

TAO 在舰载指控系统中的应用研究

陈靖, 曹万华, 赵恒

(武汉数字工程研究所, 武汉 430074)

摘要: 在舰载指控领域中, 基于 TAO 进行构件化的软件开发, 由于缺乏构件开发和运行时管理等工具的支持, 因此增加了应用软件开发与管理的复杂性。该文针对舰载指控系统的应用特征, 对 TAO 进行了适应性改造和扩充, 形成一套构件化开发和运行的集成环境, 以简化舰载指控应用构件的开发, 增强其灵活性及运行效率。

关键词: 舰载指控系统; 构件管理; 构件集成开发环境

Application Research of TAO in Shipborne Command and Control System

CHEN Jing, CAO Wan-hua, ZHAO Heng

(Wuhan Digital Engineering Institute, Wuhan 430074)

【Abstract】In Shipborne Command and Control System(SCCS), development software based on The ACE ORB(TAO) increases the complexness of development and management of application software, due to lacking in support of development's and runtime management's tools. According to the characteristics of SCCS, TAO is adaptably reconstructed and extended. A component-based integration development and runtime environment is built to simplify SCCS software development and strengthen the flexibility and runtime efficiency.

【Key words】Shipborne Command and Control System(SCCS); component management; component-based integration development environment

1 概述

近年来, 分布实时嵌入式系统(Distributed, Real-time, and Embedded systems, DRE)得到了广泛的应用。基于构件的开发(Component-Based Development, CBD)方法在通用软件开发领域已广泛应用, 如何在 DRE 系统中应用 CBD 方法来提高软件开发效率已成为当前研究的热点。舰载指控系统是一类典型的 DRE 系统。随着作战需求的快速变化和扩大, 舰载指控系统的软件规模、复杂度急剧膨胀, 要求容错能力、实时性能和动态重组能力进一步提高, 同时对软件的研制效率、研制质量、研制周期等也提出了更高的要求。

TAO是美国Washington大学的Douglas C. Schmidt教授领导开发的开源、实时的ORB产品, 具有很强的实时特性, 已在国防和通信网络领域获得广泛应用, 是美国国防部和OMG组织推荐的一个CORBA标准实现^[1]。

2 TAO 和舰载指控系统简介

TAO 建立在操作系统和 Adaptive Communication Environment(ACE)之上, 基于 CORBA3 规范, 使用了 ACE 中提供的框架结构对象与模式, 实现了针对高效与实时系统的中间件架构, 可满足 DRE 系统的高性能要求。与其他的ORB相比较, TAO 具有如下明显的优点:

(1)源代码开放, 开发社区由当前分布式计算界著名的 Douglas C. Schmidt 教授领导, 实力强大, 当前开发很活跃, 版本更新升级快;

(2)基于实时 CORBA 的设计要求, 提供高效、可预测、且能实现端到端的服务质量(Quality of Service, QoS)配置;

(3)设计和实现过程中, 大量采用优化的软件设计模式和方法, 是一个高效并且实时的 ORB;

(4)TAO已在嵌入实时领域, 如航天和国防、通信与网络管理等获得了广泛的应用, 经实践检验是一个高效成熟的 ORB产品^[1]。

舰载指控系统是海上指挥、控制、通信、计算机、情报、侦察和监视(C4ISR)系统的核心部分, 是一种具有高可靠性和实时性的复杂信息处理与控制系统, 采用全分布式体系结构。如图 1 所示, 它输入来自信息探测设备和数据传输设备等各类信息, 经过信息融合和战术辅助决策, 将舰载武器系统状态和战场综合态势呈现给指挥员, 并接收指挥员下达的各类操控命令实施作战指挥。

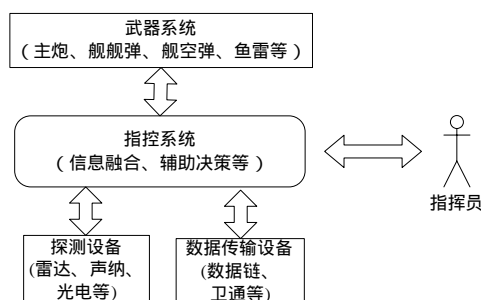


图 1 舰载指控系统的结构简图

指控系统内部及与其他舰载设备间的信息交换遵循作战系统信息交换协议(Combat System Message eXchange Protocol, CSMXP)。CSMXP 采用二进制字节流报文格式, 系

作者简介: 陈靖(1977-), 男, 硕士, 主研方向: 软件工程, 软件开发, 构件化技术; 曹万华, 研究员; 赵恒, 博士

收稿日期: 2007-06-19 **E-mail:** redjing@gmail.com

统运行效率高,但系统间接口紧耦合,报文的任何改变都会引发系统软件的修改,而且二进制字节流报文的软件实现容易出错、调试困难、开发周期长。随着计算机硬件技术的发展,在不影响系统性能的前提下,对指控系统接口应用层协议抽象层次进行提升,由二进制字节流报文格式提升为CORBA接口,可以极大地提高系统的开发效率,同时通过接口的动态组合可提高系统的动态重组与升级能力。

3 对TAO的改进

CORBA标准涉及分布式计算的功能和性能等各个层面的需求,对于舰载指控系统这样的特殊嵌入实时应用环境来说,显得过于庞大,需要进行裁减。TAO仅仅是一个运行时ORB核心,对于软件开发过程的支持较弱。而软件构件化开发是一个复杂的过程,为了简化构件化开发的复杂性,必须对TAO进行改造,增加支持构件化开发的辅助工具。

3.1 TAO的优化

TAO ORB的调用路径如图2所示。

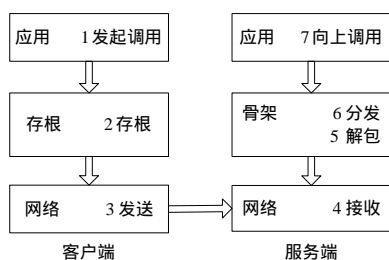


图2 ORB调用路径示意图

通过对TAO ORB调用路径上的关键点进行插桩测试,发现产生性能损失主要来自并置对象(collocated object)访问、内存管理、服务端请求向上分发、以及表示层的转换等方面的因素,由此可采用基于预先分配和静态固定的思想对TAO内核进行优化^[2]。

(1)并置对象访问^[3]。CORBA对象都是位置无关的,通常采用远程对象方式进行访问,需要经历一系列复杂的过程(连接选择、参数序列化、方法分发等)。但是当访问对象是本地并置对象时,这些过程就显得多余和低效。因此,在ORB处理对象寻址时,把并置对象和远程对象区分开,对并置对象采用特殊的访问方式来优化性能。当访问双方位于同一地址空间时,直接把对象指针传给客户端方调用;而不在同一地址空间但位于同一机器时,采用本地间进程通信的方式来访问并置对象。

(2)内存管理。内存缓冲区被分配用来进行参数编组和解编。在对象请求/应答过程中,ORB会频繁地对内存池请求分配和释放内存,当多个线程对内存池访问时就存在竞争。为了减轻内存池的分配竞争,采用线程本地存储(Thread-Specific Storage, TSS)技术来优化内存池的管理。TSS使多个线程可以通过一个逻辑的全局访问点来获取线程特定的数据而不会引起访问竞争。因此,使用TSS内存池消除了分配锁的开销,减小了内存碎片。

(3)服务端请求向上分发。服务端ORB在收到请求后经过一系列的查找过程,最终将请求传递到服务实施代码。分发请求的时间依赖于POA层次的组织,POA、servant和服务对象方法的数量,以及对各种表分发查找线程的竞争等。常用的分发查找策略包括线性搜索和二分法搜索等,线性搜索在表的规模较大时效率较低,二分法对表的插入删除操作复杂。为了优化分发查找的过程,采用动态哈希(dynamic Hash)

的策略,该分发查找策略的耗费可以达到常量级,具有很好的可伸缩性。

(4)表示层转换^[4]。指消息的打包和解包,这是由客户端的存根程序和服务器端的骨架程序完成的。优化表示层就是优化IDL编译器,减少打包解包时的内存消耗,使其产生高效的存根和骨架代码。

3.2 TAO的裁减

舰载指控系统对实时性具有较高的要求,但对系统的动态性要求不是很高。为了使CORBA技术较好地适应指控系统应用,需要对CORBA提供的技术平台进行裁减,把那些影响系统性能、增加系统代码的部分进行裁减。可从2个方面对标准CORBA进行裁减:动态CORBA的部分和不必要的CORBA服务。

CORBA标准包括静态和动态的两部分,每部分都提供了等同的功能。静态编程在CORBA客户程序访问服务对象时,对象实现操作及操作中的参数在编译时就确定下来;这要求客户程序必须包括由IDL编译器为接口所产生的存根(stub)代码,服务程序必须包括产生的骨架(skeleton)代码。而动态编程在客户程序编译时不能确定所访问对象的类型,在运行时才获得对象的接口信息,然后动态组织参数调用。使用静态编程时,性能要优于动态编程,但灵活性不如动态编程。对于舰载指控系统,运行时性能的重要性明显大于系统的灵活性。因此对舰载指控应用系统来说,可以裁掉CORBA系统中的动态部分。

CORBA标准提供了数量众多的标准服务供应用程序使用。对于舰载指控系统,为了实现服务的查找,命名服务是必须的;对于作战报文和命令的多点分发这种多对多的通信模式,通告服务提供的发布/订阅的通信模式为其提供了很好的支持,因此通告服务也是必须的;指控系统是一个对安全级别要求很高的系统,CORBA的安全服务提供了很好的支持,因此安全服务也是必须的。除此以外的其他CORBA服务,例如交易服务、事务处理服务等均可以进行裁剪。

3.3 TAO的构件管理支持扩展

在CORBA编程中,实现IDL定义接口的构件只是一个静态的对象,把实现对象绑定到ORB并启动服务监听还需要程序员编写额外的启动main函数,CORBA规范对于构件的运行实例也没有相应的监控应用程序编程接口(Application Program Interface, API),无法在运行时对构件实例进行替换,构件版本的管理也依赖于程序的具体实现。这些都加大了编写可替换性好、灵活性高的构件的难度。传统的CORBA规范缺乏一个构件运行时管理的方案。

虽然TAO提供了一个CCM规范的实现CIAO,但该实现还处于实验阶段,而且CCM本身就是一个异常复杂庞大的解决方案^[5]。因此,对TAO进行扩展,实现一个高效简单易用的构件管理方案就非常必要。

构件管理的主要功能如下:

(1)构件二进制代码的装载和构件实例的启动激活。常规的CORBA构件开发流程为:编写体现构件功能接口的IDL描述,用IDL编译器编译IDL描述生成存根和骨架,由骨架类继承实现伺服类,编写进程入口main函数,绑定伺服类到ORB并启动ORB监听循环。该方式把构件代码静态联编成一个整体,不能满足构件动态管理的要求,需要建立一个构件装载器和启动器使构件代码的装载和启动过程自动化。

(2)构件运行状态的监控。管理器需要在构件运行过程中

知道构件的运行情况以做出构件的配置调度决定。这些构件的状态信息包括作战系统网络中已启动了哪些构件、某个构件配置在哪个节点、构件当前运行负荷如何、构件的资源消耗和利用情况等。

(3)构件运行时资源的分配与回收。管理器可以灵活启动或关闭构件，这要求对构件运行时的资源进行及时的分配与回收。其中最重要的资源就是线程和内存的管理。

(4)构件依赖关系管理。构件虽然是系统一个相对独立的组成部分，以服务的形式对外提供相应功能，但构件之间也需要相互协作来完成系统的整体功能。构件间可能存在一些依赖关系，当构件数目越多时，这种依赖关系越复杂。因此，需要对这种依赖进行有效的管理。

(5)提供构件必要的运行时公共服务支持。在面向舰载指控领域的构件中，很明显存在一些公共功能要求，如作战报文收发、定时器管理、CORBA 事件收发和共享数据管理等。

(6)在网络节点单点失效情况下的构件管理。网络环境下“单点失效”问题是不可避免的，因此必须具有探测崩溃节点并采取相应措施使之从错误中恢复的容错机制，使运行于崩溃节点上的构件能在其他节点上重新启动，并通知其他节点。

按照分级管理的思想，将构件管理设计为 4 层管理结构，如图 3 所示，包括域、节点、容器和构件。

(1)域管理器：负责制定和执行应用网络中的全局管理和配置策略，在整个应用系统中是唯一的，维护全局信息，直接管理各节点上的节点管理器，充当节点管理器的指挥者和仲裁者。

(2)节点管理器：负责制定和执行本节点的管理和配置策略。它直接管理容器，负责将构件的管理指令传递到容器。

(3)容器：构件运行的包容环境，管理构件的生命周期，分发服务调用请求到构件，并通过容器上下文向构件提供通用服务。一个节点中可以有多个容器。

(4)构件：被管理的实体，对外提供应用功能服务。

(5)构件库：构件代码文件集中存放的仓库。用于构件的添加、删除、查询和代码文件的下载等。

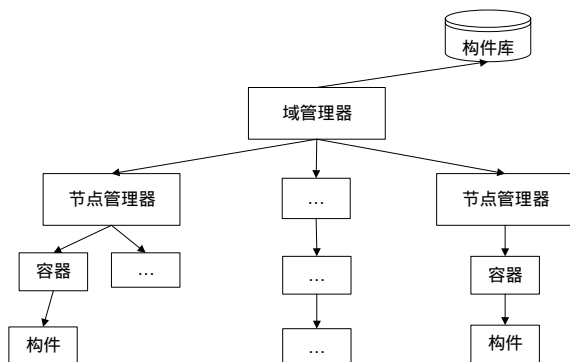


图 3 构件管理层次结构

3.4 TAO 的构件开发支持扩展

基于构件的软件开发是涉及整个生命周期的软件开发方法，TAO 目前实现的还只是构件运行时的核心支持，基本上没有支持构件化软件开发的外围工具。如果缺乏外围开发工具的支持，对于经验不够丰富的开发人员来说，编写 CORBA 程序是很困难的^[6]。为此，有必要在 TAO 的基础上研制一套构件开发工具集来降低构件开发的复杂性。

图 4 说明了构件开发过程中所涉及的角色、步骤、工具和成果。

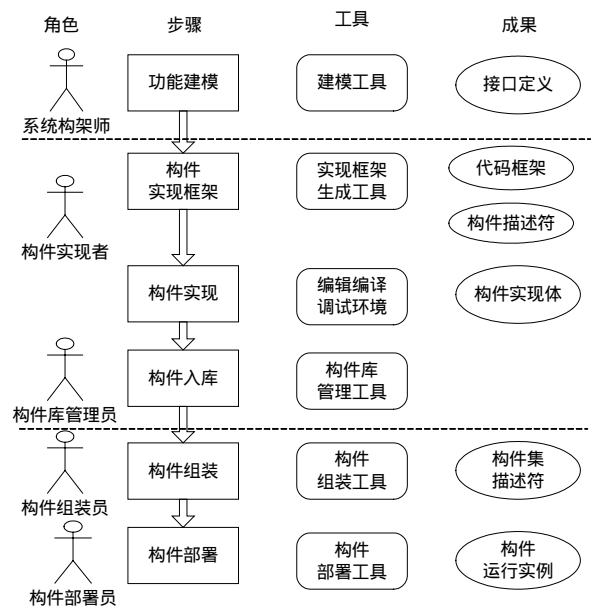


图 4 构件化开发过程

针对开发过程中不同的角色，支持构件化开发的工具主要包括：

(1)建模工具，供系统构架师使用。按照面向对象的思想分析和设计应用系统的功能，输出 IDL 接口定义文件。

(2)实现框架生成工具，供构件实现者使用。用于构件框架的代码自动生成，以及生成集成开发环境的工程文件。

(3)代码的编辑、编译和调试环境，供构件实现者使用。用于构件的实现。

(4)构件测试仿真工具，供构件实现者使用。该工具仿真构件的外部环境，方便开发者验证构件内部业务逻辑的正确性。

(5)构件库管理工具，供构件库管理员使用。用于构件入库、出库和管理等工作。

(6)构件组装工具，供构件组装员使用。以图示方式显示构件端口及构件端口交互的依赖和约束，通过构件端口间的连接关系组装应用，输出构件组装描述文件。

(7)构件部署工具，供构件部署员使用。它定义构件运行网络环境，包括所有网络节点以及应用中构件的运行节点，确定应用的物理运行环境，生成部署描述文件。

所有工具以 Eclipse 插件形式进行集成，形成集系统设计、代码编辑、调试和构件组装、部署为一体的构件集成开发环境，简化构件开发流程，降低构件开发的复杂性，提高构件的开发效率。

4 结束语

本文对 TAO 进行了适应性的改造和扩展研究，包括 TAO 的实时性能优化和测试、嵌入式裁减、运行时的构件管理扩展，以及构件开发工具扩展等。在舰载指控领域应用 TAO 进行构件化开发，不仅能在确保系统性能的情况下提高系统的开发效率，而且可使系统具有良好的动态组合性，便于扩展和升级。

参考文献

- [1] Schmidt D C. Overview of the ACE+TAO Project[EB/OL]. (2006-09-28). <http://www.cs.wustl.edu/~schmidt/TAO-Overview.html>.
- [2] 鲁波, 邹华, 王柏. CORBA 性能分析及优化[J]. 计算机工程, 2001, 27(9): 3-4. (下转第 242 页)