

基于 XML 的组员关系协议的设计与实现

余 勇¹, 黄志球¹, 刘 毅², 周安宁²

(1. 南京航空航天大学计算机系, 南京 210016; 2. 南京航空航天大学航空宇航学院, 南京 210016)

摘 要: 结合开发的协同设计系统, 给出了协作层基于 XML 的组员关系协议的设计与实现方法。分析了组员关系协议的通信行为, 并使用 UML 的面向对象方法分析与设计了请求、响应和通知协议, 详细说明了使用 XML Schema 实现通信协议的方法, 结合协议实例, 给出了基于 XML 的协议的打包和解析, 并证明了协议的可行性。该协议能应用于异构平台下多种协同应用系统。

关键词: XML; XML Schema; UML; 组员关系; 协议

Design and Implementation of XML-based Membership Protocols

YU Yong¹, HUANG Zhiqiu¹, LIU Yi², ZHOU Anning²

(1. Department of Computer Science, Nanjing University of Aeronautics and Astronautics, Nanjing 210016;

2. College of Airspace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016)

【Abstract】 Concerning the developed cooperative design system, a new XML-based method to design and implement membership protocols in collaboration layer is presented. The communication behavior of membership protocols is analyzed and with the aid of UML the typical requiring, responding and notifying protocols are designed through object-oriented concept. Based on object-oriented analysis of the protocols, the mechanism of protocols design following XML Schema specification is illustrated. The main flow of embedding the protocols is given by packing and parsing XML-based protocols in the cooperative design system prototype. This kind of protocols can be used in the other heterogeneous cooperative application platforms.

【Key words】 XML; XML schema; UML; Membership; Protocol

1 概述

计算机支持的协同工作(Computer Supported Cooperative Work, CSCW)是地域分散的群体借助计算机及网络技术, 共同协调与协作来完成一项任务^[1]。参与协同的群体成员间需要交换信息。由于协同工作的群体复杂性, 要求下层的通信系统支持多种通信模式, 包括一对一、一对多和多对多通信; 协同工作的协作复杂性, 要求通信系统能支持动态的组员关系管理, 支持群组通信。这里的组是计算机集合的逻辑概念。例如, 共同编辑共享文件的用户属于一个组; 参与多媒体会议用户为一个组。组员关系协议就是对协同工作中以组为单位的复杂组员关系进行管理的协议^[2]。

本文讨论的组员关系协议应用于协同设计与分析支撑环境(Cooperative Design and Analysis Environment, CDAE)。CDAE 是面向飞机产品的分布式协同设计系统, 采用多服务器多客户端、偏集中式体系结构; 使用 XML 描述操作语义信息并传输, 以实现细粒度的基于三维模型的透明协作等。CDAE 系统从功能上分为 3 个层次, 如图 1 所示, 组员关系协议位于协作层, 对参与协作的用户关系进行管理, 提高协作效率; 一方面调用通信层的信息传递功能, 同时为应用层提供服务。

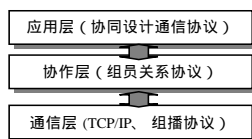


图 1 协同设计系统协议层次

以往的组员关系协议大多采用参数帧或对象流表示, 但

这 2 种方式都存在着缺点: 参数帧表示方法的模式过于限制, 且易造成数据空间浪费; 对象流表示方法需要服务器和客户端采用某种特定语言实现。XML 可以克服上述 2 种方式的缺点。使用 XML 描述协议模型, 直观易懂, 形式灵活, 不仅可以定义统一的格式, 而且能使用已有的 XML 解析器, 方便了协议的扩展和系统对协议的集成。目前已有很多使用 XML 作为通信协议的先例。本文采用 XML 设计组员关系协议的思想来源于此, 这里主要侧重于在协同设计系统中协作层的应用。

2 组员关系协议分析

CDAE 中采用集中式的组员关系服务实现组员关系管理。文献 3 给出了集中式管理的优点。组员关系服务实现各客户端本地组员关系视图的一致性和错误检测, 是客户端建立消息传递的基础服务, 客户端通过它获取其它客户端信息, 以进行消息传递。在这种 C/S 模式下, 交互过程描述如下:

(1) 当协同用户 A 执行组员关系操作, 如加入组或申请发言权等, 则 A 的本地应用工具的消息处理将组员关系操作写成请求信息, 通过本地的通信模块发送给组员关系服务。

(2) 接收到 A 的请求之后, 组员关系服务进行相应处理, 并向 A 返回响应信息, 作为对组员关系操作请求的答复。

(3) 如果组员关系操作执行成功, 组员关系服务在向 A 发出响应信息的同时, 向其他的协同用户发出组员关系操作通知消息以进行状态更新。

基金项目: 国家高技术“863”-706 重大专项子课题基金资助项目

作者简介: 余 勇(1982-), 男, 硕士生, 主研方向: CSCW;

黄志球, 教授、博导; 刘 毅, 副教授; 周安宁, 博士生

收稿日期: 2006-01-18 E-mail: gasyuyong@hotmail.com

(4)协同用户的消息处理器接收到响应信息和通知信息后，相应地修改本地协同工具的状态机。

因此，可将组员关系协议的通信行为归纳为 3 类：协同用户向组员关系服务发出的请求 Request，组员关系服务对协同用户请求的响应 Response 和向其他的协同用户发出的通知 Notify。

3 协议的 UML 设计

针对组员关系协议的请求、通知和响应协议，本文采用面向对象的 UML 方法建模，此方法能兼顾面向对象方法和 UML 的优点，而且 UML 类图可以方便地映射为基于 XML 的协议元数据，简化了后续工作。

3.1 请求协议的 UML 描述

在 CDAE 中，协同用户首先要向组员关系服务发出请求，构成组员关系请求协议以实现协作组的建立、加入、退出和组员发言权控制等，主要包含以下 3 方面：

(1)与用户相关请求，应用于协同用户本地组员关系初始化和独立工作方式，主要包括新用户登录请求 NewUserReq，退出请求 DelUserReq，获取当前在线用户列表 GetAllUsersReq，其中 NewUserReq 需要用户信息，因此需聚合用户信息 GcsUser。

(2)与组相关请求，应用于组操作请求和多人协同工作方式，主要有请求新建组 NewGroupReq，请求关闭组 DelGroupReq，请求获取当前组列表 GetAllGroupsReq，用户加入组请求 NewGroup MemberReq 和用户退出组请求 DelGroupMemberReq，其中 NewGroupReq 聚合组信息 GcsGroup。

(3)与发言权相关请求，应用于多人协同工作时的组内发言权控制，主要有请求发言权 RequestFloorReq，授予发言权请求 GrantFloorReq，剥夺发言权请求 DepriveFloorReq 和释放发言权请求 FreeFloorReq，这些都需聚合发言权 Floor。

根据上面对请求主要内容的分析，图 2 给出了请求协议的 UML 描述。请求协议开始于类 MembershipRequest，它是一系列请求的容器，包含优先级属性 priority 和记录请求时间的 timeStamp 属性。抽象类 Request 是具体请求的超类，每一个请求具有一个 requesterID 标记。各个具体请求如 NewUserReq 等从 Request 派生。

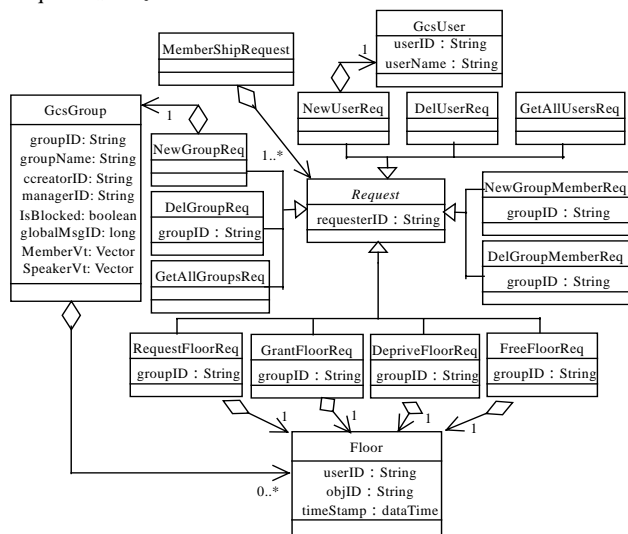


图 2 请求协议的 UML 描述

图中的用户类 GcsUser，组类 GcsGroup，发言权类 Floor 是公用类结构，在响应、通知协议及组员关系服务中都有用到。此外，响应协议描述了组员关系服务对用户请求的响应，形式与请求协议类似，这里不做介绍。

3.2 通知协议的 UML 描述

对于用户的请求，组员关系服务在响应请求的用户同时，需要将变更通知其它用户。通知协议用于实现这一功能，如图 3 给出了通知协议的 UML 描述，始于类 MembershipNotify 和抽象类 Notify，各派生类具体内容如下：

(1)用户相关通知，主要有新用户登录通知 NewUserNot，退出通知 DelUserNot。

(2)组相关通知，主要有新建组通知 NewGroupNot，关闭组通知 DelGroupNot，用户加入组通知 NewGroupMemberNot 和用户退出组通知 DelGroupMemberNot。

(3)发言权相关通知，主要有用户请求发言权通知 RequestFloorNot，授予发言权通知 GrantFloorNot，剥夺发言权通知 DepriveFloorNot 和释放发言权通知 FreeFloorNot。

(4)此外还有用于新用户加入组时同步组员消息的组员消息同步通知 SynchronizeMsgNot 和新用户登录时离线消息通知 OffLineMsgNot，其分别聚合了应用层点对点应用消息 P2pAppMsg 和组消息 GroupAppMsg。

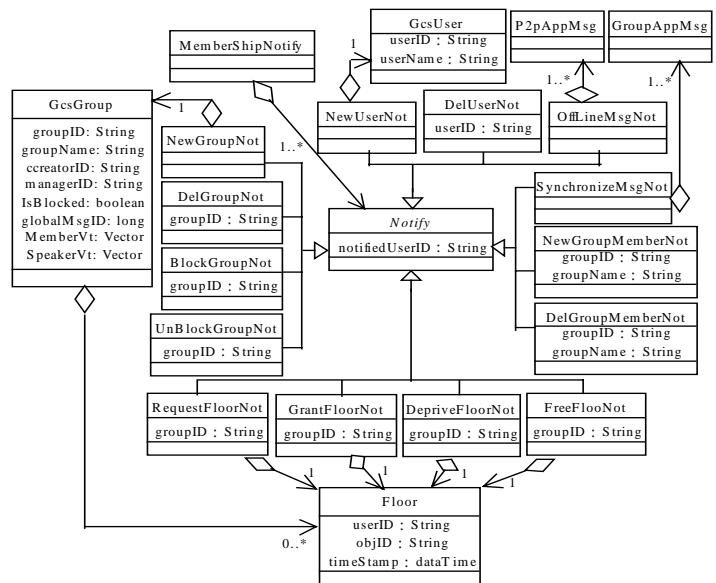


图 3 通知协议的 UML 描述

3.3 UML 描述方法总结

从图 2 和图 3 可以看出，组员关系协议的结构因为采用了类继承和类聚合而具有良好的可扩展性，这体现在：

(1)面向对象本身对属性和方法的封装易于协议模块的稳定，对协议中一个类内部属性的修改不牵涉其它属性和类。

(2)类继承对一类相似的请求或通知作了高层次的概括，提取出公有的属性，利于协议表达的简洁；协议在实现的初期随着应用的扩展和功能的深入必然会进行扩充，继承类的添加过程独立于其他兄弟继承类，最大限度地保留了原有协议的有效性。例如，在 Request 抽象类下面添加获取某一组内所有组员的请求 GetAllUsersInGroupReq，继承类只需定义 GetAllUsersInGroupReq 类本身即可。

(3)类聚合表达了一个请求等的具体内容，而这正是协议中频繁变动的地方。使用公用类结构描述具有很好的灵活性。当需要引用或者修改内容时，只要置换协议中的公用类就可以了。如请求、响应和通知协议中都使用的发言权类 Floor，是对应具体对象的发言权，目前的设计中发言权控制的是对象整体，如协同编辑中整篇文章档，协同设计中的图形文件等，协同粒度较大，改进的方法是降低粒度，如把协同编辑中文档的段为单位进行发言权控制。因为其它类聚合了发言权类 Floor，日后对 Floor 的修改将不会改变协议的其它部分。

4 基于 XML 的组员关系协议实现

如果能定义协议的 XML Schema，也就定义了基于 XML

表达的协议的元数据。使用元数据规范化协议，能从数据模式的抽象层次定义基本的协议表达，方便协议验证；对协议的元数据的修改，能够完成一类协议的重定义，有利于协议集的更新和扩展。因此，协议的具体实现包括将 UML 设计映射成 XML Schema、基于 XML 的协议验证、打包和解析等。

4.1 UML 到 XML Schema

基于文献[4,5]中给出的映射规则，UML 类图可以方便地转换到 XML Schema。上节中组员关系协议的 UML 类图转换后的 XML Schema 共 481 行，且其中类型和元素的定义遵循通用的 XML 文档描述规则，限于篇幅，这里不再列出。

4.2 协议的验证、XML 打包与解析

通过定义协议的 XML Schema 给出了协议的元数据描述，基于 XML 的组员关系协议要严格遵循 XML Schema 中的类型和元素的定义，因此使用 XML Schema 可以验证协议的有效性。然而要真正实现 CDAE 集成基于 XML 的协议，必须在 C/S 两端将具体的组员关系操作打包为 XML 数据流，并且相应地能够将 XML 数据流还原为原操作。

下面以请求协议的处理过程为例详细描述基于 XML 的组员关系协议的打包和解析方法，如图 4 所示，首先获取协同客户端的组员关系操作，根据请求协议的 UML 对象类建立内存中相应的协议对象树，然后使用转换程序将对象树打包成符合请求协议的 XML Schema 规范的 XML 数据流。数据流通过 C/S 之间的 Internet 到达组员关系服务并被接收下来，经过转换程序解析、重新形成请求协议的对象树。组员关系服务识别出具体请求，将其分派给相应处理模块。可见，请求协议处理的完整过程包含一次 XML 打包和解析。

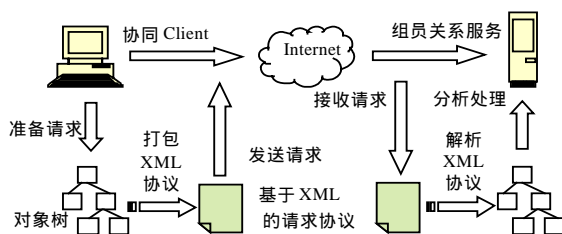


图 4 请求协议的 XML 打包和解析流程

类似情况，从组员关系服务到协同客户端的响应和通知协议要经过服务器端的 XML 打包和客户端解析过程。因此，协议的 XML 打包和解析是同时存在于 C/S 两端的协议处理模块。

协议的 XML 打包和解析过程本身是简单的，这和从协议的 UML 图生成协议的 XML Schema 是类似的。但要注意后者是元数据级的对象类到 XML 语法的映射，而前者的映射是在数据级完成。

4.3 应用实例

CDAE 原型系统采用 Java 语言开发，其中协同设计通信协议和本文主要论述的组员关系协议都基于 XML，协议的 XML 打包和解析由协议转换器统一实现。考虑到文档完整性、转换性能和易用型，采用基于 XMLBean 的方法实现协议转换器^[6]。

图 5 是原型系统运行中生成的新建组请求 NewGroupReq 和新建组响应 NewGroupRes 的 XML 协议实例：ID 为 001 的用户提出新建组“起落架干涉问题协商会议”，并设置相应组属性；请求成功，组员关系服务返回了包含组 ID 为 005 的新建组请求成功的响应。

<pre><?xml version="1.0" encoding="UTF-8"?> <gcsMessage xsi:schemaLocation="gcsProtocolSchema.xsd"> <memberShipRequest> <priority>1</priority> <timeStamp>2005-07-12 15:22:48</timeStamp> <newGroupReq> <requesterID>001</requesterID> <group> <groupName>起落架干涉问题协商会议 </groupName> <creatorID>001</creatorID> <isBlocked>false</isBlocked> <managerID>001</managerID> <groupMembers> <userID>001</userID> ... </memberShipRequest> </gcsMessage></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <gcsMessage xsi:schemaLocation="gcsProtocolSchema.xsd"> <memberShipResponse> <priority>1</priority> <timeStamp>2005-07-12 15:22:48 </timeStamp> <newGroupRes> <replytoUserID>001</replytoUserID> <groupID>005</groupID> </newGroupRes> </memberShipResponse> </gcsMessage></pre>
--	--

图 5 新建组请求和响应的 XML 协议实例

5 总结

本文结合 CDAE 的应用背景，提出协同工作系统中以 XML 作为组员关系协议的基本思想，并详细给出了协议的分析、设计和实现。主要工作如下：(1)分析并归纳组员关系协议的通信行为；(2)给出了协同工作系统中组员关系请求协议和通知协议的基本框架和 UML 描述，并总结了采用面向对象的 UML 设计协议的优点；(3)结合 XML 以及 XML Schema 定义元数据技术，实现了协议的 UML 描述到 XML Schema 的映射；(4)给出了与协议相关的验证、XML 打包和解析的设计思路；(5)结合原型系统，具体介绍协议处理流程中 XML 打包和 XML 解析的实现，并给出了 XML 协议实例。

基于 XML 的组员关系协议的结构特点使其具有良好的灵活性、可扩展性和跨平台性，可方便地应用于其它协同应用的通信系统中；另外，可以借鉴本文的协议设计方法，应用于其它协议的设计中。

参考文献

- 1 史美林, 向 勇, 杨光信等. 计算机支持的协同工作[M]. 北京: 电子工业出版社, 2000-12.
- 2 彭 维. 网络化协同 CAD 系统的关键技术研究[D]. 西安: 西北工业大学, 2001-03.
- 3 Keidar I, Sussman I, Marzullo K, Moshe: A Group Membership Service for WANs[J]. ACM Transactions on Computer Systems, 2002, 20(3): 191-238.
- 4 Malik A. Design XML Schemas Using UML[Z]. <http://www-128.ibm.com/developerworks/library/x-umlschem/index.html>, 2003-02.
- 5 W3C Proposed Recommendation, XML Schema Part 2: Datatypes[Z]. <http://www.w3.org/TR/2001/PR-xml schema-2-20010330>, 2001.
- 6 WebLogic Workshop Help for XMLBean[Z]. <http://dev2dev.bea.com.cn/download/school/workshop/WorkshopCNHelp/doc/zh/core/index.html>.