(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101094223 B
(45)授权公告日 2010.12.08

(21)申请号 200610093193.5

(22)申请日 2006.06.23

(73)专利权人 国际商业机器公司
地址 美国纽约

(72)发明人 周宇辰 刘昕鹏

(74)专利代理机构 北京市中咨律师事务所 11247
代理人 李峥 于静

(51)Int.Cl.
*H04L 29/06*(2006.01)
*H04L 12/28*(2006.01)

(56)对比文件
US 2005027871 A1,2005.02.03, 全文.
WO 2005017654 A2,2005.02.24, 全文.
US 6442620 B1,2002.08.27, 全文.

审查员 周丹

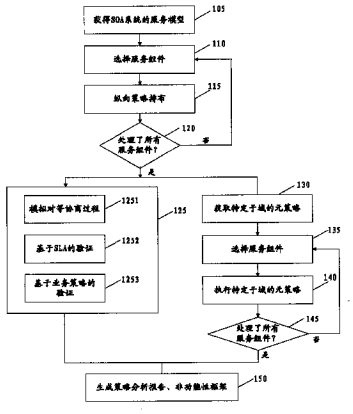
权利要求书 4 页 说明书 7 页 附图 3 页

(54)发明名称

# 在面向服务体系结构系统的服务模型中排布策略的方法和装置

(57)摘要

本发明公开了一种在面向服务体系结构系统的服务模型中排布策略的方法，其中上述服务模型至少包括多个服务组件和上述多个服务组件之间的服务依赖关系，并且上述多个服务组件的至少一个被指定了策略，该方法包括：为上述服务模型中多个服务组件的每一个进行纵向策略排布，以获得该服务组件的有效策略；以及按照每个应用域，对上述多个服务组件的有效策略进行横向排布。




CN 101094223 B

1. 一种在面向服务体系结构系统的服务模型中排布策略的方法，其中上述服务模型至少包括多个服务组件和上述多个服务组件之间的服务依赖关系，并且上述多个服务组件的至少一个被指定了策略，该方法包括：

为上述服务模型中多个服务组件的每一个进行纵向策略排布，以获得该服务组件的有效策略；以及

按照每个应用域，对上述多个服务组件的有效策略进行横向排布；

其中，为上述服务模型中多个服务组件的每一个进行纵向策略排布进一步包括：为上述服务模型中多个服务组件的每一个合并要对该服务组件执行的所有有效策略；

对上述多个服务组件的有效策略进行横向排布进一步包括：验证上述多个服务组件中具有依赖关系的每一对服务组件的策略正确性；和/或对上述多个服务组件的有效策略，根据该应用域的元策略进行排布。

2. 根据权利要求 1 所述的在面向服务体系结构系统的服务模型中排布策略的方法，其中上述服务模型还包括系统，上述多个服务组件中的至少一个属于该系统，且上述系统被指定了策略；

上述为服务模型中多个服务组件的每一个合并要对该服务组件执行的所有有效策略的步骤包括：

对于上述系统中的每个服务组件，计算其策略与该系统的策略的与集合。

3. 根据权利要求 2 所述的在面向服务体系结构系统的服务模型中排布策略的方法，其中上述系统还包括子系统，该系统的服务组件中的至少一个属于该子系统，且上述子系统被指定了策略；

上述为服务模型中多个服务组件的每一个合并要对该服务组件执行的所有有效策略的步骤包括：

对于上述子系统中的每个服务组件，计算其策略与上述系统、该子系统的策略的与集合。

4. 根据权利要求 3 所述的在面向服务体系结构系统的服务模型中排布策略的方法，其中上述系统包括第一子系统和第二子系统，该系统的服务组件中的至少一个同时属于该第一子系统和第二子系统，且该第一子系统和第二子系统均被指定了策略；

上述为服务模型中多个服务组件的每一个合并要对该服务组件执行的所有有效策略的步骤包括：

检查上述第一子系统与上述第二子系统之间是否存在策略冲突；

在上述第一子系统与上述第二子系统之间存在策略冲突时，解决策略冲突；以及

对于上述同时属于该第一子系统和第二子系统中的每个服务组件，计算其策略与上述系统、第一子系统及第二子系统的策略的与集合。

5. 根据权利要求 4 所述的在面向服务体系结构系统的服务模型中排布策略的方法，其中上述解决策略冲突的步骤包括：

根据执行序列、子集关系以及预定义的策略优先级中的任意一种来解决子系统之间的策略冲突。

6. 根据权利要求 1 所述的在面向服务体系结构系统的服务模型中排布策略的方法，其中上述验证多个服务组件中具有依赖关系的每一对服务组件的策略正确性的步骤包括：

对于上述具有依赖关系的服务组件对，通过模拟对等协商过程，验证其策略的一致性。

7. 根据权利要求 1 所述的在面向服务体系结构系统的服务模型中排布策略的方法，其中上述验证多个服务组件中具有依赖关系的每一对服务组件的策略正确性的步骤包括：

利用预先定义的服务级别协议验证上述具有依赖关系的服务组件的策略。

8. 根据权利要求 1 所述的在面向服务体系结构系统的服务模型中排布策略的方法，其中上述验证多个服务组件中具有依赖关系的每一对服务组件的策略正确性的步骤包括：

利用与服务质量有关的业务策略验证上述具有依赖关系的服务组件的策略。

9. 根据权利要求 1 所述的在面向服务体系结构系统的服务模型中排布策略的方法，其中上述对多个服务组件的有效策略，根据该应用域的元策略进行排布的步骤包括：

定义用于对上述多个服务组件进行策略排布的元策略；

指定一个或多个服务组件，以定义进行策略排布的范围；以及

将上述用于策略排布的元策略应用到上述所指定的一个或多个服务组件，以验证策略的一致性和正确性。

10. 根据权利要求 9 所述的在面向服务体系结构系统的服务模型中排布策略的方法，其中上述用于策略排布的元策略包括：服务组件之间的目标关系类型和对具有该目标关系的服务的约束。

11. 根据权利要求 10 所述的在面向服务体系结构系统的服务模型中排布策略的方法，其中上述指定一个或多个服务组件的步骤包括：

根据上述用于策略排布的元策略的目标关系类型指定一个或多个相关系的服务组件，作为策略排布的执行范围。

12. 根据权利要求 1 所述的在面向服务体系结构系统的服务模型中排布策略的方法，还包括：

根据上述纵向策略排布和横向排布的排布结果，生成上述多个服务组件的非功能性框架。

13. 根据权利要求 12 所述的在面向服务体系结构系统的服务模型中排布策略的方法，其中上述根据上述纵向策略排布和横向排布的排布结果生成上述多个服务组件的非功能性框架的步骤包括：

根据上述纵向策略排布和横向排布的排布结果，生成上述多个服务组件的策略分析报告。

14. 根据权利要求 12 所述的在面向服务体系结构系统的服务模型中排布策略的方法，其中上述根据上述纵向策略排布和横向排布的排布结果生成上述多个服务组件的非功能性框架的步骤包括：

增加用于确保策略的一致性和正确性的附加策略断言。

15. 根据权利要求 12 所述的在面向服务体系结构系统的服务模型中排布策略的方法，其中上述根据上述纵向策略排布和横向排布的排布结果生成上述多个服务组件的非功能性框架的步骤包括：

为未解决的策略冲突生成非功能性框架的补充组件的框架。

16. 根据权利要求 1 所述的在面向服务体系结构系统的服务模型中排布策略的方法，其中上述策略包括可靠性消息传送策略、安全策略和基本事务策略中的至少一种。

17. 一种用于在面向服务体系结构系统的服务模型中排布策略的装置，其中上述服务模型至少包括多个服务组件和上述多个服务组件之间的服务依赖关系，并且上述多个服务组件的至少一个被指定了策略，该装置包括：

纵向策略排布单元，用于为上述服务模型中多个服务组件的每一个进行纵向策略排布，以获得该服务组件的有效策略；以及

横向策略排布单元，用于按照每个应用域，对上述多个服务组件的有效策略进行横向排布；

其中，上述纵向策略排布单元进一步为上述服务模型中多个服务组件的每一个合并要对该服务组件执行的所有有效策略；

上述横向策略排布单元包括：策略正确性验证单元，用于验证上述多个服务组件中具有依赖关系的每一对服务组件的策略正确性；和/或元策略排布单元，用于对上述多个服务组件的有效策略，根据该应用域的元策略进行排布。

18. 根据权利要求 17 所述的在面向服务体系结构系统的服务模型中排布策略的装置，其中上述服务模型包括第一子系统和第二子系统，上述多个服务组件中的至少一个同时属于该第一子系统和第二子系统，且该第一子系统和第二子系统均被指定了策略；其中上述纵向策略排布单元包括：

策略冲突解决单元，用于检查上述第一子系统与上述第二子系统之间是否存在策略冲突，并在存在策略冲突时，解决策略冲突。

19. 根据权利要求 17 所述的在面向服务体系结构系统的服务模型中排布策略的装置，其中上述策略正确性验证单元包括：

协商过程模拟单元，用于对于上述具有依赖关系的服务组件对，通过模拟对等协商过程，验证其策略的一致性。

20. 根据权利要求 17 所述的在面向服务体系结构系统的服务模型中排布策略的装置，其中上述策略正确性验证单元包括：

SLA 验证单元，用于利用预先定义的服务级别协议验证上述具有依赖关系的服务组件的策略。

21. 根据权利要求 17 所述的在面向服务体系结构系统的服务模型中排布策略的装置，其中上述策略正确性验证单元包括：

业务策略验证单元，用于利用与服务质量有关的业务策略验证上述具有依赖关系的服务组件的策略。

22. 根据权利要求 17 所述的在面向服务体系结构系统的服务模型中排布策略的装置，其中上述元策略排布单元包括：

元策略定义单元，用于定义用于对上述多个服务组件进行策略排布的元策略；以及

元策略存储库，用于存储所定义的用于策略排布的元策略。

23. 根据权利要求 17 所述的在面向服务体系结构系统的服务模型中排布策略的装置，其中上述元策略排布单元还包括：

排布范围选择单元，用于指定一个或多个服务组件，以定义进行策略排布的范围。

24. 根据权利要求 17 所述的在面向服务体系结构系统的服务模型中排布策略的装置，其中上述元策略排布单元还包括：

元策略执行单元，用于将上述用于策略排布的元策略应用到上述所指定的一个或多个服务组件，以验证策略的一致性和正确性。

25. 根据权利要求 17 所述的在面向服务体系结构系统的服务模型中排布策略的装置，还包括：

非功能性框架生成单元，用于根据上述纵向策略排布和横向排布的排布结果，生成上述多个服务组件的非功能性框架。

# 在面向服务体系结构系统的服务模型中排布策略的方法和装置

## 技术领域

[0001] 本发明涉及数据处理领域，具体地，涉及在面向服务体系结构 (Service-Oriented Architecture，SOA) 系统的服务模型中排布策略的方法和装置。

## 背景技术

[0002] 随着 web 服务应用的日益复杂化，第一代 web 服务体系结构已不再适应目前 web 服务的开发需求。因此，在第一代体系结构的基础上提出了一种面向服务的体系结构 (Service-Oriented Architecture，SOA)。SOA 是一个概念，它定义一个 web 服务系统可以由一系列独立但又相互协作的子系统或者服务组成。这样的结构使各个服务独立出来，只将需要声明的接口告诉给其他的服务即可。因此，SOA 允许创建松散耦合的企业业务流程。这样，在 SOA 中，一个 Web 服务系统可以由远程的、不同域中的 web 服务所构成，从而能够使跨越多个业务范围的服务流程取代传统的分层结构。

[0003] SOA 系统应满足服务级别协议 (Service Level Agreement，SLA)，以保证企业所购买服务的可靠性和实用性，从而满足企业对服务质量的要求。web 服务策略则用于描述 web 服务在与其他 web 服务或用户的交互中的需求和能力，其是服务级别协议的重要保证。

[0004] 策略是一个宽泛的术语，其涉及到与 SOA 系统中功能性的部分 (functional) 相对应的非功能性的部分 (non-functional)，诸如安全性、可靠性、事务、隐私等各个领域。类似的，表达策略的方式也不局限于通用策略或安全性策略的表达方式。对于 web 服务策略而言，一般包括策略框架 (Policy Framework，WS-Policy) 文档，其定义了表达 web 服务策略的语法；策略附加 (Policy Attachment，WS-Policy Attachment) 文档，其定义了如何将这些策略附加到 web 服务；通用策略断言 (WS-PolicyAssertions) 及一组安全性策略断言 (WS-Security Policy)。

[0005] 由 IBM、BEA、Microsoft 等定义的 web 服务策略框架 (Web ServicesPolicy Framework，WS-Policy) 是 web 服务策略的实际标准。它提供描述 web 服务的策略的通用模型和相应的语法。WS-Policy 旨在允许可扩展性。也就是说，WS-Policy 定义了一组基本的结构，这组结构可由其他 web 服务规范使用及扩展，以描述广泛的服务需求和能力。在 WS-Policy 的基础之上，已针对系统的不同方面而定义出一组标准，这些标准包括 web 服务可靠性消息传递策略 (WS-RM Policy)、web 服务安全策略 (WS-SecurityPolicy)、web 服务基本事务 (WS-Atomic Transaction)、web 服务策略断言 (WS-Policy Assertions) 等。用户也可根据需要基于 WS-Policy 和相关的标准定义策略语言。

[0006] 当前，大部分系统开发人员的注意力都集中在运行时策略执行审核上。然而，在所设计的服务发布之前的设计时策略执行和验证对于确保服务的正确性也是相当重要的。目前，例如 Systinet 策略管理器 (Systinet PolicyManager) 提供了用于验证单个服务的对于 WS-I(Web ServicesInteroperability Organization，web 服务互操作性组织，其发布了 WS-I 基本概要，概要包含核心 web 服务规范的实现指导原则，这些指导原则是一组要求，定

义了应该如何用这些规范来开发可互操作的web服务）的遵从性、文档完整性、句法有效性的工具。但是，这种工具仅针对于单个服务的策略的正确性、完整性和有效性的验证。

[0007] 因此，目前还不存在用于在SOA系统中进行设计时服务策略的正确性和一致性的验证的方法和工具，也不存在用于生成SOA系统的非功能性结构的框架的工具。

[0008] 发明内容

[0009] 本发明正是鉴于上述现有技术中的问题提出的，其目的在于提供在面向服务体系结构系统的服务模型中排布策略的方法和系统，以便能够在SOA系统设计时对于其多个服务组件的策略进行正确性和一致性的验证，从而有助于SOA系统的设计以及设计的正确性。

[0010] 根据本发明的一个方面，提供了一种在面向服务体系结构系统的服务模型中排布策略的方法，其中上述服务模型至少包括多个服务组件和上述多个服务组件之间的服务依赖关系，并且上述多个服务组件的至少一个被指定了策略，该方法包括：为上述服务模型中多个服务组件的每一个进行纵向策略排布，以获得该服务组件的有效策略；以及按照每个应用域(domain)，对上述多个服务组件的有效策略进行横向排布。

[0011] 根据本发明的另一个方面，提供一种用于在面向服务体系结构系统的服务模型中排布策略的装置，其中上述服务模型至少包括多个服务组件和上述多个服务组件之间的服务依赖关系，并且上述多个服务组件的至少一个被指定了策略，该装置包括：纵向策略排布单元，用于为上述服务模型中多个服务组件的每一个进行纵向策略排布，以获得该服务组件的有效策略；以及横向策略排布单元，用于按照每个应用域，对上述多个服务组件的有效策略进行横向排布。

## 附图说明

[0012] 相信通过以下结合附图对本发明具体实施方式的说明，能够使人们更好地了解本发明上述的特点、优点和目的。

[0013] 图1是根据本发明实施例的在面向服务体系结构系统的服务模型中排布策略的方法的流程图；

[0014] 图2示出一个示例性SOA系统的服务模型；以及

[0015] 图3是根据本发明实施例的在面向服务体系结构系统的服务模型中排布策略的系统的方框图。

[0016] 具体实施方式

[0017] 下面就结合附图对本发明的各个优选实施例进行详细的说明。

[0018] 图1是根据本发明实施例的在面向服务体系结构系统的服务模型中排布策略的方法的流程图。如图1所示，首先在步骤105，获得SOA系统的服务模型。该服务模型包括多个服务组件及服务组件之间的依赖关系。

[0019] 其中，每一个服务组件都可作为服务提供者而实现一个或多个服务，这些服务可利用WSDL(Web Service Definition Language，web服务描述语言）来详细说明。同样，每一个逻辑服务组件又可以作为服务用户而调用其他的零到多个逻辑服务组件。

[0020] 此外，在SOA系统的服务模型中，服务组件之间的依赖关系由带方向的链接来表示。其中，从一个服务组件到另一个服务组件的带方向的链接表示从服务提供者到服务用

户的服务调用方向。

[0021] 此外，在SOA系统的服务模型中，有时还存在系统或子系统。系统或子系统是一组具有虚拟界限和关系的服务组件。一个系统中可以包含一个或多个子系统。

[0022] 下面举例来具体说明SOA系统的服务模型。图2示出了一个示例性SOA系统的服务模型。如图2所示，该SOA系统的服务模型包括多个服务组件SC1、SC2、SC3、SC4、SC5和SC6，这些服务组件中的每一个均可实现一个或多个服务。并且这些服务组件通过代表它们之间的依赖关系的箭头“→”相互连接。每一个箭头“→”都表示从服务用户到服务提供者的服务调用方向。此外，在该SOA系统的服务模型中，服务组件SC1、SC2、SC3、SC4及其相互之间的依赖关系一起组成子系统1，服务组件SC5和SC6及其相互之间的依赖关系组成子系统2，子系统1和子系统2及其相互之间的依赖关系则构成了整个系统。此外服务组件SC1、SC3、SC4和SC5及子系统1、系统均被指定了一个或多个策略。

[0023] 需要说明的是，图2中SOA系统的服务模型仅是一个例示，其中所示的服务组件、系统及子系统的组成并不意味着是对本发明中的服务模型的形式上的限制，本发明对SOA系统的服务模型的形式并没有特别限制。此外，为了便于说明，图2中仅示出了6个服务组件及两个子系统，而实际上，随着web服务体系结构的发展，一个SOA系统所要实现的功能及提供的服务是相当繁杂的，因此其逻辑服务模型也会远比图示的情况要复杂。另外，虽然图2示出的是逻辑服务模型，但是本实施例的方法同样适用于物理服务模型。

[0024] 此外，在本实施例中，步骤105的目的是获得所设计的SOA系统的服务模型。也就是说，在本实施例中，该SOA系统的服务模型可通过任何形式获得，可以是人工新创建的，也可以是从另一SOA系统预设计阶段直接获取的相应的SOA系统的服务模型的创建结果。

[0025] 接着，在步骤110-120，为上述服务模型中多个服务组件的每一个进行纵向策略排布，以获得该服务组件的有效策略。

[0026] 具体地，在步骤110，选择上述多个服务组件中要进行纵向策略排布的对象，即服务组件。

[0027] 在步骤115，对上述多个服务组件中所选择的服务组件进行纵向策略排布。具体地，为该服务组件合并要对其执行的所有有效策略。

[0028] 一个服务组件的策略可包括该服务组件作为服务提供者的服务提供者策略和该服务组件作为服务用户的用户策略。这些策略具体可以是针对该服务组件的操作、端口、端口类型和消息而指定的。在为一个服务组件进行有效策略的合并时，要考虑对该组件直接指定的所有策略。此外，由于一个服务组件所属系统及子系统的策略对于该服务组件也是有效的，所以在为该服务组件进行策略合并时还要考虑该服务组件所属系统及子系统的策略。

[0029] 因此，在该步骤中，对于一个服务组件SC而言，按下式来计算该组件的最终有效策略：

[0030] AP(S) = Aggregate(PP, Nesting(SSP1, ..... SSPn), SP)

[0031] 其中，PP是对该服务组件SC指定的策略，SPPi是对子系统i指定的策略，SP是对系统指定的策略。Aggregate()是一个函数，其使用在web服务策略附件(WS-Policy Attachment)中定义的算法计算作为参数的策略的“与”集合。而Nesting()是解决具有相交部分的嵌套子系统之间的策略冲突的函数，也就是说，在该服务组件SC同时属于具有相

交部分的多个子系统 SSP1，.....SSPn 中时，针对该服务组件 SC，要由 Nesting() 函数来检查并解决这些子系统之间的策略冲突。

[0032] 服务模型中嵌套子系统之间的策略冲突可根据执行序列来解决，不同主体级别之间的策略冲突可根据主体之间的子集关系通过断言覆盖来解决。另外，还可根据预定的策略优先级来解决主体之间的策略冲突。

[0033] 接着，在步骤 120，判断是否为服务模型中的所有服务组件均进行了策略合并，若是则前进到步骤 125，否则返回步骤 110，选择下一个服务组件。

[0034] 接着，在步骤 125-145，按照每个应用域 (domain)，对上述多个服务组件的有效策略进行横向排布。在步骤 110-120 期间为这些服务组件计算出的有效策略可应用到此阶段中，以便实现对具有依赖关系的服务组件的横向策略排布。

[0035] 在横向策略排布阶段，可以同时或依次进行两种类型的策略排布 :模拟验证及基于元策略的排布。

[0036] 首先，说明第一种类型的策略排布。

[0037] 在步骤 125，进行模拟验证，以验证上述多个服务组件中具有依赖关系的每一对服务组件的策略正确性。

[0038] 步骤 125 又可细化为三个具体步骤 1251、1252、1253。

[0039] 在步骤 1251，对于每一对具有依赖关系的服务组件，特别是对于系统和子系统内具有固定依赖关系的每一对服务组件，通过模拟实际的对等协商过程并生成用于服务调用的有效策略，来检查该对服务组件中作为服务提供者的服务组件的提供者策略与作为服务用户的服务组件的用户策略的一致性。

[0040] 在步骤 1252，对于服务模型中具有动态依赖关系的每一对服务组件，利用预先为用户定义的 SLA（服务级别协议），来验证该对服务组件中作为服务提供者的服务组件的提供者策略和作为服务用户的服务组件的用户策略对于该 SLA 的遵从性。

[0041] 在步骤 1253，对于服务模型中具有动态依赖关系的每一对服务组件，利用与 QoS（服务质量）有关的业务策略，来验证该对服务组件中作为服务提供者的服务组件的提供者策略和作为服务用户的服务组件的用户策略对于该 QoS 的遵从性。

[0042] 需要说明的是，步骤 1251、1252、1253 之间不存在相互依赖关系，因此，它们的执行顺序是任意的，可以同时执行，也可以依次执行。此外，也可以省略其中的某个步骤。

[0043] 接着，说明第二种类型的策略排布。

[0044] 在步骤 130-145，对上述多个服务组件的有效策略，进行基于元策略的排布。

[0045] 首先，在步骤 130，获取上述服务模型中特定于应用域的用于策略排布的元策略。

[0046] 在本实施例中，步骤 130 的目的是获得特定于应用域的元策略，而其对元策略的获得方式并没有特别限制。因此，在本实施例中，特定于应用域的用于策略排布的元策略可以是预先针对该应用域定义并存储在元策略存储库中的，也可以是为该应用域新定义的。

[0047] 下面对本实施例的特定于应用域的用于策略排布的元策略进行详细说明。基于 SOA 系统的应用域的属性，一个应用域中的主体的策略之间都是相互依赖的。因此，在本实施例中，将用于策略排布的元策略的形式定义为包括 :服务组件之间的目标关系类型（诸如依赖关系链接、子集关系等）、对于具有该目标关系类型的服务的约束。

[0048] 此外，根据策略验证的需要，可定义下列两种类型的用于策略排布的元策略 :

[0049] MPT1 :策略出现的正确性 ;

[0050] MPT2 :策略的一致性。

[0051] 以图 2 的 SOA 系统的服务模型为例,在 web 服务可靠消息传送 (WS-Reliable Messaging,WS-RM) 的情况下,上述的目标关系类型是调用链接,并且存在下列元策略 :

[0052] MPT1 :RM(SC1,SC3)A EP(SC1,SC2,SC3)- > RM(SC1,SC2)A RM(SC3)

[0053] 对于 WS-RM 策略的 inactiveTimeout 断言的值来说,如果 SC1、SC2 是执行路径的元素,则存在下列元策略 :

[0054] MPT2 :L(SC1)<L(SC2)->IT(SC1)>IT(SC2)

[0055] 其中 L(SC2) 是 SC2 在该执行路径中的位置,而 IT(SC1) 是 SC1 的 inactiveTimeout 的值。

[0056] 在 web 服务事务 (WS-Transaction) 的情况下,上述的目标关系类型是子集合,且存在下列元策略 :

[0057] MPT1 :TX(SC1)A(L(SC1)<(SC2)->TX(SC2)

[0058] MPT2 :AT(SS1)A Belong(SC1,SS1)->AT(SC1)

[0059] 接着,在步骤 135,根据用于策略排布的元策略的目标关系类型,指定服务模型中处于一个应用域内的一个或多个服务组件,以指定在服务模型中进行基于元策略的排布的范围。

[0060] 目标关系表达描述服务组件之间的依赖关系,这种依赖关系通常包括有方向的链接或子集合关系。例如,在图 2 中,可以定义下列执行路径 :

[0061] EP1(Consumer,SC1,SC2,SC4,SC5)

[0062] EP2(Consumer,SC1,SC3,SC4,SC5)

[0063] 下面给出元策略的一个具体实例。

[0064] 如果服务模型中某一路径的始端的用户 1- 该路径的尾端的提供者 n 都通过策略要求可靠的消息传送,则为了满足用户 1 及提供者 n 的可靠调用的需求,该 RM 路径 (RMPath) 上的所有用户 / 提供者策略都被要求是 WS-RM 形式的。因此,这种对于 WS-RM 策略附件的特定于应用域的限制可由下面的元策略来表示 :

[0065] Meta-Policy1 :{Id =”RMChain”,Domain =”WS-RM”Rule =< RM($Consumer_1$)AND RM($Provider_n$)AND Composite($Provider_i$,$Consumer_{i+1}$)AND Invoke($Consumer_1$,$Provider_i$) AND Invoke($Consumer_{i+1}$,$Provider_n$) → EnableRM($Provider_i$)AEnableRM($Consumer_{i+1}$),i = 1,2,..,n-1,其中

[0066] $Consumer_1$ = Header(RMPath),

[0067] Providern = Tail(RMPath),

[0068] RMPath = Vector($Consumer_1$,$Provider_1$,$Consumer_2$,$Provider_2$,...,$Provider_n$) > }

[0069] 在上面元策略 Meta-Policy1 的 Rule 表达式中,RMPath 是具有由标准的向量组成的用户策略 / 提供者策略的模型结构。每一个特定于 WS-RM 应用域的谓词的含义是 :RM 将返回作为参数的策略是否具有 WS-RM 形式 ;Composite 将返回作为参数的策略是否是同一服务组件的 ;Invoke 将返回用户 - 提供者策略对是否属于具有直接或间接调用关系的服务 ;EnableRM 是一个动作,其在参数中的相应服务不具有 WS-RM 形式策略时,对该服务指定

这样的策略。

[0070] 在步骤 140，对所选择的一个或多个服务组件，执行特定于这些组件所属应用域的用于策略排布的元策略，以检查其策略的一致性和正确性。

[0071] 此外，在存在策略冲突的情况下，可以根据预先定义的策略优先级，通过使具有较高优先级的策略优先于与其不一致的较低优先级的策略，来自动解决策略冲突。

[0072] 在步骤 145，判断是否对上述服务模型中的所有服务组件都进行了基于元策略的排布，若是，则前进到步骤 150，否则返回步骤 135 继续选择未排布的服务组件。

[0073] 需要说明的是，上述两种类型的策略排布，即步骤 125 与步骤 130-140 之间不存在相互依赖关系，因此，它们的执行顺序是任意的，可以同时执行，也可以依次执行。此外，也可以省略其中某种类型的策略排布。

[0074] 接着，在步骤 150，根据上述策略排布的结果，生成策略分析报告和该服务模型的非功能性框架或其中的一种，以便作为服务组件的设计和实现阶段的参考。

[0075] 在本实施例中，上述非功能性框架包括服务组件的非功能性能力、附加的策略断言、非功能性框架的补充组件的框架。

[0076] 其中，服务组件的非功能性能力包括可靠性消息传送、事务和安全特征等非功能性特征。

[0077] 附加的策略断言是为了系统级别的策略的一致性和正确性而自动添加的。例如，在对事务策略进行排布的情况下，如果 TX(SC1)and ! TX(SC2)，应用 MPT1：TX(SC1)A(L(SC1)<(SC2))->TX(SC2)，则可在 SC2 的提供者策略中插入基本事务 (AtomicTransaction) 断言，以便为 TX(SC2) 实现对基本事务特征的支持。

[0078] 非功能性框架的补充组件的框架是为了弥补不能自动解决的策略冲突的缺陷而生成的。例如，如果 AT(SC4)and BA(SC5)，则将生成附加的非功能性组件，以便在上述两种不同的事务类型之间进行转换。

[0079] 以上，就是对本实施例的在面向服务体系结构系统的服务模型中排布策略的方法的描述。本发明首先以 SOA 系统的服务模型中具体的服务组件为对象，为其合并有效策略从而进行纵向策略排布；然后，再按照 SOA 系统中应用域的不同，进行横向的策略排布。因而，在本实施例中，能够在 SOA 系统设计时验证服务策略的准确性和一致性，同时进行自动的策略冲突检测 / 解决，并根据策略验证结果生成策略分析报告以及服务模型的非功能性框架。

[0080] 在同一发明构思下，图 3 是根据本发明实施例的在面向服务体系结构系统的服务模型中排布策略的系统的方框图。如图 3 所示，本实施例的在面向服务体系结构系统的服务模型中排布策略的系统 300 包括：策略合并单元 301、协商过程模拟单元 302、SLA 输入单元 303、SLA 验证单元 304、业务策略输入单元 305、业务策略验证单元 306、服务组件选择单元 307、元策略定义单元 308、元策略存储库 309、元策略执行单元 310、策略分析报告及非功能性框架生成单元 311。

[0081] 策略合并单元 301，用于根据从外部获得的服务模型，为该服务模型中的多个服务组件合并要对其执行的有效策略。如上所述，该服务模型包括多个服务组件及服务组件之间的依赖关系，并且至少一个服务组件被指定了策略。该策略合并单元 301 在操作上可以实现前面结合图 1 说明的策略排布方法中的纵向策略排布阶段。

[0082] 协商过程模拟单元 302，用于对于上述服务模型中每一对具有依赖关系的服务组件，通过模拟实际的对等协商过程，验证该对服务组件的提供者策略和用户策略的一致性。

[0083] SLA 输入单元 303，用于输入预先为用户定义的 SLA（服务级别协议），并将该 SLA 转换为该策略排布系统 300 可用的格式。

[0084] SLA 验证单元 304，用于根据 SLA 输入单元 303 输入的 SLA，验证服务模型中每一对具有依赖关系的服务组件的相应提供者策略和用户策略对于该 SLA 的遵从性。

[0085] 业务策略输入单元 305，用于输入预先定义的与 QoS（服务质量）有关的业务策略，并将该业务策略转换为该策略排布系统 300 可用的格式。

[0086] 业务策略验证单元 306，用于利用业务策略验证单元 306 输入的与服务质量 (QoS) 有关的业务策略，验证服务模型中每一对具有依赖关系的服务组件的相应提供者策略和用户策略对于该业务策略的遵从性。

[0087] 服务组件选择单元 307，用于指定服务模型中处于一个应用域内的一个或多个服务组件，以指定在服务模型中进行基于元策略的排布的范围。

[0088] 元策略生成单元 308，用于为上述服务模型中的所有应用域定义用于策略排布的元策略。

[0089] 元策略存储库 309，用于存储通过元策略生成单元 308 定义的特定于应用域的用于策略排布的元策略。

[0090] 元策略执行单元 310，用于根据通过服务组件选择单元 307 选择的一个或多个服务组件，从元策略存储库获取针对这些服务组件所处应用域定义的用于策略排布的元策略，并对这些服务组件执行该元策略，以检查其策略的一致性和正确性。

[0091] 策略分析报告及非功能性框架生成器 311，用于根据上述元策略排布单元 310 及协商过程模拟单元 302、SLA 验证单元 304、业务策略验证单元 306 的排布结果，生成策略分析报告和非功能性框架或其中的一种。

[0092] 元策略执行单元 310 及协商过程模拟单元 302、SLA 验证单元 304、业务策略验证单元 306 在操作上可以实现前面结合图 1 说明的策略排布方法中的横向策略排布阶段。

[0093] 需要说明的是，协商过程模拟单元 302、SLA 验证单元 304、业务策略验证单元 306 之间不存在相互依赖关系，因此，它们之间的连接顺序是任意的，可以是顺序连接的，也可以是并列连接的。此外，也可以省略其中的某个单元。

[0094] 以上，就是对本实施例的在面向服务体系结构系统的服务模型中排布策略的系统的描述。本发明首先以 SOA 系统的服务模型中具体的服务组件为对象，为其合并有效策略从而进行纵向策略排布；然后，再按照 SOA 系统中应用域的不同，进行横向的策略排布。因而，在本实施例中，能够在 SOA 系统设计时验证服务策略的准确性和一致性，同时进行自动的策略冲突检测 / 解决，并根据策略验证结果生成策略分析报告以及服务模型的非功能性框架。

[0095] 以上虽然通过一些示例性的实施例对本发明的在面向服务体系结构系统的服务模型中排布策略的方法和系统进行了详细的描述，但是以上这些实施例并不是穷举的，本领域技术人员可以在本发明的精神和范围内实现各种变化和修改。因此，本发明并不限于这些实施例，本发明的范围仅以所附权利要求为准。

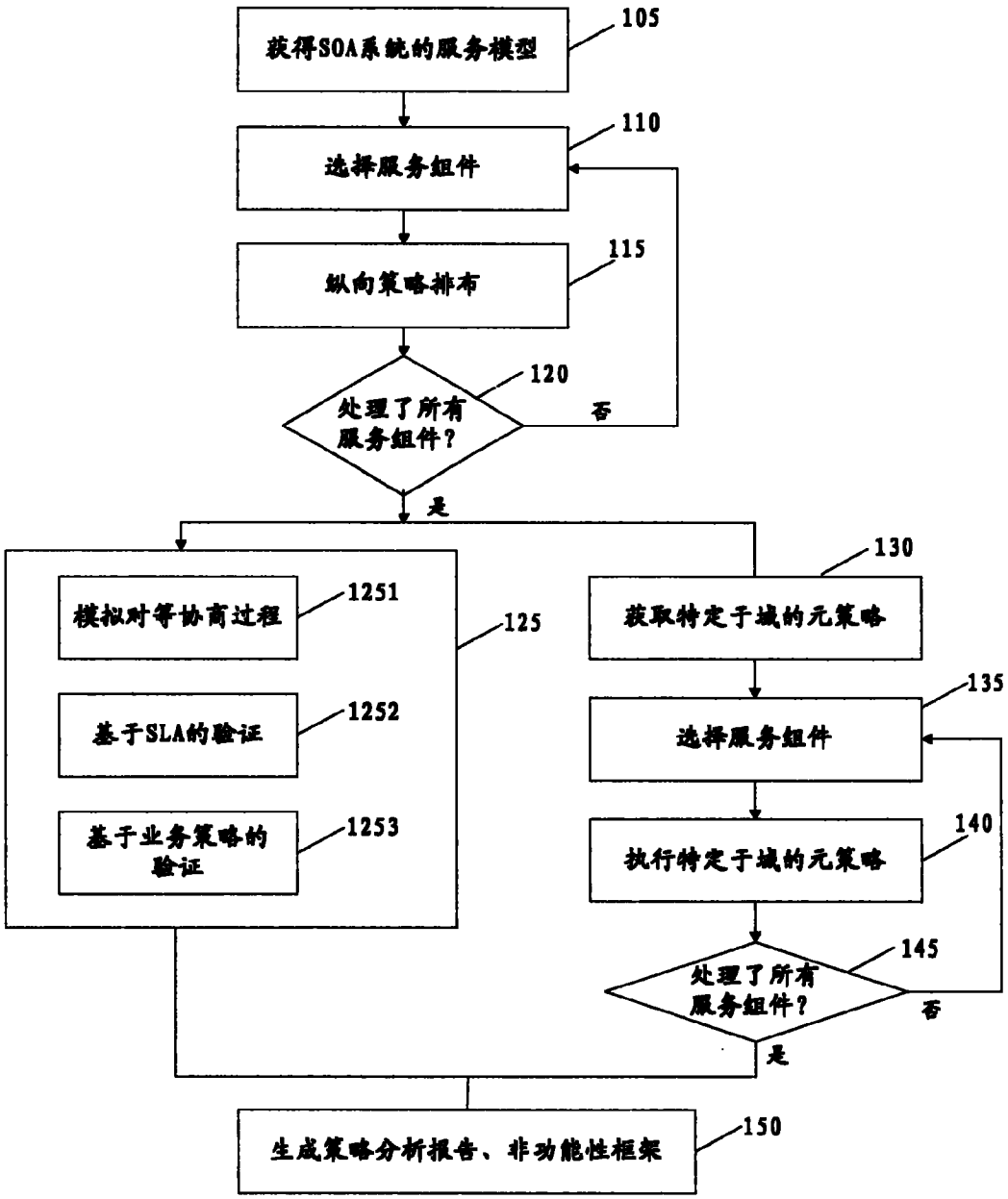

获得SOA系统的服务模型
105
选择服务组件
110
纵向策略排布
115
处理了所有服务组件?
120
否
是
模拟对等协商过程
1251
基于SLA的验证
1252
基于业务策略的验证
1253
125
获取特定于域的元策略
130
选择服务组件
135
执行特定于域的元策略
140
处理了所有服务组件?
145
否
是
生成策略分析报告、非功能性框架
150

图 1

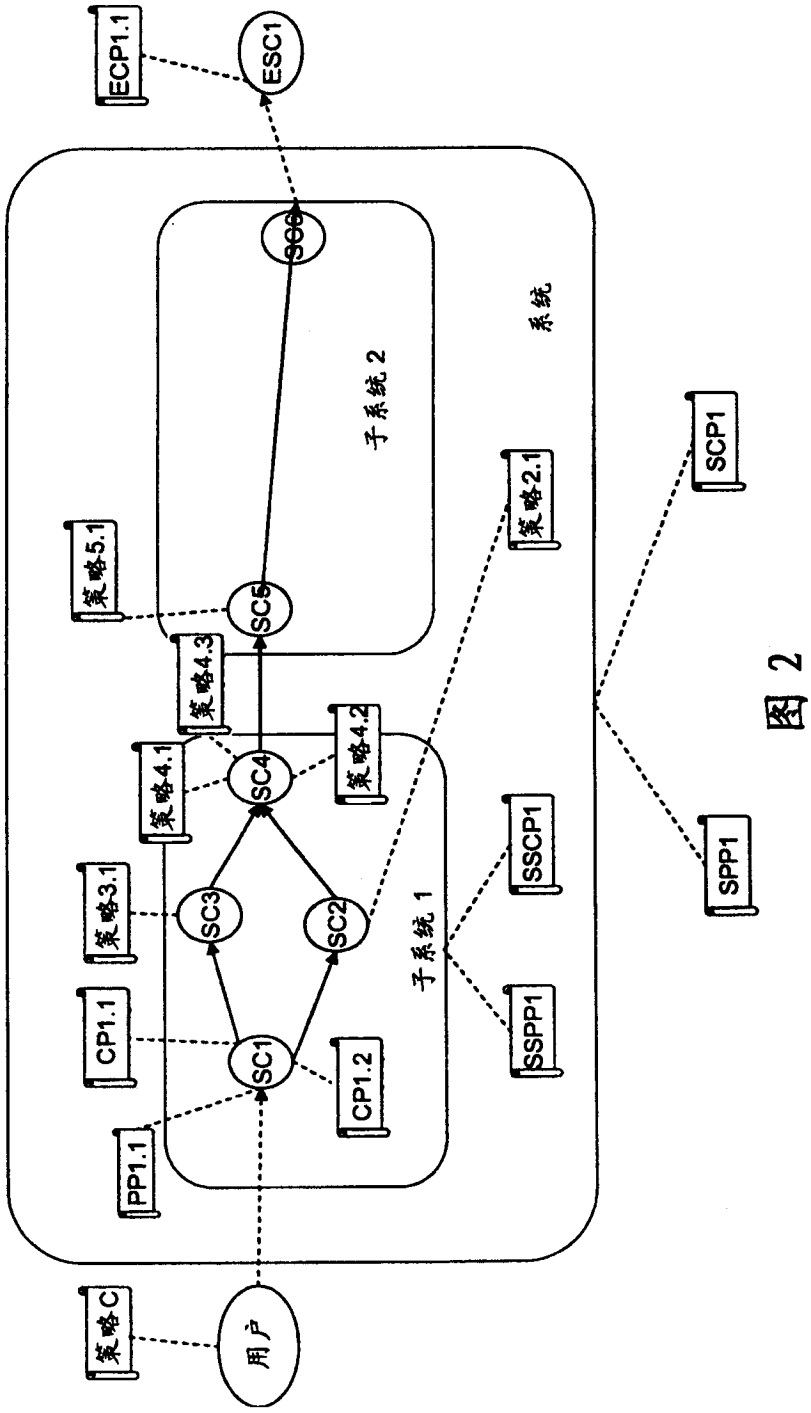

ECP1.1
ESC1
SC6
子系统 2
系统
策略5.1
SC5
策略4.3
策略4.1
SC4
策略4.2
策略3.1
SC3
SC2
CP1.1
SC1
CP1.2
PP1.1
子系统 1
策略2.1
SSCP1
SSPP1
SCP1
SPP1
策略C
用户

图 2

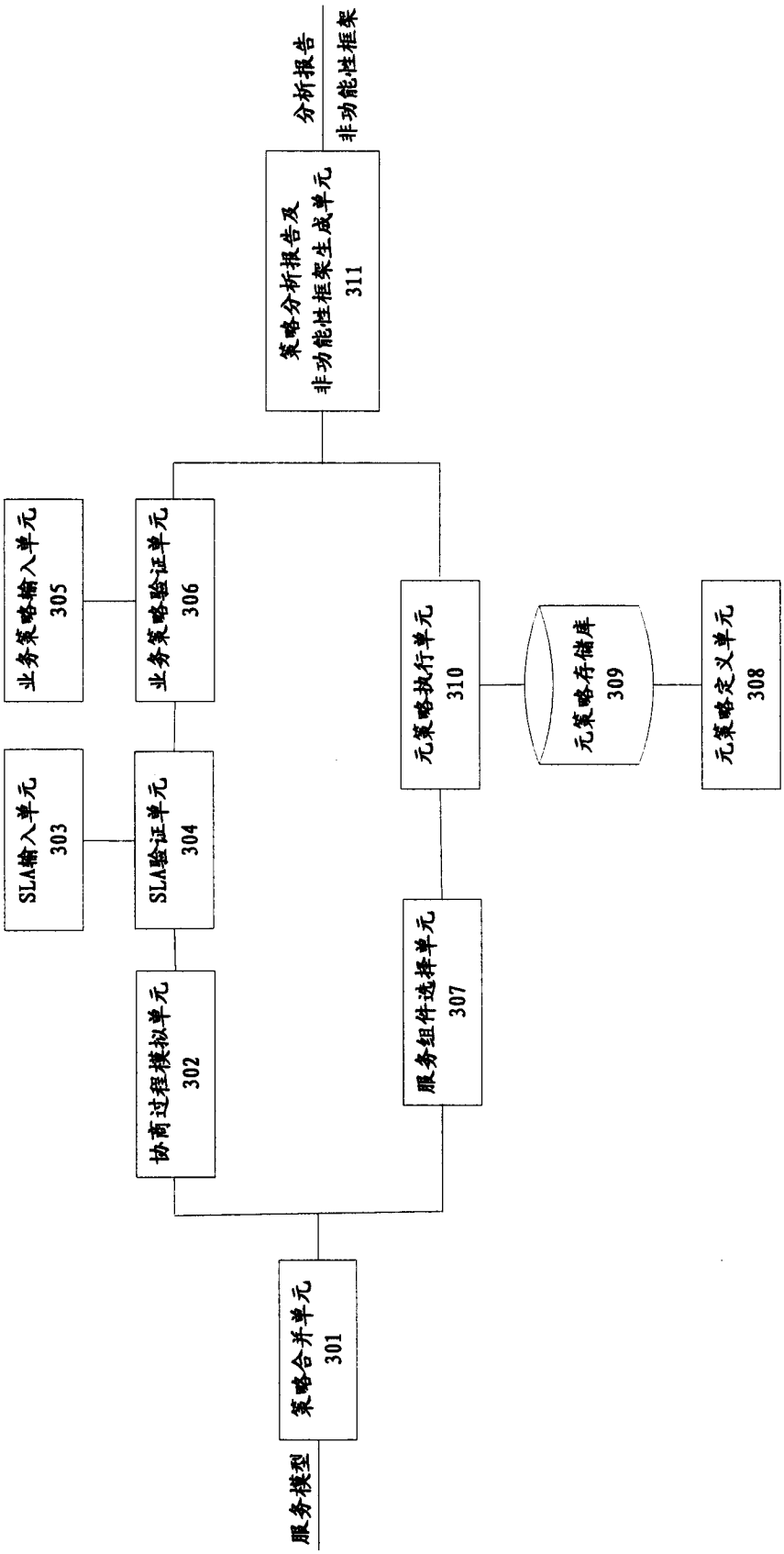

服务模型
策略合并单元
301
协商过程模拟单元
302
SLA输入单元
303
SLA验证单元
304
业务策略输入单元
305
业务策略验证单元
306
服务组件选择单元
307
元策略定义单元
308
元策略存储库
309
元策略执行单元
310
策略分析报告及
非功能性框架生成单元
311
分析报告
非功能性框架

图3