

基于灵活对象的指控系统应用技术研究

袁杭萍, 覃 垚, 朱 立

(解放军理工大学 指挥自动化学院 指挥自动化系, 江苏 南京 210007)

摘 要:随着信息技术的不断发展,面向服务协作的应用对加快信息化条件下联合作战军事行动反应速度的作用日益明显,但是面对越来越丰富的 Web 服务和数据的多样性,现有技术在可扩展性、实时性以及交互性方面已经无法满足日益增长的需求。为了解决面向服务协作的上述不足,文中引出了面向组件的灵活对象的概念,提出了基于灵活对象的 Web 开发模型,设计了一个基于灵活对象的 SOC 实例。实例分析表明,该模型能够有效简化指控数据集成过程,增强基于 SOC 的指控系统的交互性和实时性。

关键词:面向服务协作;灵活对象;面向组件;指控系统

中图分类号: TP39

文献标识码: A

文章编号: 1673-629X(2012)07-0189-04

Research on Command and Control System Application Technology Based on Agile Object

QIU Hang-ping, QIN Yao, ZHU Li

(Institute of Command Automation, PLA Science and Technology University, Nanjing 210007, China)

Abstract: With the development of the information technology, service oriented collaboration (SOC) applications are often promoted as crucial to facilitating the response of joint operations. But in the face of increasing Web services and the diversity of data sources, existing technologies cannot meet the requirements for good scalability, nimble reaction and better interaction. In order to solve these problems, with introducing the concept of agile object based on component oriented, a Web development model based on agile object is put forward, a SOC case is designed based on agile object. The analysis of the case shows that the application based on agile object enables facilitating the process of data integration and enhancing the interaction and reaction of C2 system.

Key words: SOC; agile object; component oriented; command and control system

0 引 言

近几年,面向服务的体系结构(Service Oriented Architecture, SOA)在指控系统方面的应用越来越广泛,系统重用性和互操作性不断增强。随着对指控系统关于信息共享以及协同工作方面的要求的逐渐提高,面向服务协作(Service-Oriented Collaboration, SOC)的方式应运而生。SOC能够有效增强指挥信息系统的信息共享能力和协同工作能力,提高信息化条件下联合作战军事行动的反应速度,但是面对越来越丰富的 Web 服务和数据的多样性,现有 SOC 的实现技术在可扩展性、实时性以及交互性方面已经无法满足日益增长的需求^[1-4]。如何简化不同信息来源的数据集成过程和提高 SOC 系统的交互性和实时性是未

来基于 SOC 指控应用的研究方向。

1 SOC

1.1 SOC 的概念

面向服务协作(Service-Oriented Collaboration, SOC)^[5]指的是根据用户的需求,运用互联网技术,把来自同一领域或者不同领域内的信息数据结合起来,供用户查询和操作。它的构建过程可以描述为:使用区域内的企业信息和已有系统等软件资源,创建复合解决方案;由这些解决方案组合得到合成服务,提供给网络另一端的用户使用。服务收集者承担服务的合成任务,并以服务提供者身份出现。服务收集者制定所收集服务的访问规则,对外宣称这些服务由他们创建^[6]。

一个 SOC 应用是由多个不同的“mashups”构成的,mashups^[7]指的是由许多不同类型的服务应用构建而成,直接用于交互的应用,在 mashup 中,不同数据源的数据以分层的方式整合到同一个多层交互式的图形

收稿日期:2011-11-24;修回日期:2012-02-27

基金项目:国家“863”高技术发展计划项目(2007AA012Z126)

作者简介:袁杭萍(1965-),女,博士,教授,研究方向为系统工程和作战信息管理;覃 垚(1986-),男,硕士研究生,研究方向为系统集成与优化。

用户界面 (Graphical User Interface, GUI) 中。同一个协作用户组中,各个用户把不同数据来源的数据放入同一个 mashup 中,之后通过这个单元共享这些数据。

1.2 SOC 应用需求

SOC 应用的重点偏向于交互场景上,服务平台通常会针对某一个特定类型的内容发布一个专用组件,并通过嵌入式脚本开发相应的交互页面,这些嵌入式脚本直接与数据中心的后端服务相关联,这些服务是无法直接通过编程进行访问。典型的 SOC 实现和部署主要包括单个专用组件对应单个交互页面和多个专用组件对应单个交互页面两种方法。架构一个 SOC 的应用平台,需要满足以下几点要求^[8]:

1) 可以通过构建或者调整已经存在的组件来快速开发一个新的协作应用;

2) 可以把来自不同位置、不同格式和通过不同协议和接口获得的数据整合在一起;

3) 能够实时对应用程序进行配置与更改,以此来获取新的类型的数据,并且不会影响程序的操作^[9];

4) 允许用户直接发布数据,并且进行点对点的式的数据交换;

5) 在网络环境改变的情况下,系统仍然可以正常运行。

现在的 Web 服务在访问方式上大多采用企业服务总线 (Enterprise Service Bus, ESB)^[10] 的方法,而且对于 Web 服务而言,除了性能以外,可扩展性和安全性也越来越受到关注^[11]。但是由于专用组件的结构比较复杂,并且相互间交互性很差,很难把不同数据源的信息融合到一起。因此,文中提出一种新的构建方法来创建 SOC 应用。

2 基于灵活对象的 SOC 策略

2.1 灵活对象

灵活对象 (Agile Object) 是一种面向组件的编程模型,每个灵活对象代表一个可重复使用的协议组件、一个或多个来自各地的 Web 服务的应用组件或者是整合了多个灵活对象的组件等。用户通过组件拖放的方式把不同的对象放在同一个界面上,采用分层融合信息共享的方法,对这些不同数据来源的数据以及传输这些数据所用到的网络传输协议进行整合,构建出来的实时应用以 XML 文件的方式进行存储。用户可以通过本地灵活对象运行环境打开该文件来运行这个应用。

灵活对象具有以下 3 个基本特性:

(1) 构建方式便捷。

与现有的网络开发模型不同的是,灵活对象是通过组件融合的形式来实现用户交互的,每个组件从别的组件的显示接口获取数据,这与上文提到的专用组件的方法有所区别。允许用户通过类似拖放组件的方式,迅速快捷地开发一个 SOC 应用。

(2) 可扩展性强。

灵活对象中的组件不像那些专用组件那样是被预先设置好的,为了实现某个功能,灵活对象的组件从某个文件系统的目录或者某个数据库中获取需要的对象,当组件的内容发生改变时,则会对所需的对象进行必要的添加或者删除,用户可以根据实际需求进行必要的更改和添加。

因此,灵活对象可以支持诸如态势信息等需要动态计算的对象,并且在显示的过程中动态添加或删除这些对象的应用。这类模型可以支持非常多的协作与合作的模式。

(3) 交互性强。

不同的灵活对象组件在同一个应用中使用,要有统一的交互方式。无论是 SOC 应用之间的交互,还是在 SOC 应用内部的组件之间的交互都具有一致的事件和表现方式。为了实现这种统一的交互方式,可以采用分层的方式对 SOC 系统的功能进行划分,可以分成四个层面:可视层、连接层、更新层和传输层。每层之间使用基于事件的接口进行交互。在网络环境发生改变的情况下,SOC 应用可以迅速切换到合适的传输层来保持通信。因此,SOC 应用之间的交互方式不再具有单一性。

2.2 基于灵活对象的 SOC 模型

灵活对象的实现实际上是一个分布式应用的实现:不同网络节点上的组件通过交换信息的方式相互协作,从而实现某个设计好的应用。组件的类型包括

灵活对象的分布式协议

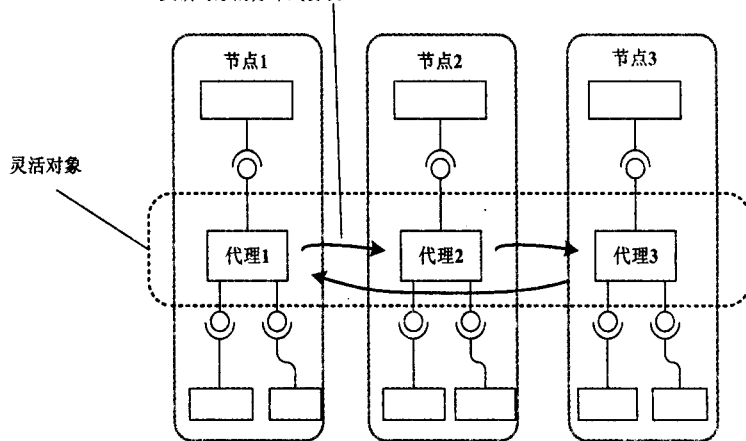


图 1 灵活对象模型结构

Web 服务、数据库和别的灵活对象等。当网络中的节点需要使用灵活对象时,可以通过调用该对象的代理组完成。

2.2.1 灵活对象的代理

代理 (Agent) 是灵活对象的功能的一部分,每个灵活对象的调用是通过一个或者一组代理来完成的,代理之间根据灵活对象的分布式协议进行数据同步与交互,不同的灵活对象之间是通过同一类型的代理进行交互,如图 1 所示。每个代理具有一系列的可与外部进行事件交互的端点 (endpoint),每对类型相同的端点可以构成一个通道,通过这些通道,对象代理可以与在同一个节点上的其他软件进行交互。代理间以及与别的组件之间进行的交互必须通过这些交互端点来实现,端点具有严格的类型区分,每种类型的端口只允许相同类型的事件进行传递,代理与代理之间通过消息进行交互,如图 2 所示。

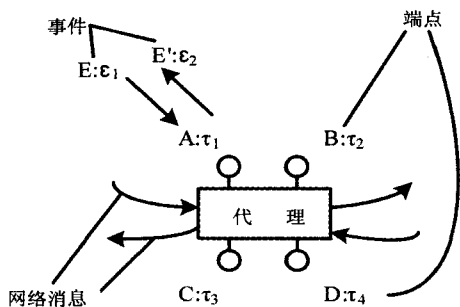


图 2 代理结构

这些端口具有以下几个特点:

(1) 每对端口之间可以进行多次连接:每个端口拥有一种类型,该类型包含可通过端口进行传输的事件格式以及排序、安全、容错性和别的一些属性信息,只有当端口的类型声明了连接时所需要的属性时,运行环境才允许连接的建立;

(2) 每个代理可以根据在交互体现出的作用来定义可以使用的端口数量,主要根据和该端口连接的另一个端口的需求和类型有关。

2.2.2 代理的实现

每个灵活对象具有一串引用代码,该引用代码是一整套构建和配置该灵活对象代理组的指令,该应用代码以 XML 的形式进行存储,保存在网络中的共享服务器中。当网络中某个节点需要调用某个灵活对象时,在已具备灵活对象平台的条件下,从共享服务器上下载或者直接从本地打开该灵活对象对应的引用代码,系统就会根据引用代码中的指令创建该对象的代理组。

在实际应用中,用户在具体构建协作应用之前,从共享服务器上下载一个引用代码共享文件夹,在该文

件夹中存在大量可供用户使用的灵活对象引用代码,用户可以根据实际需求情况,任意组合相应的灵活对象应用代码,从而成功构建出基于灵活对象的 SOC 协作应用。这不仅增强了程序的适应性,在提高面对突发需求时程序的可调整性方面也起到了巨大的作用。

3 应用分析

现有一个抢险救灾的场景:由于连日大雨,某地区发生了洪灾,并伴有大规模的山体滑坡和泥石流现象,许多群众被困,上级机关高度重视,向受灾地区的部队迅速下达抢险救灾任务,接受任务的部队分别派出若干个救援组,其中包括后勤保障组、医疗保障组和抢险分队^[12],前往该地区展开救援。每组都配备有必要的救援设备、移动通讯设备和笔记本等设备,用来获取以及发送受灾和救援情况,抢险救灾指挥中心设置有大屏幕,用来显示救灾情况,如图 3 所示。

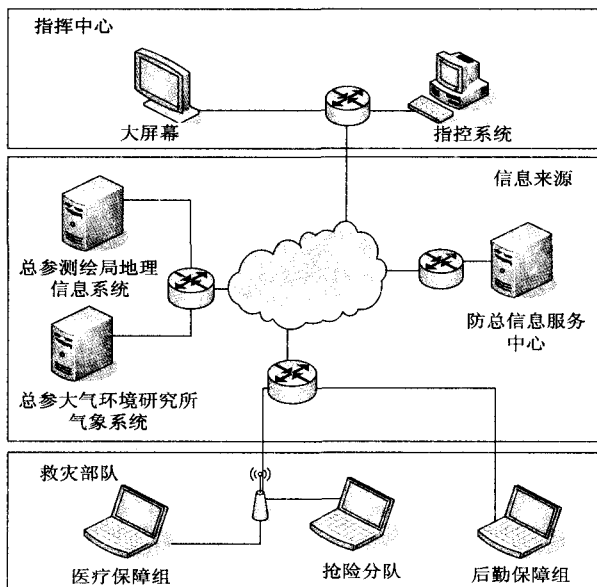


图 3 救援部署情况

负责本次抢险救援任务的总指挥员把各个救援组派遣至各个受灾区域,每个小组根据由指挥中心指定的营救方案展开抢救灾员的行动。到达受灾区域后,救灾人员迅速把当地具体的受灾情况、公路情况、人员受困情况和己方所处位置等信息反馈给指挥中心,指挥中心根据反馈回来的信息进行必要的救灾措施的调整。

如图 3 所示,构建这么一个救援系统所涉及到的信息来源并不单一,地理地图信息来源于总参测绘局地理信息系统 (GIS),人口信息来自于防总信息服务中心,天气信息来自总参大气环境研究所,部分信息来源还可能牵涉到军队内部的指挥系统。这些特定来源的数据要求对应接口能够解析不同的协议和特定的格式。如果采取传统的方法来构建这个协作工具,需要

动态开发和协作,满足智能家居系统的分布、异构和个性化的需求。在 Multi-agent 系统中嵌入本体,解决了不同 Agent 间的交互和协作问题。同时为了保持各个 Agent 的独立性,使 Agent 有效的同其他 Agent 发生直接或间接的数据交换和共享,采用本体来负责沟通 Agent 与环境之间的信息交流。这样的家居系统具有通信可靠、协作有效等特点,对于今后开展后续工作,有着十分重要的意义。

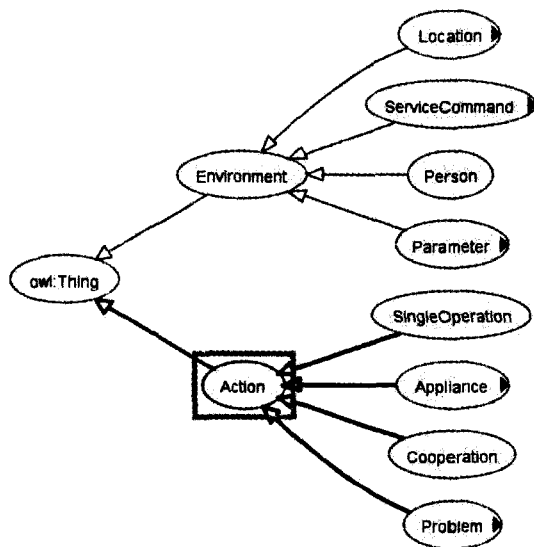


图 3 推理之前类图

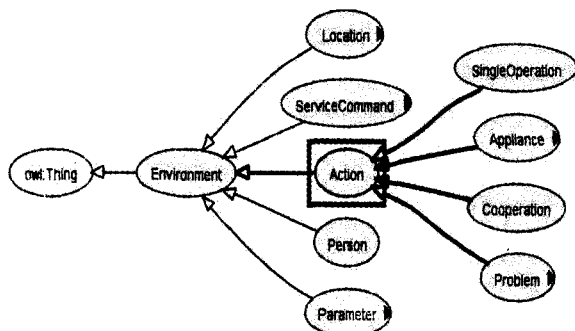


图 4 推理之后类图

参考文献:

- [1] Adams M D, Kossentini F. Jasper: A Software-based JPEG-2000 Code Implementation[C]//Proc. of Int'l Conf. on Image Processing. Vancouver; [s. n.], 2000.
- [2] Boliek M, Christopolos C, Maiani E. ISO/IEC JTC1/SC29/WG1 N1646R-2000, JPEG2000 PartI Final Committee Draft (Version 1.0)[S]. 2000.
- [3] Hanssens N, Kulkarni A, Tushinda R, et al. Building agent based intelligent workspaces[C]//ABA Conference Proceedings. USA; [s. n.], 2002.
- [4] Wang Lu, Wang Zhiliang, Gao Xiaoliang, et al. Research on an MAS-based Smart Home Network[C]//International Conference on Intelligent Control and Information Processing. Dalian; [s. n.], 2010.
- [5] Wang Lin, Zhao Hongshuai. Ontology for Communication in Distributed Multi-agent System[C]//2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science. Hong Kong; [s. n.], 2010.
- [6] 赵文龙, 侯义斌. 多 Agent 系统的组织结构与协同[J]. 计算机工程与应用, 2000(10): 59-61.
- [7] 张林, 徐勇, 刘福成. 多 Agent 系统的技术研究[J]. 计算机技术与发展, 2008, 18(8): 81-83.
- [8] 李晓华, 赵现纲. 一种基于多 Agent 的智能家居系统研究与设计[J]. 计算机工程, 2006, 32(21): 233-257.
- [9] 叶朝辉, 杨十元. 智能家庭网络研究综述[J]. 计算机应用研究, 2001, 18(9): 1-6.
- [10] 王世进, 周炳海, 陶丽华, 等. Agent 通信语言综述[J]. 计算机工程与应用, 2005(21): 32-37.
- [11] Brost W N. Construction of Engineering Ontologies for Knowledge Sharing and Reuse[D]. Enschede: University of Twente, 1997.
- [12] Usehold M, Gruninger M. Principles, Methods Applications[J]. Knowledge Engineering Review, 1996, 11(2): 93-155.
- [13] 赵海, 陈燕. 普适计算-计算混沌形式[M]. 沈阳: 东北大学出版社, 2005: 35-37.

(上接第 192 页)

- 型[J]. 计算机学报, 2005, 28(4): 514-523.
- [5] Smith D, Kay A, Rab A, et al. Croquet: A Collaboration System Architecture[C]//Proceedings of First Conference on Creating, Connecting and Collaborating Through Computing. [s. l.]: [s. n.], 2003.
- [6] 危达. Web 服务协作框架及相关技术的研究[D]. 长沙: 中南大学, 2009.
- [7] IBM developer works 中国. Mashups: WEB 应用程序新成员[EB/OL]. 2007. <http://www.ibm.com/developerworks/cn>.
- [8] Birman K, Cantwell J, Freedman D, et al. Building Collabora-

tion Applications that Mix Web Services Hosted Content with P2P Protocols[C]//ICWS. [s. l.]: [s. n.], 2009.

- [9] 孙巍, 徐学东, 徐学军. Java 反射机制在可重构 Web 框架中的应用[J]. 计算机工程与应用, 2005(36): 212-217.
- [10] 张昀. 基于 SOA 架构的分布式 ESB 总线研究与实现[D]. 南京: 南京理工大学, 2009.
- [11] 杨永森. Web 服务集成若干关键技术的研究[D]. 扬州: 扬州大学, 2006.
- [12] 吴立玖, 黄志. 浅析非战争军事行动后勤物资供应保障[J]. 军事经济研究, 2011(1): 124-135.