描述，执行的时候通过执行环境 EE 来进行），可是该 WSDL 不是可以执行的。因为只有满足用户 QoS 要求的方案才可以用来提交给用户执行；其次，按照 SCML 中提供的属性，填写用户和组合管理装置之间的合同；再次，组合管理装置按照服务请求者和其之间签订的合同选择合适的子服务，将用户要求的服务的 operation 提交给 stubs 执行。最后，当收到用户的请求时在 EE 执行这个可执行流程，并按照用户的 QoS 需求绑定每一个原子服务。

对于服务提供者来说，他们通过服务提供者的系统入口装置发现本系统提供的主题及其发布的 SCML 文件，可以将服务提供者提供的 Web 服务的 EWSDL 文件注册到本系统的注册装置中；

而本系统的管理员可以通过系统入口装置的管理员入口进入，来管理本系统中的注册装置，数据库装置，和组合管理装置。

参照图 6，其为本发明组合检测的流程图。

首先进行步骤 101，确定组合服务选择的服务类的个数初始化全局变量 i ， i 表示组合服务流程中需要进行选择的服务的数量，

接下来进行步骤 102，从本系统的管理中心注册中心中查找服务类，然后进入步骤 103，即检查现有的服务类的功能是否满足用户对该功能的需要，如果可以找到满足的服务类，则进入步骤 104，否则进入步骤 110。在步骤 110 输出本系统中没有足够的原子服务。在步骤 104，将循环的次数减少，并进入步骤 106，如果这时循环仍未结束，则跳入步骤 102。直到找到所有的业务类。

在步骤 106 要对服务的 QoS 属性进行检查，进入步骤 107，如果组合服务的 QoS 符合用户的要求则进入 108，否则进入 109。在步骤 108，将组合的结果输出，并结束组合。在步骤 109，判断是否对服务类中的候选服务的组合遍历完全，如果没有，则继续进行步骤 106，否则进入步骤 110。在步骤 110，输出服务组合结果，并结束组合。

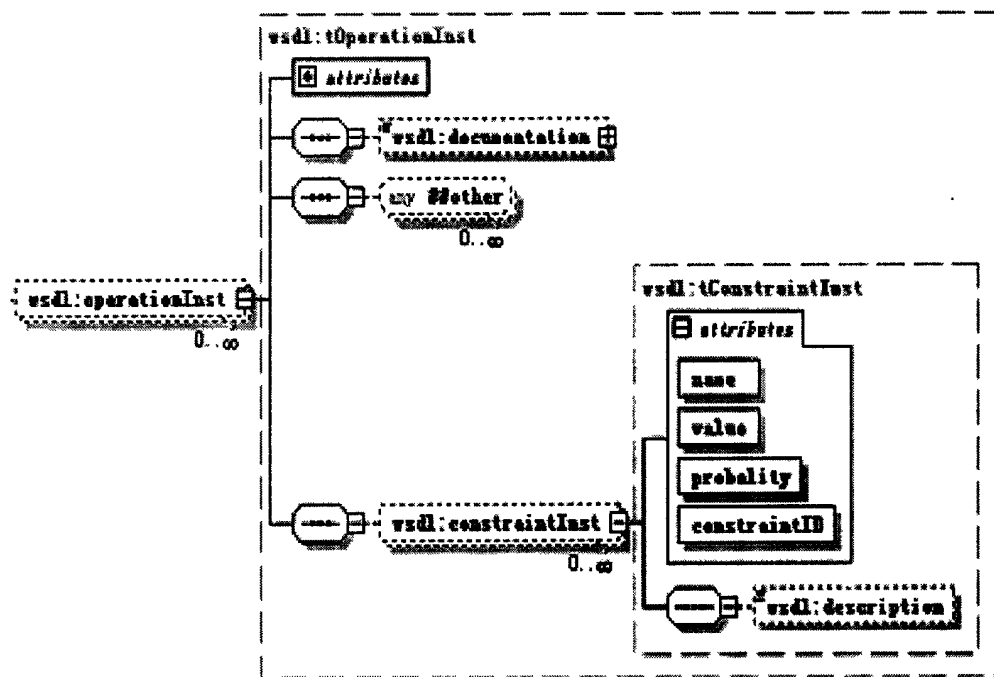


图 1

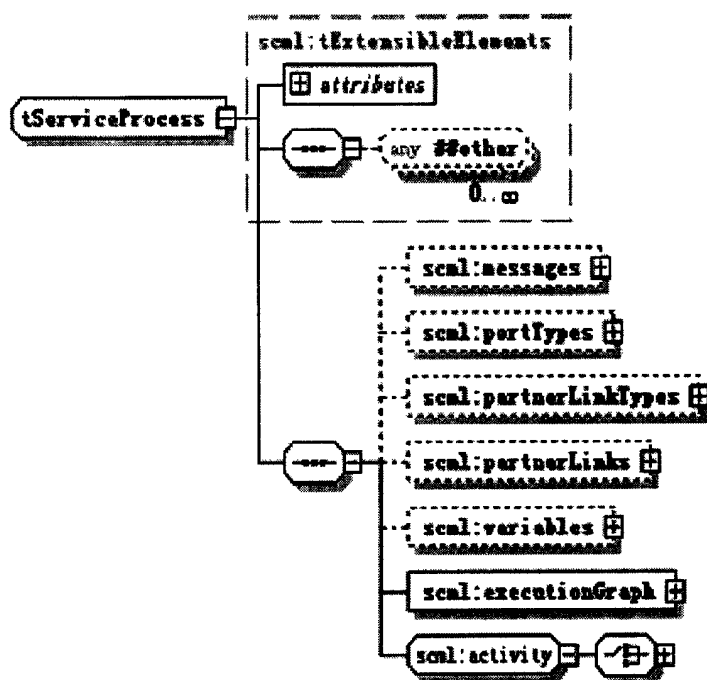


图 2

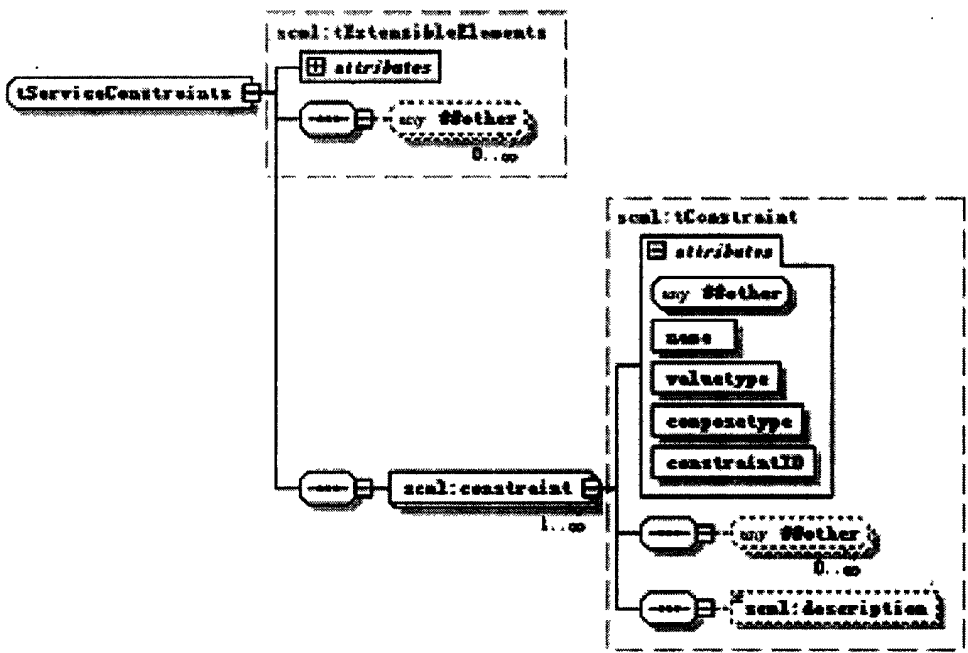


图 3

组合服务的主过程

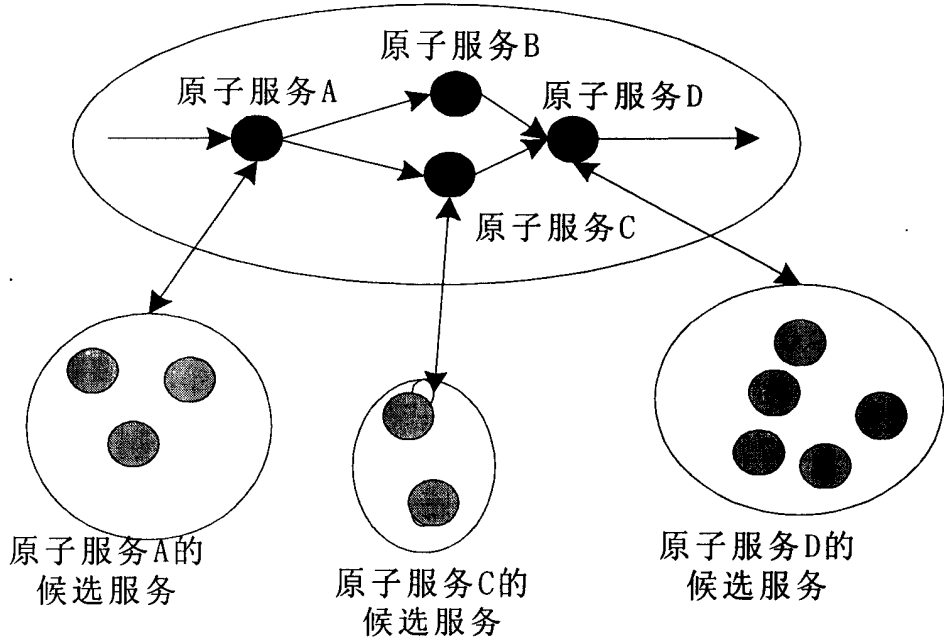


图 4

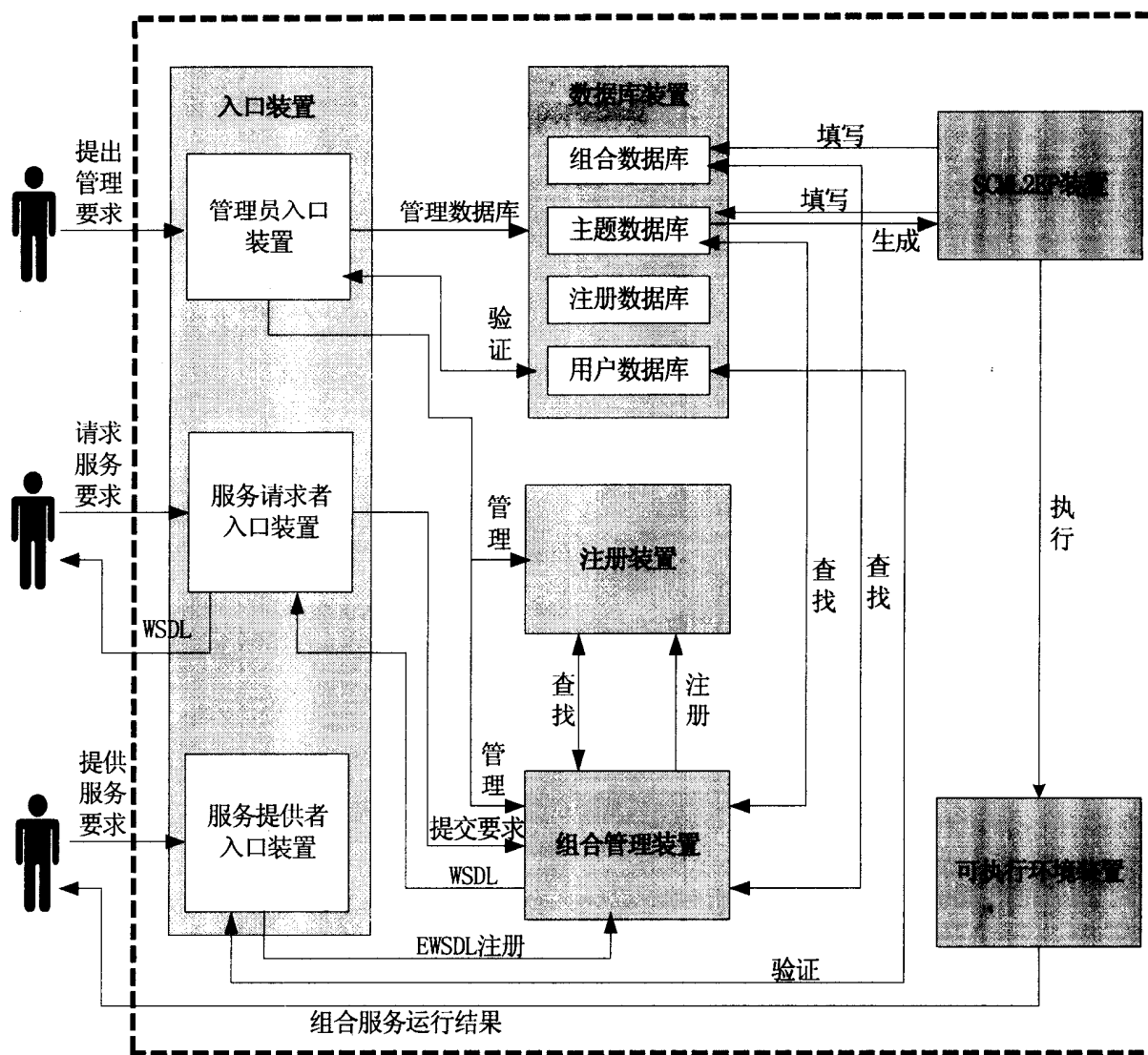


图 5

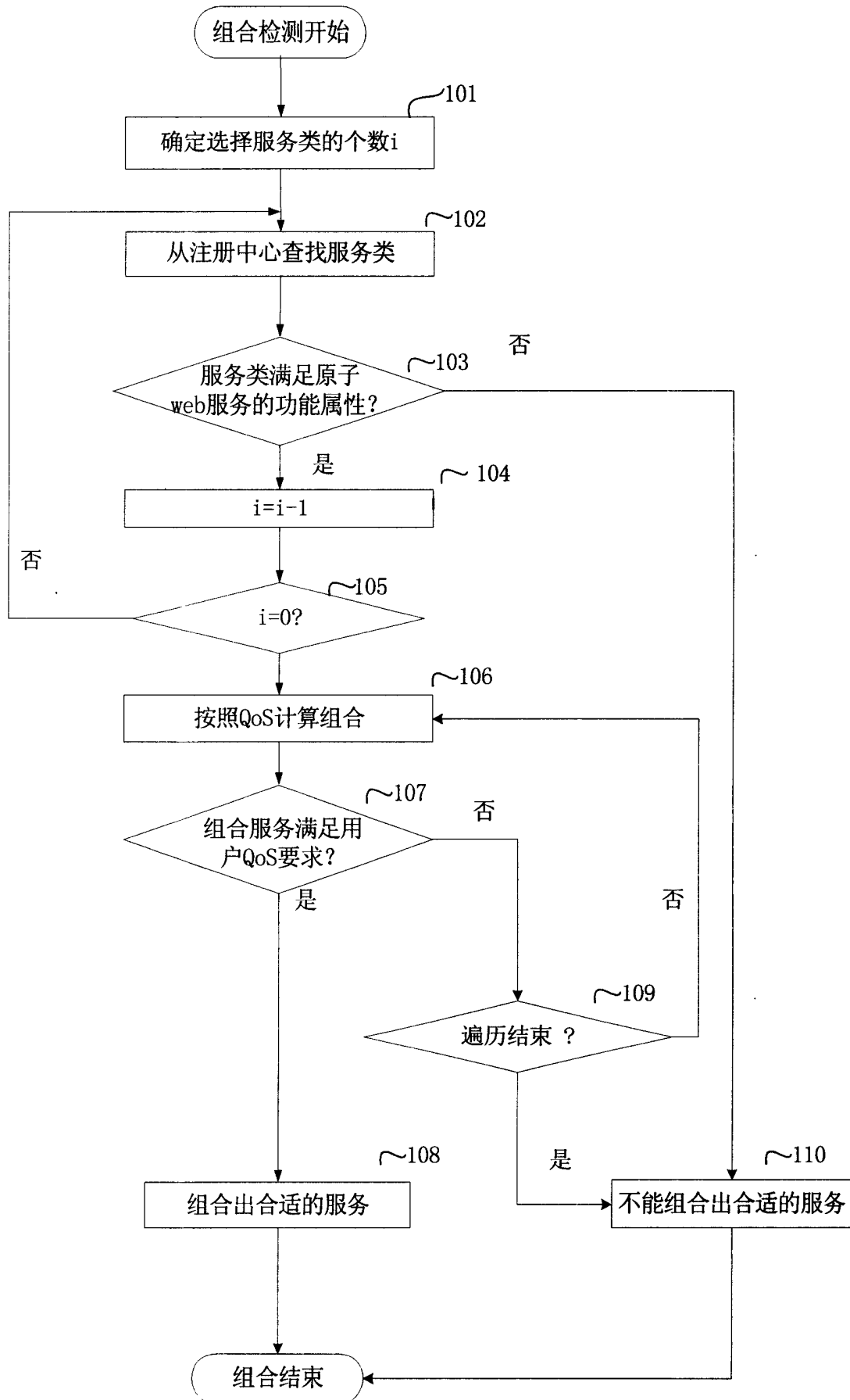


图 6