

# SOA 架构思想在仿真实验系统中的应用研究

research on applying SOA to build Emluator System

谢俊 (武警杭州指挥学院 浙江 杭州 310000)

翁文勇 (浙江大学城市学院 计算机系 浙江 杭州 310015)

**摘要:**SOA 架构的软件系统因其计算的高效性、规模性和可扩展性,可以将系统组织成群组服务的模式,近来发展迅速。本文根据 SOA 架构思想的特点,设计了一个分布式仿真实验系统 SOAEmluator。

**关键词:**SOA 仿真实验 架构 建模

## 1 引言

当前,各种分布式技术的发展,特别是面向服务的体系结构(SOA)思想的提出,为仿真实验系统的设计提供了新思路。通过软件总线的方式建立仿真实验系统中心,并把各种仿真实验作为服务单元,通过 SOA 架构,建立分布式的仿真实验系统是一种值得推荐的思路。本文首先介绍 SOA 架构思想,并通过对一个分布式仿真实验系统(SOAEmluator)的分析设计过程的介绍,描述 SOA 架构思想在建构分布式仿真实验系统中的应用,并通过 UML 建模的方式对系统的各个节点进行描述。

## 2 SOA 架构思想

为提高计算的高效性、可用性、规模性,可以将系统组织成群组服务的模式,每一个服务都提供一组定义良好的功能集合;使得系统中的某些服务能够充分利用其他的服务的同时却无需考虑其物理位置。系统通过添加新的服务来不断的升级,这样面向服务的体系结构(SOA)就应运而生了。迄今为止,对于 SOA 还没有一个公认的定义,许多组织从不同角度和不同侧面对 SOA 进行了描述;W3C 将 SOA 定义为:“一种应用程序体系结构,在这种体系结构中,所有功能都定义为独立的服务,这些服务带有定义明确的可调用接口,可以以定义好的顺序调用这些服务来形成业务流程”。SOA 的关键是“服务”的概念,W3C 将服务定义为“服务提供者完成一组工作,为服务使用者交付所需的最终结果”。SOA 并不仅是一种现成的技术,

而且是一种架构和组织 IT 基础结构及业务功能的方法,是一种在计算环境中设计、开发、部署和管理离散逻辑单元(服务)模型的方法。它将应用程序的不同功能单元(称为服务),通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的,它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种这样的系统中的服务可以以一种统一和通用的方式进行交互。这种具有中立的接口定义的特征称为服务之间的松耦合。松耦合系统的好处有两点,一点是它的灵活性,另一点是,当组成整个应用程序的每个服务的内部结构和实现逐渐地发生改变时,它能够继续存在。SOA 架构体系可以用图 1 描述。

目前,SOA 的实现可以采用三种主要的技术思路:CORBA2, WSDL3, 服务代理注册方式。这三种方式中,WSDL 的 Web 服务的方式是我们目前最为关注的核心。所以运用 Web 服务和代理注册方式就成为主要的实现方案。Web 服务建立在开放标准和独立于平台的协议的基础之上,与开发语言无关。

## 3 一个 SOA 架构的分布式仿真实验系统——SOA Emluator

为对一个单位的所有仿真实验进行统一架构,我们设计一个 SOA 架构的分布式仿真实验系统——SOAEmluator。在忽略了安全控制等方面的因素后,我们提取得到 SOAEmluator 的主框架:一个

仿真实验注册中心、多个仿真实验服务节点,多个仿真实验终端,如图 2 所示。仿真实验注册中心:扮演 SOA 中的服务代理角色,该注册中心保存和管理所有仿真实验的 WSDL 描述,并提供接口供仿真实验服务节点注册、注销相应的仿真服务单元,还通过接口供仿真实验终端查询仿真服务。仿真实验服务节点:提供具体的仿真实验服务,每个仿真实验为一个节点,在注册中心注册服务后,可被仿真服务终端访问。仿真实验终端:用户通过该节点访问仿真实验注册中心,选择仿真服务节点,获取仿真实验服务,进行实验。



图 1 SOA 体系结构

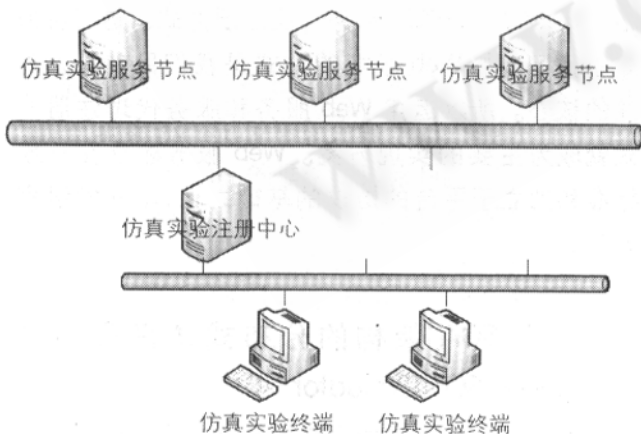


图 2 SOA Emluator 体系结构

### 4 SOA Emluator 系统架构和建模

根据 SOAEmluator 系统的体系结构,我们对系统的整体及其各个节点进行了设计。其软件总体结构可用图 3 表述。系统的软件模块包括:仿真实验注册中心(EmluatorPortal)、仿真实验服务节点(Emluator-Agent)、仿真实验终端(Emluator)。EmluatorPortal:存储所有仿真实验服务的描述信息和访问接口信息;EmluatorAgent:部署具体的仿真实验服务模块;Emluator:提供进行仿真实验的基础平台,并能够解析这个服务节点提供的内容。

#### 4.1 Emluator 设计

Emluator 的核心功能是访问 EmluatorAgent 提供的仿真实验服务,并根据与 EmluatorAgent 的交互,进行场景展现,最后给出实验结果报告。因此,Emluator 包括多个场景展现引擎 SceneEngine、多个实验报告引擎 ReportEngine、多个仿真驱动 EmluatorModule,如图 5 所示。其中仿真驱动模块由各个仿真实验节点提供。

#### 4.2 典型仿真实验过程

下面我们通过一个典型的仿真实验从开发部署到进行实验的过程,来描述系统实现功能的流程。图 6 为此过程的简化顺序图。首先由仿真服务提供者开发相应仿真服务节点软件模块和驱动模块,并根据 EmluatorAgent 规范建立相应的 Web 服务,同时抽象生成 WSDL 描述。服务节点开发测试完成后,利用通用的 EmluatorAgentManager 程序读取相应的 WSDL 描述,并通过仿真服务注册中心提供的 web 服务接口 EmluatorPortal 提出注册申请,由注册中心管理员审核并测试该仿真服务后,把此仿真服务纳入服务节点注册表中对应的目录中,此时该仿真实验服务处于活动状态,可以被仿真实验终端查找。

当有仿真实验用户要进行此仿真实验时,仿真实验终端程序首先向仿真实验注册中心发出查询请求,并获取该仿真服务的 WSDL。仿真服务终端获取该标准 WSDL 后进行解析,检查本地是否已经存在此仿真实验当前版本的驱动模块,如果不存在,则向仿真服务节点获取驱动模块,并在本地进行注册,同时完成仿真实验的一些初始化准备工作。

当仿真实验终端用户发出开始实验命令时,驱动

模块向仿真实验服务节点发出初始化实验请求,并获得可被某实验场景引擎解析的第一个场景描述,由场景引擎对该场景进行解析和展现;当此用户利用场景引擎完成该场景后,驱动模块收集用户行为和输入数据,并把这些行为和输入数据封装为访问服务节点定义的描述实体,提交到仿真服务节点,服务节点分析用户行为和输入数据后,向仿真实验终端发放下一场景或实验结果(如果上一场景为最后场景),该过程根据实验进程会反复执行多次。最后由仿真实验终端的某实验报告引擎根据服务节点反馈的结果描述信息给出实验结果。

### 5 结束语

分布式仿真实验是未来仿真实验的发展趋势,本文提出的基于 SOA 架构的仿真实验系统,降低了服务的提供者与使用者之间的耦合性。该设计把仿真实验的计算服务和仿真实验的展现、操作服务进行了分离,使得仿真计算服务集中于服务节点,便于升级维护和版本控制,以及软件保护。SOAEmluator 是一个架构方式和交互规范开放的体系结构,任何人和单位都可以在该开放标准下开发仿真实验服务节点及其驱动模块,可以大大加快仿真实验的发展和交流。另外,仿真服务节点可以收集一些实验数据等信息,

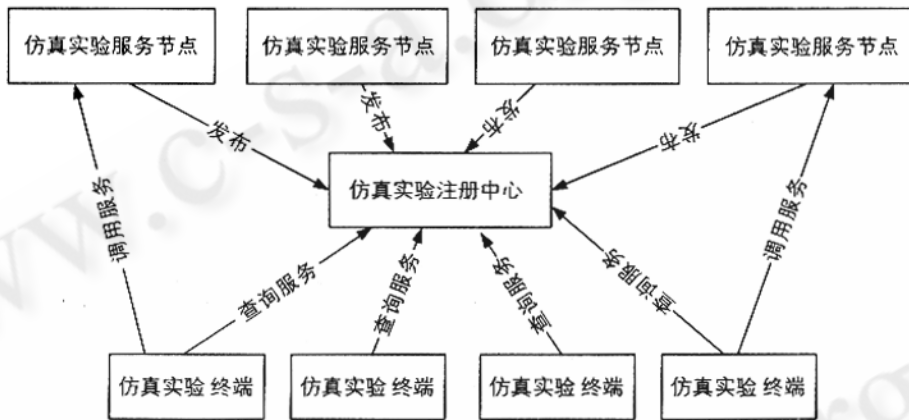


图 3 SOA Emluator 软件总体结构

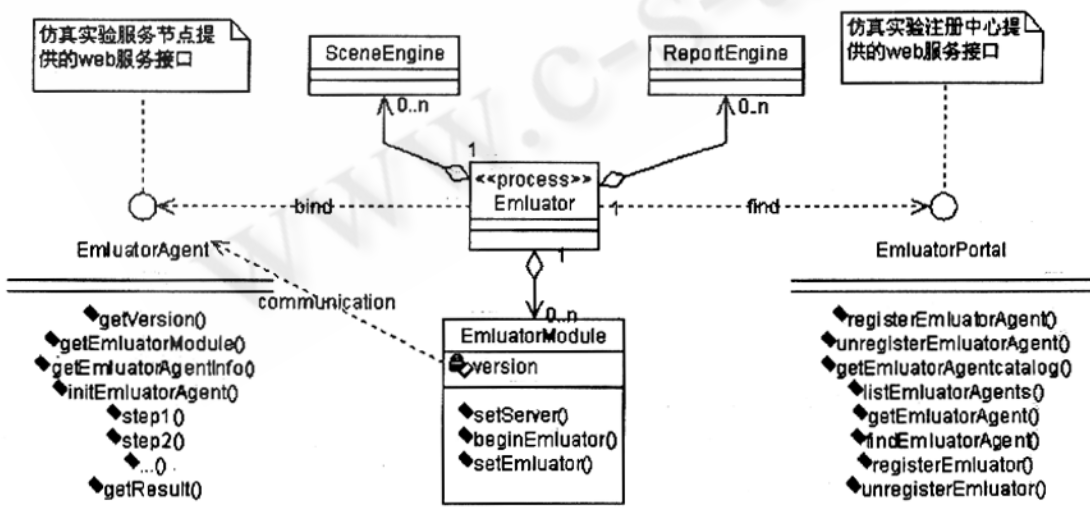


图 4 仿真实验终端类图

通过数据挖掘等辅助手段抽取有用信息,为仿真实验的改进提供参考依据。

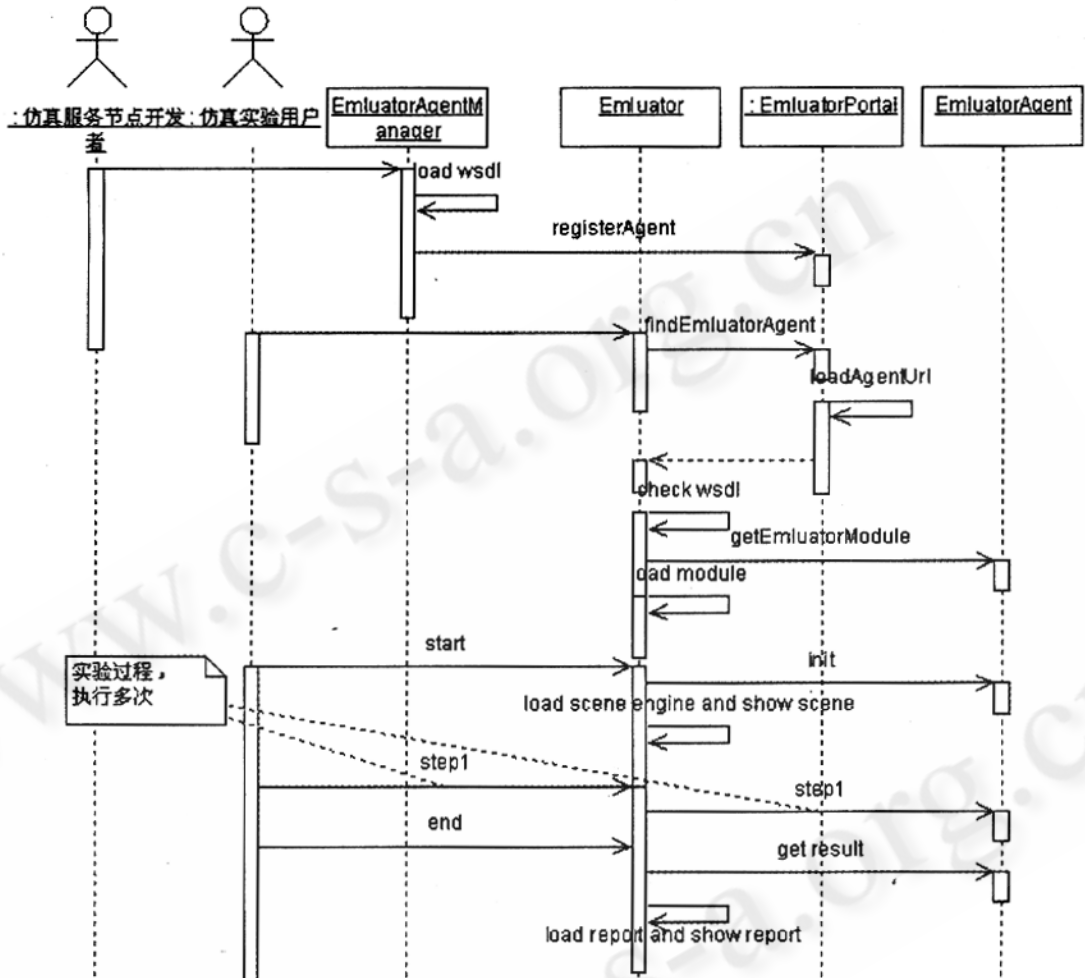


图 5 典型仿真实验过程

参考文献

- 1 孙永强, Web 服务深入编程, 清华大学出版社, 2002.
- 2 Peter Massuthe, Wolfgang Reisig, and Karsten Schmidt; An Operating Guideline Approach to the SOA
- 3 Yefim V. Natis ;Service - Oriented Architecture Scenario ;Gartner reacher AV - 19 - 6751
- 4 Bill Schneider; Design Patterns for SOA.
- 5 XML Schema Part 0: Primer, W3C Recommendation, 2 May 2001, <http://www.w3.org/TR/xmlschema-0/>.
- 6 W3C: The World Wide Web Consortium, 万维网联盟
- 7 CORBA ;Common Object Request Broker Architecture, 通用对象请求代理体系结构
- 8 WSDL: Web 服务的方式描述语言