

基于动态服务协作的实时数据图形化

Real-time Data Visualization Base on Dynamic Service Cooperation

林海 陈建明 薛莹 (西安交通大学软件学院 西安 710049)

摘要:新兴的面向服务体系结构正在加速软件的发展和变化,且实时信息在人们的生活中的地位越来越重要。鉴于服务间密切的联系,本文定义了动态服务协作流程,提出了动态服务协作模型,动态地重构服务间的协作流程,并基于这种模型,设计和实现实时数据图形化系统。借助于模型驱动机制,根据用户需求,动态进行模型变换,更好的适应快速变化的需求。

关键词:面向服务 服务协作 SVG 数据可视化

1 引言

随着计算机技术、通信技术、控制技术的发展和变化,实时性信息都对人们的生活越来越重要,能否找到一种合理的、通用的系统来可视化这些实时信息显得非常重要。由于不同的实时数据发布系统都是独立的,且分布在多样的操作环境和系统环境中,所以,服务间的协作显得格外重要。面对着快速变化的市场需求,企业计算必须以动态和灵活作为应对策略;面对不同用户在使用各个服务的差异性,提供的服务协作必须是可动态调整的。本文讨论首先讨论的面向服务设计,然后提出了动态服务协作的概念并提出了模型,并且根据实时数据的特点设计了系统的体系结构,最后给出了服务协作的模型实例。

2 动态服务协作模型

2.1 面向服务的设计

服务作为一个逻辑实体,是由一个或多个已发布接口定义的契约;由服务提供者实现服务规范,而服务使用者(即客户端)请求服务,并且服务的实现对使用者是透明的;通过服务定位器查找服务提供者接口和服务位置,也可以通过服务代理将服务请求传送到一个或多个其他的服务提供者。通常,服务具有以下几个特征:

(1) 粗粒度。相对于组件而言,服务上的操作被实现为封装了更多的功能,并且依赖于更大的数据集。

(2) 基于接口的设计。服务实现为单独定义的接

口。这样带来的好处是单个服务可以实现多个接口,而多个服务也可以实现统一的接口。

(3) 可发现。不论是在设计时还是在运行时,服务都应该能被发现。这种发现可能是通过接口标志,也可以是通过服务标志。

本文所涉及的服务可看作是一个由交互式服务组成的联合体。由此,如果系统有递归的含义,就能发现一个服务是一个多个子服务的联合,而其中的每一个子服务又可以是一个多服务的联合,并依此进行分解。为了达到实用的目的,这个分解的过程必须是可终止的。因此,我们需要有这样一个“服务”概念,它允许当我们希望停止分解的时候可以将系统视为原子服务,在希望进行分解的时候又能继续。即可以将整个系统作为一个服务,而这个服务是通过用户对多个子服务的动态配置所组成的。

2.2 动态服务协作的定义

服务协作是按一定流程将服务组织在一起,并且符合其中每个服务的输入和输出标准,共同完成用户的任务。动态服务协作就是动态地进行服务组合,按照不同的需求建立不同的组态模型,满足用户的需求。如何获取服务的描述,以及对服务的定位,必须通过一个服务协作代理(Service Cooperation Agent, SCA),通过 SCA 获得相应的服务信息,结合用户对应的需求进行服务协作建模。

服务集由服务描述和服务存根组成,服务存根就是动态调用服务的本地接口,用户根据需求,结合已有

的服务描述信息,建立动态的服务模型,通过服务协作代理,产生服务协作投影。因此,将动态服务协作过程分为三层结构:

(1) 协作活动 (Cooperative Activity, CA) 是指服务协作中对应的服务间调用的最小单元,即服务中方法的一次实现,它可能涉及一个或者更多参与者协作完成,参与者提供调用方法时的参数信息,并且参与者提供的参数信息应与方法的参数规定相符合。

(2) 服务组合 (Service Assembly, SA) 是指用于协调或者指挥特定服务协作执行的序列,由一系列服务和相关服务协作活动组成。

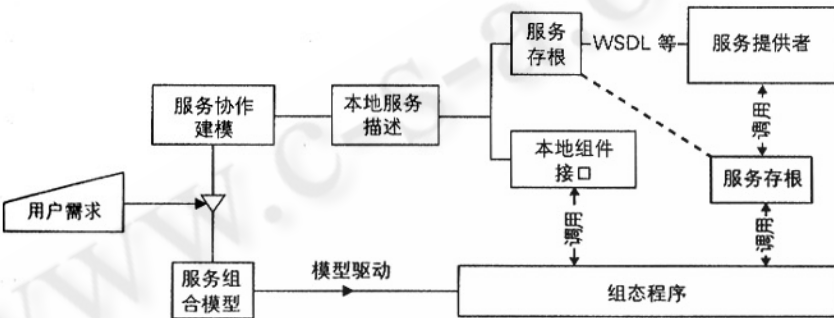


图 1 动态服务协作模型

(3) 协作投影 (Cooperation Projection, CP) 是指某个特定参与者为了完成特定任务的协作流程的一次实例,是服务组合在计算机中的实现。

2.2.1 动态服务协作的算法描述:

符号说明:①起始 X :程序的入口点;②终点集合 Y :用户对程序输出结果的形式和要求的集合;③服务集 Σ :相应服务调用的组件集合;④用户需求集合 UR :用户需求的表现;⑤输入需求集合 IR :计算机所能接受的输入。

IR 为 UR 和 Σ 的交集 $UR \cap \Sigma$; f 为一个集合到一个序列的函数, $f: IR \rightarrow SA$, 其中 SA 是以 $x \in X$ 为起始且以 $y \in Y$ 为结尾的序列; Π 是一个从一个序列到程序的一次执行的同态函数, $\Pi: SA \rightarrow CP$ 。

2.3 动态服务协作模型

以动态服务协作流程为基础,结合面向对象的设计,建立动态服务协作模型 (Dynamic Service Cooperation Model, DSCM) 如图 1 所示。

服务也就是由提供者提供给接收者的功能系统,并以服务描述的形式构成两者之间的一种契约关系,其中提供者承诺提供,而接收者同意接收。在面向服务的设计中,服务提供者将服务以一种标准的服务描述,使得接收者能够发现、绑定以及调用。

服务描述可以通过 Web 服务描述语言 (WSDL)、Web 服务发现协议 (DISCO)、以及能够一致理解的服务描述语言获得。WSDL 协议定义了服务描述文档的结构,如类型、消息、端口类型、端口和服务本身。DISCO 协议定义了如何从资源或者资源集合中提取服务描述文档、相关服务发现算法等。为远程服务建立本地服务存根,这样就使得远程

服务和本地组件接口的表示形式相同,再对这些接口进行描述,并将这些描述插入用户服务描述数据库中,通过用户服务描述接口 (USDI) 来访问。

通过服务协作建模模块将服务描述转化为相应的服务功能单元,用户可以根据自己的需求,结合服务协作流程控制,重新组合服务。通过编译得到以一定的次序的服务

协作流程模型,最后将其转化为确定的 (无二义性的) 服务协作模型描述。

当服务组合模型转化为服务协作投影时,实际上为一个组态的程序。将所需的用户服务通过反射机制装载到内存中,通过执行有序的协作活动 (Cooperative Activity) 来完成系统功能。

3 基于 DSCM 的实时数据图形化的设计

3.1 服务协作代理 (SCA) 的设计

SCA 实际上为一个中间件,是协调本地服务和远程服务相互协作的中间代理 (见图 2)。服务注册是连接客户端和服务端端的桥梁,服务注册必须是客户端所能访问一个服务。在服务器端,服务分为显式服务和隐式服务。所谓显式服务则是对外界公开的,服务接收者可以直接利用;隐式服务则是要通过显式服务获得,并且通过显式服务来激活,通常这类服务需要安全验证的。

服务查询模块提供服务信息,服务定位模块将服务信息转换为相应的用户服务,并且将每个服务建立唯一的标识。可视化建模模块将每个服务对应的标识、接口信息、消息(方法)等信息转化为图形化模块,以便用户建模更直观。建立好模型之后,由服务定位模块为所选的远程服务建立本地存根,并且将服务协作模型进行投影。

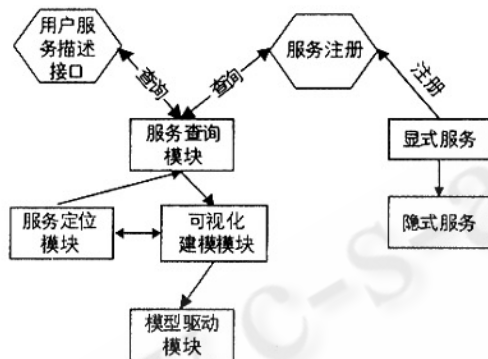


图2 服务协作代理的体系结构

模型驱动最主要的就是对模型的语法设计,而SCA就可以根据模型的配置,对模型进行分析和实例化,从而实现预定的功能目标。

首先,需要对服务进行说明和描述。以<Service>元素表示来描述服务,在<Service>元素中包含服务的名字等基本属性; <Info>元素作为<Service>的子元素, <Info>中说明该服务的具体获取的信息(包括具体组件的位置、调用的类或服务等等)。

再次,就是对协作模型的描述。服务的执行包括顺序执行<Sequence>和并发执行<Concurrency>,在执行中包括协作活动CA。

最后,就是对CA的描述。以<Activity>元素表示,包括调用的方法、调用的服务名、调用的参数等等。

3.2 基于SVG扩展的实时数据图形化服务的设计

实时数据是过程中某对象的瞬间采样值,通过这一系列离散的数据来模拟表示现实中过程数据。实时数据在不同时刻是变化的,如何将这种变化又有联系的数据更直观的展示给用户,直接关系到数据信息利用价值。图形化是一种可视化数据非常有效的方法,而实时数据是一种有联系的数据组成,所以采用矢量

图形的标记语言是一种很好的选择。目前流行的矢量图形的标记语言包括SVG和VML(Vector Markup Language)。SVG由W3C在融合了VML和PGML以后提出的,并且在2001年7月发布了SVG1.0规范。SVG和VML都是基于XML的,并且都是在表示图形信息同时加入了图形的显示信息,同时这两者都是支持动画交互的(VML通过脚本、SVG通过SMIL),这些都是非常适合流程监控的应用。而且通过对SVG的扩展,对其中的元素进行所需服务的自描述性,能更好实现服务的动态调用。

因为SVG是基于XML描述文档,所以只要将变化的属性与实时数据进行绑定,设定获取实时数据的频率,这样只传输实时数据值的集合,而不需要将整个SVG文档传送到客户端,就可以使得通过网络交换的数据大大减少,也减少了服务器的负担,提高了整个系统的性能。

实时数据图形化服务包括两个接口,一个为SVG图源接口,该接口以SVG文档为参数;另一个为实时数据调用接口,该接口是获取远程实时数据的接口。SVG解析模块是对SVG文档解析为文档对象模型(Document Object Model, DOM),该DOM树由DOM树维护模块进行维护。如果文档中有实时数据的元素,就将启动实时数据解析模块,该模块实时获取远程的数据。图元操纵模块更改DOM树的属性信息,这样使得SVG图形达到动态的效果。实时数据绑定模块是用户将某个图形元素进行实时绑定设置。各模块间的流程和协作见图3。

3.3 实时数据相关的服务说明

(1) 数据点配置服务。用户可以选择自己需要的数据信息,根据用户的身份,确定实时数据库中的权限。

(2) SVG文档生成服务。通过对数据点的配置选择,服务器根据实时数据点分布情况,动态生成相应的SVG文档。

(3) 实时数据发布服务。根据数据点的名称及其相应的属性,获取实时数据的值。

(4) 属性更改服务。动态更改所选择的数据点属性信息,如刷新时间、图像位置等等。

(5) 远程控制服务。通过相应的命令,对现场设备进行控制。

4 实时数据图形化的实例

模型驱动是通过使用形式化模型来支持系统描述和互操作的一种方法,旨在通过把系统描述背后的逻辑

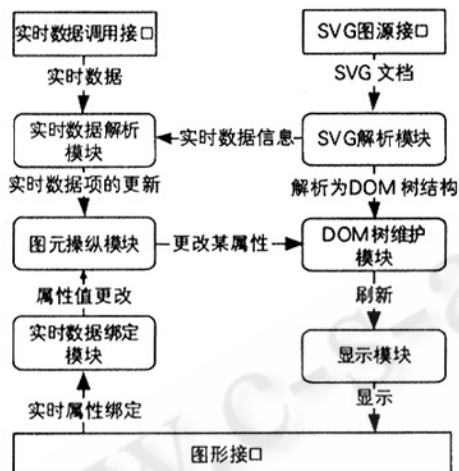


图 3 基于 SVG 扩展的实时数据可视化服务功能模块

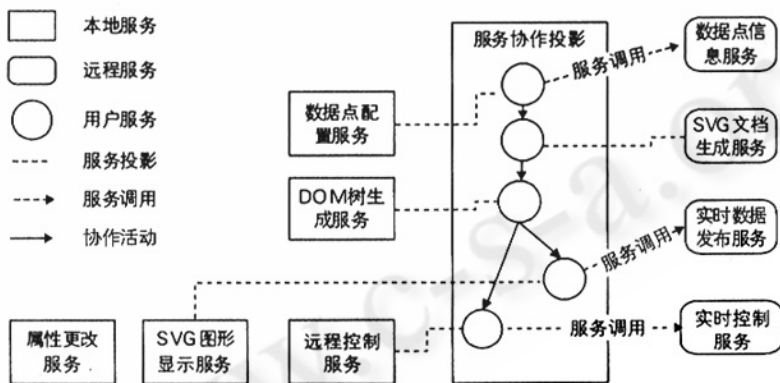


图 4 服务组合模型实例的例子

辑与它的具体实现相分离,以保证构造的信息系统能够适应新的硬件和软件平台。该方法最显著的特点在于,系统描述是与具体的实现技术和平台无关的。系统定义独立地存在于任何实现模型之外,同时又具备到许多可能平台基础设施(如 Java,XML,SOAP 等)的形式映射。

本文采用上述的服务进行动态服务协作构造(见图4),生成服务组合模型,通过基于XML描述组合模

型,并通过反射机制,生成服务协作实例。

5 结论

本文提出了动态服务协作流程,并且通过 SCA 来实现服务的动态重组,很好的适应快速变化的需求,在一定的服务集合中,可以灵活的重构不同的功能,大大提高了软件的可重用性;采用基于 SVG 扩展来进行实时数据图形化的描述,使得服务器和客户端间的平台无关性。但是,本文的模型驱动并不是基于标准的协议,在通用性上受到了一定的限制。在系统中,几乎很少考虑服务协作间的并发控制,所以在分布计算的并发控制有所欠缺。

参考文献

- 1 Min Luo, Mark Endrei, Philippe Comte, 面向服务的体系结构概述[C], <http://www.ibm.com>, 2004.
- 2 刘绍华、魏峻、黄涛, 基于服务协作中间件的动态流程模型[J], 软件学报, Vol. 15, 2004.
- 3 Object Management Group (OMG). Model - Driven

architecture [C], <http://www.omg.org/mda/> .
2001.

- 4 刘勘、周晓崢、周洞汝, 数据可视化的研究与发展 [J], 计算机工程, Vol.28, 2002。