

一种基于 ECA 规则的 Web Service workflow 模型的研究

贺春林¹ 滕云¹ 彭仁明²

(西华师范大学计算机学院 南充 637002)¹ (绵阳师范学院物理与电子信息工程系 绵阳 621000)²

摘要 workflow 技术的深入应用要求 workflow 管理系统增强 Web 服务功能,基于 Petri 网 workflow 过程模型存在无法表述状态变迁过程或状态变迁边缘时刻事件的缺陷。给出了基于 ECA 规则 Web Service workflow 模型形式化定义,基于 ECA 规则的过程模型以事件推动 workflow 实例的执行,通过严格定义事件的语义来保证 workflow 的正确执行和监控并支持 workflow 在运行中修改实例。

关键词 workflow, ECA 规则, Web service

中图分类号 TP311 **文献标识码** A

Web Service Oriented Workflow Model Based on ECA Rules

HE Chun-lin¹ TENG Yun¹ PENG Ren-ming²

(College of Computer, China West Normal University, Nanchong 637002, China)¹

(Dept. of Physics and Electronic Information, Mianyang Normal University, Mianyang 621000, China)²

Abstract While workflow technology is increasingly used to manage complex processes in scientific and business field, the enhancements to Web service capabilities are required. The Petri net based workflow model has limitations that could not depict the course of state transition or the event happened at the moment of state transition. An ECA-rule based Web Service workflow model was proposed. The process instances of the model is driven by events, Which not only ensure workflow to run and monitor correctly but also sustain dynamic modification at runtime.

Keywords Workflow, ECA rules, Web service

1 引言

Web Service 技术和 workflow 技术是两个相对独立发展的 IT 技术。随着 IT 技术的深入应用,企业 IT 开始向更广的区域延伸,企业各部门之间,各企业业务系统之间的交互变得越来越频繁。传统的工作流管理系统往往基于特定的操作系统平台和网络协议,各 workflow 厂商往往使用不同的 API,数据结构也不一致,这就使实现跨企业的、异构的、分布式的工作流管理系统变得十分困难。同时过去一些有限集成采用紧耦合方式,使得系统集成的柔性欠佳。

Web Service 的出现一定程度上解决了这个问题。首先,Web Service 使用已成为工业标准的 Http 协议和 Xml 为通信手段,减少了异构环境之间对私有适配器和连接器的需要。其次,系统各部分之间既是一种松散的耦合关系而又具有高度的集成能力;另外,Web Service 协议栈中的安全协议保障了交互的安全性和交互信息的完整性。

Web Service 和 workflow 技术的结合使建立分布式 workflow 管理系统的实现相对简单,同时也对 workflow 过程模型提出了更高的要求。基于 Petri 的工作流模型因其兼顾语义和图形语言的优势在 workflow 技术中运用较多,但存在无法表述状态变迁过程或状态边缘时刻事件的缺陷,在分布式等复杂环境下表达能力不足。因此本文致力基于 ECA 规则 Web Service

workflow 过程模型研究,基于 ECA 规则过程模型以事件驱动 workflow 实例的推进,具有较强的表达能力及动态变化能力。

2 workflow 和 Web Service

2.1 workflow 技术

workflow 技术的目标是用计算机辅助业务过程,通过流程的建模执行监控构建 BPM,实现业务过程在计算机应用环境下的自动或半自动化。进入 21 世纪以来,workflow 技术得到很大发展,与企业应用集成结合日益紧密。因此增强 workflow 管理系统的 Web 服务功能,使其适用于 Web 服务结构体系是未来 workflow 技术的发展趋势。

2.2 Web Service

Web Service 是崭新的分布式计算模型,具有松耦合、跨平台和语言实现的特性。从表面上看,Web Service 是一个应用程序,向外界暴露能够通过 Web 调用 API 提供的服务。从深层次上看 Web Service 是一种新的 Web 应用程序分支,是自包含、自描述、模块化应用,可以在 Web 中描述、发布、查找以及通过 Web 调用。

3 基于 Web Service workflow 的产品缺陷及发展趋势

workflow 和 Web Service 融合有两种场景,workflow 管理系统可以根据需要控制一组构成应用程序的 Web 服务流,也可

到稿日期:2009-01-18 返修日期:2009-03-10 本文受四川省重点科技基金项目(02GG006-036),校级科研项目(412167)资助。

贺春林(1971-),男,副教授,主要从事计算机网络安全方面研究工作和专业教学工作,E-mail:chunlin_he@163.com;滕云(1971-),男,讲师,主要从事计算机教学和数据挖掘方面的研究工作;彭仁明(1969-),男,副教授,主要从事电子类的教学及研究工作。

以作为 Web 服务来提供。在一个由 Web 服务构建而成的合作应用程序里,应用程序中的业务流程是一组 Web 服务参与的任务, workflow 控制极为重要。随着应用的深入,基于 Web Service 工作流的产品体现出如下缺陷和发展趋势。

3.1 工作流过程模型描述能力不足

工作流过程模型是 workflow 管理系统的基础,包括 workflow 定义、活动、转换条件。基于 Petri 网的过程模型运用较多,根据 Petri 网原理可以表达 21 种 workflow 模式(5 个基本模式、5 个高级分支与同步模式、2 个结构模式、4 个多实例模式、3 个基于状态的模式、2 个取消模式),基本上能够描述现实生活中遇到的大部分业务流程。

Petri 网过程模型是基于状态的建模方法,它明确定义了模型元素的状态并通过状态驱动实例执行,但无法表述在状态变迁过程或状态变迁边缘时刻发生的事件。

例如一个财务报销审批流程,报销人填写报销单后由部门经理审批,对于金额小于 1000 元的直接到财务室报销;对于大于 1000 元的须总经理审批同意后方可报销。在部门经理和总经理审批不同意时,报销单返回报销人以便重新修改。如果提出下面的 2 个需求,该流程的表达则会出现困难:

(1) 希望在每个环节产生工作项时,都给参与者发送一个邮件进行提醒,提示参与者有新的待办事宜。

(2) 在财务会计进行报销这个环节,一旦产生工作项则同时给报销人发送邮件提醒,以便报销人及时获知报销信息并来财务室领取报销现金。

对于第一个需求,一种解决方案是在每个活动前再跟一个自动活动,由自动活动的应用程序向参与者发送邮件提醒。在流程比较复杂活动环节较多的情况下此种方式无法实施。

对于第二个需求,由于要求产生工作项和通知报销人同时进行(并行),这在基于 Petri 网的过程模型下无法实现。

3.2 要适应现在的应用系统轻量化、松耦合的发展趋势

轻量化是指应用系统之间、应用系统的各个模块之间要简单化、松耦合。要求各模块相对独立,模块间依赖性低,通过标准的规范进行连接。

工作流和 Web Service 相结合,workflow 管理系统可以根据需要控制一组成应用程序的 Web 服务流,也可以作为 Web 服务来提供,workflow 控制极为重要,完全不同的 workflow 之间的交互也不可避免。

这就需要尽量减少应用系统与 workflow 引擎之间的代码连接,workflow 开发的应用系统各部分不是“build into”而是“plug-in into”,同时要通过标准的规范连接。

综上所述,进一步增强 workflow 过程模型的表达能力,降低 workflow 引擎与应用系统之间的耦合度,是现在 workflow 技术发展趋势之一。鉴于此,提出了基于 ECA 规则的面向 Web Service 工作流模型。

4 基于 ECA 规则的面向 Web Service 工作流模型研究

4.1 基于 ECA 规则的面向 Web Service 工作流框架

基于 ECA 面向 Web Service 工作流框架如图 1 所示。用户通过 workflow Design Tool 提供的图形界面业务过程建模;Workflow DataBase 存储了 3 类数据,建模数据(如建模需要的元素),流程数据(如流程运行相关信息)和系统数据(如 Web Service 地址等)。

workflow 引擎根据 workflow design Tool 定义的工作流模型建立实例,通过 ECA 规则驱动实例执行。任务的完成需要调用 Web Service 时,workflow Design Tool 通过 Adapter 从 UDDI 得到调用服务信息,将调用服务作为流程定义中的一个活动,实例执行后 workflow 引擎根据流程定义调用相应的 Web Service。Workflow monitor 用于监控 workflow 引擎实时信息。

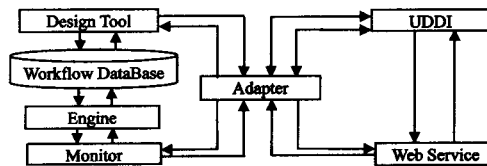


图 1 基于 ECA 规则的 Web Service 工作流框架

4.2 ECA 规则与流程模式

ECA 规则是一个(Event, Condition, Action)三元组,即在某一事件(Event)下满足条件(Condition)将执行的相应动作(Action)。基于 ECA 规则的工作流建模方法以事件驱动 workflow 实例的推进,可以通过严格定义事件的语义来保证 workflow 的正确执行和监控并支持 workflow 在运行中修改实例。

下面是用 ECA 规则表达顺序、选择、并行、单一归并、多路归并 5 个基本流程模式。

1) 顺序路由

假设流程存在 3 个串行活动:活动 A、活动 B、活动 C。

- 规则 1:〈A 结束, if = true, 启动 B〉
- 〈A 结束, if = false, 空操作〉
- 规则 2:〈B 结束, if = true, 启动 C〉
- 〈B 结束, if = false, 空操作〉

2) 选择路由

假设存在 4 个活动:活动 A、活动 B、活动 C、活动 D。依赖关系为:活动 A 结束, Condition1 满足启动 B, Condition2 满足启动活动 C, Condition1 Condition2 都不满足启动活动 D。

- 规则:〈A 结束, Condition1, 启动 B〉
- 〈A 结束, Condition2, 启动 C〉
- 〈A 结束, Condition3, 启动 D〉

3) 并行路由

假设存在 4 个活动:活动 A、活动 B、活动 C、活动 D。依赖关系为:活动 A 结束以后,同时启动活动 B、活动 C、活动 D。

- 规则:〈A 结束, if = true, 启动 BCD〉

4) 单一归并路由

假设存在 4 个活动:活动 A、活动 B、活动 C、活动 D。依赖关系为:活动 A、活动 B、活动 C 任意一个结束以后,活动 D 就启动。

- 规则:〈A 结束, if = true, 启动 D〉
- 〈A 结束 B 结束 C 结束, if = true, 启动 D〉

4.3 基于 EAC 规则的 Web Service 工作流过程模型

面向 Web Service 的工作流通过 workflow 引擎建立流程实例,通过 ECA 规则驱动实例执行,面向消息的中间件将流程任务分配给人或服务代理,服务代理根据需要调用相应的 Web Service,workflow 调用的 Web Service 看成是过程模型中的一个活动。基于 ECA 规则面向 Web Service 的工作流过程模型可以用一个八元组来表示。

定义 1 Workflow = (E, A, C, R, LC, TC, F, D)。其中 A 是活动集, E, C 分别是事件集和条件集, R 是 ECA 规则集, 用于建立活动之间的关系。D 是数据类型集, 包含了工作流所涉及的各种数据类型。

定义 2 活动集 A 是一个十元组 Activity = (IdName, Tasks, Role, Rdata, State, f, θ , ν , σ)。Id 为活动的标识, Name 为活动名字; Tasks 是任务集合; Rdata 是与此活动相关的工作流相关数据, 包括输入数据和输出数据; State 是活动所处的状态; f 是一功能映射, 它把执行活动中的任务与一组工具相映射; θ 是角色和执行者的匹配函数, 负责把此活动的角色与执行者(人/Agent)进行匹配; ν 是赋值函数, 在活动执行中负责给 Rdata 中的变量赋值; σ 是状态转换函数, 用于改变活动的状态。

定义 3 活动有 6 种状态, 等待、准备、执行、完成、超时、取消。等待表示活动触发条件未满足, 活动执行条件还未就绪, 是初始状态; 准备表示活动的触发条件已经满足, 可以开始执行活动中的任务; 运行表示该活动中的某些任务已经开始执行, 但没有全部完成; 取消表示活动没有完成就不再被执行; 超时表示发生异常时退出执行状态; 完成表示活动任务集中的所有任务都已执行完成。

定义 4 事件集 E 包括原子事件和复合事件, 复合事件通过组合原子事件或复合事件而成。两种最基本的组合为:

And: e1 and e2 表示事件 e1 和 e2 同时发生。

Or: e1 or e2 表示事件 e1 和 e2 至少有一个发生。

定义 5 数据类型集 D 包括 4 种数据类型。分别是内部变量类型, xml 对象类型, 其他文档对象类型, 对象变量类型。内部变量指基本的数据类型, 可以设置成监控条件或决策点; xml 对象类型是基于 XML schema 的数据, 用来描述 Service 的输入输出消息; 其他文档对象类型是对除了 xml 格式的其他文档格式的抽象描述, 如 word pdf 等; 对象变量是从 xml 对象和其他文档对象中提取的数据项。

定义 6 一个具体的 Web 服务调用由 5 类活动组成, 即 As, Ae, Aws, Atx, Asv。As 是开始活动, Ae 是结束活动, Aw 是调用某个 Web Service 活动, At 是将数据转换为 xml 格式活动, Av 是变量赋值活动。工作流激活后, 客户端发出请求, 活动 Atx 将数据转换 xml 格式, 调用相应的 Web Service, 调用成功将 Web Service 输出的值转换为 xml 格式返回给客户端, 流程结束, 调用不成功或超时直接结束。用 ECA 规则描述这个流程为:

1. 动作集 Action(包括以下动作):

End(): 进入结束状态

TransformBelief(): 转换为输入数据格式动作。

TransformData(): 转换为输出数据格式动作。

Invoke(): 调用某个 Service 动作。

DataFlow(): 将数据输入到某个活动。

事件集 Event(包括以下事件):

Transformed(): 标志数据已经被转换事件。

StartOf(): 标志活动已从等待状态进入准备执行状态事件。

TimeOut(): 标志活动进入超时状态事件。

Ws: web service

2. 活动集 Activity(包括以下活动):

As: 开始活动。

At: 将数据转换为 xml 格式活动。

Aw: 调用某个 web service 活动。

As: 变量赋值活动。

Ae: 结束活动。

3. 数据集和数据转换(包括以下数据集及数据转换):

D1: service 的输入数据集。

D2: service 的输出数据集。

C1: 将客户端输入数据转换为 D1 的规则。

C2: 将 D2 转换成输出数据返回客户端。

4. ECA 规则集:(包括以下规则)

<End(As), if=true, Initialize(At)>

活动 As 结束, 初始化活动 At;

<StartOf(At), Null, TransformBelief(D1, C1)>

活动 At 开始, 将输入数据转换成 xml 格式;

<Transformed(D1), Null, EndOf(Atx)>

输入数据已被转换成 xml 格式, 活动 Atx 结束;

<EndOf(Atx), Null Initialize(Aw)>

活动 Atx 结束, 初始化活动 Aw;

<EndOf(At), Null, DataFlow(D1, Avs)>

活动 At 结束, 数据向活动 Aw 转移;

<StartOf(Aw), Null, Invoke(WS)>

活动 Aw 开始, 调用相应的 web service;

<Invoked(WS), Null, End(Aw)>

调用 web service, 活动 Aw 结束。

<EndOf(Aws), If "IsSuccess"=True, Initialize(Av)>

活动 Aw 结束, 调用成功则初始化活动 Av。

<EndOf(Aw), Null, DataFlow(D2, Av)>

活动 Aw 结束, 数据流向活动 Av

<StartOf(Av), Null, TransformData(D2, C2)>

活动 Av 开始, 将 web service 输出数据转换为 xml 格式。

<Transformed(D2), Null, Initialize(Ae)>

转换成输出数据, 初始化活动 Ae;

<TimeOut(Ws), Null, Initialize(Ae)>

调用 Ws 超时, 则初始化 Ae

<EndOf(Aw), If "IsSuccess"=False, Initialize(Ae)>

活动 Aw 结束, 调用不成功则初始化活动 Ae。

<StartOf(a5), Null, End(a5)>

Ae 开始, 流程结束。

根据此模型, 工作流运行中过程状态的变化(如过程实例启动后)、活动状态的变化(如工作项完成后)、接收到特定消息、系统异常信息的出现、用户自定义的事件均属于事件集。一个事件包括事件类型、触发条件、执行的动作列表和参数列表等要素。事件产生后根据相应的触发条件执行相应的动作或动作集。

与基于 Petri 网的模型在描述复杂流程时流程图较烦琐以及无法表述状态变迁过程或状态变迁边缘时刻事件的缺陷相比较, 用该模型所描述的流程图较为简洁(其建模工具外观上可采用图形化的表示方法, 流程定义由若干活动组成, 具体的流程逻辑用 ECA 规则表示而不必全部体现在流程图形上)同时具有更高的表达能力。下面利用该模型解决上述财务报销审批流程的两个需求: 针对第一个需求可在“填写报销单”

这个活动的“产生工作项之后”这个事件添加发送邮件动作；至于第二个需求，要求发送消息与产生工作项同时进行，只需在“产生工作项之前”事件中添加发送消息动作。

结束语 基于 Petri 网工作流过程模型所存在的缺陷和工作流产品需增强 Web 服务功能的发展趋势，提出了基于 ECA 规则的 Web Service 工作流过程模型，并给出了形式化定义。基于 ECA 规则的过程模型以事件驱动工作流实例的推进，通过严格定义事件的语义来保证工作流的正确执行和监控并支持工作流在运行中修改实例。

参考文献

[1] Chen Lin, Li Minglu, Cao Ian. ECA Rule-Based workflow Modeling and Implementation for Service Composition[J]. IEICE-Transactions on Information and Systems, 2006, E89-D(2): 624-

[2] Cery S, Daniel F, Facca F M. FACCA. Modeling web applications reacting to user behaviors[J]. Computer Networks; The International Journal of Computer and Telecommunications Networking, 2006, 50(12): 1533-1546

[3] Liu Dongsheng, Wang Jianming, Chan SCF. Modeling workflow processes with colored Petri nets[J]. Computers in Industry, 2002, 49(3): 267-281

[4] Choi I, Jeong H, Song M, et al. IPM-EPDL: An XML-based executable process definition language[J]. Computers in Industry, 2005, 56(1): 85-104

[5] 范玉顺, 吴濠. 一种提高系统柔性的工作流建模方法研究[J]. 软件学报, 2002, 13(4): 833-839

[6] 赵文, 胡文惠, 张世琨, 等. 工作流元模型的研究与应用[J]. 软件学报, 2003, 14(6): 1013-1016

(上接第 93 页)

幅度特性和相位特性($a_0 = a_1 = 0.5$, 3 阶), 如图 4、图 5 所示。联立式(7)、(8)和(15), 可得由全通滤波器组成的综合滤波器组中低通滤波器 $F_0(z)$ 和高通滤波器 $F_1(z)$ 的幅度特性和相位特性($a_0 = a_1 = 0.5$, $N_i = 2, 7$ 阶), 如图 6 和图 7 所示。比较图 4—图 7 可得, 随着滤波器阶次的升高, 滤波器的幅频特性越来越平坦, 纹波越来越小。当阶次比较高时, 纹波可忽略不计, 同时延时越来越大, 这也是需要进一步解决的问题。由图 3—图 7 进一步说明了由全通滤波器设计的 QMF IIR 的正交性和良好重建性。

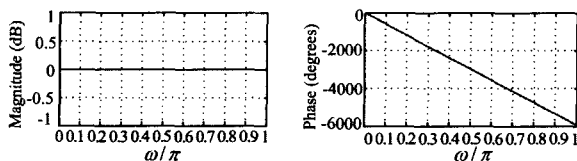


图 3 整个系统的幅度特性与相位特性

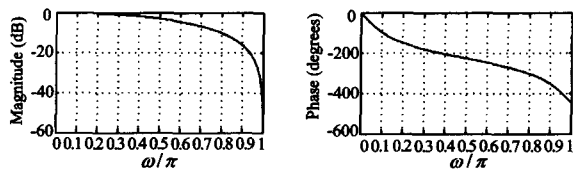


图 4 分析滤波器组中低通滤波器的幅频特性和相频特性

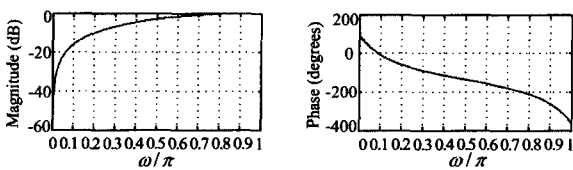


图 5 分析滤波器组中高通滤波器的幅频特性和相频特性

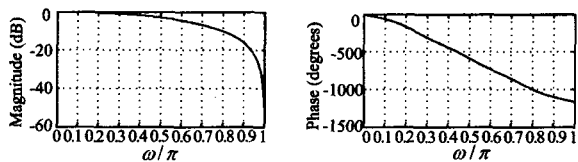


图 6 综合滤波器组中低通滤波器的幅频特性和相频特性

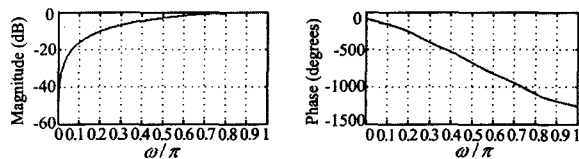


图 7 综合滤波器组中高通滤波器的幅频特性和相频特性

结束语 本文提出一种设计两通道 QMF IIR 滤波器组的新方法。本方法采用格型结构, 分析/综合滤波器组均采用全通滤波器。与传统的 FIR 设计方法相比, 新算法的计算量要小得多, 没有混叠失真, 没有幅度失真或者幅度失真可以忽略不计(重建信号的误差和延时相互制约), 群延迟比以往的方法要高, 但对许多信号处理系统来讲, 是可以接受的。

参考文献

[1] 王玲, 崔小弟. 矩阵共轭正交滤波器组的生成[J]. 计算机科学, 2000, 27(8): 80-82

[2] Nguyen T Q, Vaidyanathan P P. Two-channel Perfect-Reconstruction FIR QMF Structures Which Yield Linear-phase Analysis and Synthesis Filters [J]. IEEE Trans. on Acoustics, Speech, and Signal Processing, 1989, 37(5): 676-690

[3] Vaidyanathan P P, Hoang P Q. Lattice Structures for Optimal Design and Robust Implementation of Two-channel Perfect-reconstruction QMF Banks [J]. IEEE Trans. on Acoustics, Speech, and Signal Processing, 1988, 36(1): 81-94

[4] Lowenborg P, Johansson H, Wanhammar L. A Class of Two-channel IIR/FIR Filter Banks [C]// Proc. of European Signal Processing Conf. (EUSIPCO). Tampere, Finland, Sept. 2000: 1897-1900

[5] Zhu W-P, Ahmad M O, Swamy M N S. An Efficient Approach for the Design of Nearly Perfect-reconstruction QMF Banks [J]. IEEE Trans. on Circuits and Systems II, 1998, 45(8): 1161-1165

[6] Klouche-Djedid A, Lawson S S. A General Design of Mixed IIR-FIR Two-channel QMF Bank [C]// Proc. of Intl. Symp. on Circuits and Systems (ISCAS). Geneva, Switzerland, May 2000: 559-562

[7] Galija E. Allpass-based Near-perfect-reconstruction Filter Banks [D]. Kiel, Germany; Christian Albrechts University, 2002

[8] Lang M, Laakso T I. Simple and Robust Method for the Design of Allpass Filters Using Least-squares Phase Error Criterion [J]. IEEE Trans. on Circuits and Systems II, 1994, 41(1): 40-48