

文章编号: 1005-8451 (2005) 11-0052-03

基于开放式分布处理的交易中介技术

谢伟鹏, 周家纪

(成都理工大学 信息工程学院; 成都 610059)

摘要: 在分析开放式分布处理(ODP)中交易技术核心思想的基础上, 通过应用面向对象的编程思想, 推理并导出交易中介关键部分的类Java实现方法。包括: 接口空间, 接口引用和交易中介的接口。在该交易中介技术支持下, 处于开放式分布处理系统中的客户可以通过简练的调用接口, 动态地发现和访问潜在的可用服务, 使分布式系统提供的服务具有位置透明性, 可相互操作; 解决传统方法提供服务的静态性。

关键词: 开放式分布处理; 交易中介; 交易功能; 接口类型; 接口引用

中图分类号: TP301 文献标识码: A

Technology of trading based on ODP

XIE Wei-peng, ZHOU Jia-ji

(College of Information Engineering, Chengdu University of Technology, Chengdu 610059, China)

Abstract: The essential of trading in ODP was introduced, and the implementation considerations of its key components were analyzed, which involved interface space, interface reference and trading interface. Depending on ODP trading, a client might dynamically discover and access the potentially available server in Open Distributed Processing System.

Key words: open distributed processing; trading; trading function; interface type; interface reference

“异构性”是分布式处理系统的一个重要特征, 也是构建时的技术难点。其中有大量的异构软件, 异构硬件平台和异构网络等。如何使用户透明地获得其中的服务并实现互操作是开放式分布处理(ODP)的主要任务。

1 ODP 中的交易中介

ODP 中的所有软件实体是以对象的形式存在的, 任何一个对象都可以向其他的对象提供服务和请求服务。各用户是通过调用服务方提供的对象接口来获得对方的服务^[1]。交易中介是个特殊的对象, 它能向客户提供潜在服务方的接口引用。图 1 给出了交易中介与客户的交易过程。

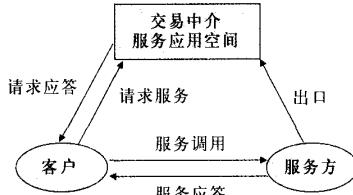


图 1 客户交易中介交易过程

交易中介通过服务方出口的服务提供服务, 客户在需要时动态地向交易中介申请所需的服务, 交易中介根据请求, 检索所辖的服务提供空间, 成功后向客户返回服务方的相应接口引用, 通过该引用, 客户可以调用该接口而获得所需的对象服务。

2 交易中介分析

根据上述交易中介的思想, 设计服务提供空间是实现交易中介的重要部分。其中的每项代表一个具体的对象, 它包括该对象提供的服务(接口)和访问该对象所需的引用信息(接口引用)。接口是接口类的实例。接口类是用接口描述语言来说明的(如IDL)^[3], 用于规定支持该接口对象必须提供的服务。

在服务提供空间里, 交易中介向客户提供了服务所需的信息。然而交易中介应向客户和服务方提供哪些操作使之分别获得服务和提供服务呢? 由图 1 可知。在交易中介的接口中必须提供两种最基本的操作: Export()和 Import(), 服务方利用 Export()向交易中介出口服务并存储在服务供应空间的适当位置; 客户利用 Import()向交易中介请求服务, 并根据客户提供的请求服务信息检索服务供应空间, 成功

后返回其中一个适当接口引用给客户。

3 接口空间

现在由交易中介的功能和接口空间的作用来描述接口所需的数据结构, 该结构不同于 RPC (远程方法调用) 交易中介是类型检索接口空间, 因为不是按名字查找, 使客户能发现系统中潜在的可用服务。因此该数据结构应该包含接口的类名, 基于可重用的考虑, ODP 支持接口的兼容关系。其思想如下: 如果接口 X 是接口 Y 的子类, 则满足如下的条件: (1) 接口 X 包括接口 Y 所有的方法。(2) 若两接口有同名方法, 则它们必须有相同的方法签名。依照面向对象的思想, 若接口空间内找不到父类接口则可以返回子类接口, 达到提供同样服务。由此为了提高检索的效率, 可以在数据结构中提供接口父类的接口引用。综合以上分析, 数据结构如下:

```
struct Interfacespace
{ string      Typename ; /*类名*/
  interfacereference Referencename ; /*接口引用*/
  string      Surpertype /*父类名*/
}
```

3.1 接口引用

为获得服务方服务, 客户首先要和提供接口的服务的对象建立连接。在 OSEWARE 中, 底层连接器完成了该任务 (该论文中不分析), 客户获得接口引用后, 以 API 的形式调用连接器功能, 完成连接^[2]。为此, 连接器必须能从客户方获得的信息应包括: 建立连接时所用的底层协议 (如 RMI—II OP) 和目标对象网址。客户请求所需的数据结构如下:

```
struct Interfacereference
{ string  Protocolidentifier[n]; /* 建立连接所需协议 */
  int     Protocoladdress /*建立连接所需的网络地址 */
}
```

3.2 交易中介接口

交易中介中的两个基本操作: (1) Export()从服务方接收接口类名, 接口父类名和该接口的引用, 并按上述数据结构实现接口, 然后放入交易中介的接口空间; (2) Import()从客户接收接口类名, 查找接口空间, 并在成功后返回其中符合客户要求的一

个接口引用。如下是对交易中介的类 Java 实现:

```
public interface Interfacespace/*两种操作方法
通过 Java 的接口来实现 */
{
  public boolean Export( string Type, string
  Surpertype , interfacereference Refer);
  public boolean Import ( string Type, string
  Policy, interfacereference Rusult );
}
```

```
public class Trader implements Interfacespace /*交易中介类 */

```

```
{private Interfacespace facespace=new
Interfacespace();/* 接口空间的引用 */
  public boolean Export( string Type, string
  Surpertype , interfacereference Refer);
  {
    if (leagel(Type ,Surpertype))/* 判定接口类与父
    类的合法血缘关系 */
    {
      facespace.Typename=Type;
      facespace.Surpertype=Surpertye;
      facespac.Referencename=Refer;
      putintospace(facespace); /* 将接口放入接口空
      间 */
    }
    else return false;
    exit(-1);
  }
}
```

```
public boolean Import ( string Type, string Policy,
interfacereference Rusult )

```

```
{
  interfacespace Match=new Interfacespace();
  /* 用于在分布式系统中所有匹配的接口引用的
  存储 */
  int space_count=0 , match_count=0;
  while (Interfacespace[space_count]!=null)
  {
    if((match(Type,interfacespac[space_count])or
    (Surpertype_match(Type,interfacespace[space_count]
    ))))
    {match[match_count]=interfacespac[space_count];
    }
  }
}
```

```

space_count++;
match_count++;
}/* 检索接口空间查找匹配的接口 */
else space_count++;
return_refer(Policy,Match,Result);/* 按客户要求
(Policy) 返回其中提供同类服务的某个接口的引用 */
}
}
}

```

4 结束语

根据 ODP 标准规范中有关交易中介的核心思想,有效地解决了分布式服务的透明性,使客户具

有动态地发现潜在的服务,经过仔细的分析和推理导出了交易中介的类 Java 实现。

研究和开发人员可以以该中介为基础作其他分布式处理系统应用类型的研究和开发;不足之处是该交易中介还未实现多中介的联合交易功能,可参考网络广播和 EJB 中的存根—骨架的技术来实现。

参考文献:

- [1] 孙亚民. 构建面向对象的应用软件系统框架 [M]. 北京: 机械工业出版社, 2004.
- [2] Rima Patel Sriganesh. Mastering Enterprise JavaBeans [M]. 北京: 电子工业出版社, 2005.
- [3] 吴越胜. Java 现代软件开发技术 [M]. 北京: 清华大学出版社, 2003.

“中国铁路客票发售和预订系统 V5.0” 获 2006 年度中国计算机学会王选奖

2006 年 10 月 26 日,中国计算机学会在京举行的计算机学会 50 年纪念会上颁发了 2006 年度中国计算机学会王选奖。颁奖仪式上,陈 錢女士(已故计算机科学家王选院士的夫人)和中国计算机学会理事长、中国科学院院士李国杰共同为获得 2006 年中国计算机学会王选奖的项目成果颁发获奖证书和奖杯。在铁道部领导下,由铁道科学研究院电子计算技术研究所负责研发的“中国铁路客票发售和预订系统 V5.0”荣获了 2006 年度中国计算机学会王选奖二等奖。据了解,本年度获王选奖一等奖的项目共有 2 个:由中国科学院计算技术研究所研制的龙芯 2 号增强型通用处理器、由清华大学和紫光比威共同开发的 IPv6 核心路由器;二等奖 1 个:中国铁路客票发售和预订系统 V5.0;入围奖共有 3 个:由清华大学完成的人类基因组重组热点的计算分析论文、由北京大学开发的大学课程在线、由浙江师范大学计算机科学研究所研制的便携式警用滚动指纹采集处理比对一体机。

为鼓励科技进步和科技创新,发现和激励创新型科技人才,使更多的科技人员多出成果,出好成果,服务于国民经济,中国计算机学会自 2005 年设立“中国计算机学会创新奖”,2006 年为纪念已故计算机科学家王选院士更名为“中国计算机学会王选奖”。此奖项的设立是中国计算机学会作为一个学术性社会团体进行学术评价的重要内容,重在奖励获得公认的计算机及相关领域的理论研究成果、重大技术发明、重大工程成果和具有显著社会效益的应用成果。中国计算机学会王选奖每年评选一次,由学会理事、高级会员、各专业委员会、省级地方学会等推荐,由专家组成的评审委员会经两轮评出。

文/摄影/本刊记者 国 敢

《铁路计算机应用》期刊 被英国 INSPEC 数据库收录

经过多项学术指标综合评估及国内外权威专家评议推荐,日前,《铁路计算机应用》期刊已被英国 INSPEC 数据库收录。英国 INSPEC 数据库,全名为《物理学、电技术、计算机及控制信息数据库》(Information Service for the Physics, Electrotechnology, Computer and Control 简称 INSPEC),是目前全球在物理和工程领域中最全面的二次文献数据库之一,它的前身是“科学文摘”(Science Abstract or SA, 始于 1898 年),曾被誉为国际六大检索期刊。它提供涵盖:理论及应用物理,电气和电子工程,计算机科学,控制技术,通讯与信息技术,生产和制造工程等专业的科学技术文献检索,并对涉及光学技术、材料科学、海洋学、核能工程、交通运输、地理、生物医学工程、生物物理学和航天航空等领域也有很广泛覆盖。它是工程技术领域最受欢迎的文献数据库之一。

据了解,英国 INSPEC 数据库其数据来源于全世界上 80 多个国家的 3 400 多种科学与技术期刊、2 000 种会议录以及大量的著作、报告和论文。INSPEC 的所有文献都含有目录和摘要,数据并以每周的速度更新。从 1969 年至今,INSPEC 数据库含有近 800 万条文献,并且以每年 40 万条新文献的速度增加。INSPEC 除了以它广而深的学科覆盖,准确的目录标引广受使用者欢迎外,还以它专业而完善的主题索引机制而著称。其中包括:自由词/重要概念索引,INSPEC 叙词表中的控制词索引,INSPEC 分类系统的分类编码索引,处理代码索引,化学物质控制词索引,航空航天对象索引。任何使用 ISI Web of KnowledgeSM 平台的用户都可以通过 Cross Search—Form 使用跨库联合检索功能检索订购的相关资源和免费提供的 13 外部学术信息资源。

文/朱 诚 国 敢