维普资讯 http://www.cqvip.com

一种基于 AOSD 的工作流管理系统的实现*)

董云卫[1] 郝克刚[2]

(西北工业大学计算机学院 西安 710072)[1] (西北大学计算机科学系 西安 710069)[2]

摘 要 采用面向方面软件开发方法建立事务工作流管理系统的软件体系结构。通过提取工作流应用的业务流程业务活动、参业者和事务这四个关注点,利用面向方面的软件开发方法实现并独立封装事务工作流的需求关注,并对这些关注的实现进编织生成事务工作流应用程序,解决不同工作流程之间、不同事务之间信息交换和协同工作,在确保系统执行状态正确的基础上,降低了工作流应用关注间的耦合性,增强了事务管理柔性管理能力。

关键词 面向方面软件开发,事务工作流,工作流管理系统

An Implementation of Transactional WfMS Based on AOSD

DONG Yun-Wei[1] HAO Ke-Gang[2]

(School of Computer Science, Nortwest Polytechnic University, Xi'an 710072)[1]

(Department of Computer Science, Northwest University, Xi'an 710069)[2]

Abstract An Aspect-Oriented software architecture is presented based on Oriented Aspect Software Development. There are four concerns, which are business process, activity, participant and transaction, are extracted from workflow application. All workflow concerns are implemented independently, and then they are weaved into a workflow application system. Different business processes can cooperate and exchange data, and do transactions to do so. It can guarantee workflow application is correct in running-time, and reduce the coupling among workflow application concerns. The workflow management is flexible.

Keywords Aspect-oriented software development (AOSD), Transactional workflow, Workflow management system (WfMS)

1 前言

目前大多软件系统的开发都是基于面向对象的软件实现方法,把软件中一些功能特性或需求设计成软件对象和组件,通过对这些对象或组件的继承、装配成软件系统,这样一方面提高软件代码的重用性,同时它还能提高系统功能模块的独立性,降低功能模块之间的耦合性,增强了系统的可装配性和维护性。然而,系统的有些特性和需求是横切(Cross-cutting)于系统的每一个层面中,并融于系统的每一个构件中。这种特性称为系统的需求方面(Aspect)或关注点(Concerns),如系统中的业务逻辑(流程)、数据的持久性、系统的可靠性(如日志、事务能力、异常处理和恢复)、认证、安全性、错误检测等。从这个意义上讲,一个复杂的软件系统可以看作是由许多关注点的实现构成的。

面向对象的软件开发方法是把系统看作是一些类和对象的集合,利用对象或类,把一个复杂的系统分解成为一个个相对独立的模块。然而,横切于全系统的需求关注点是多维的,跨越多个软件模块,而面向对象方法或面向过程方法却是一维的方法,采用这样的方法开发软件系统就是把多维的需求映射到一维的核心级模块实现中,这样会导致两类问题难于对付:

· 混杂(tangling):系统的不同需求在软件模块中同时需要满足并互相影响,一个软件模块要同时考虑业务流程、事务管理、业务功能和人机交互等方面,以及它们之间的协同和安全性;

· 分散(scattering):由于系统关注的特性是横切于整个系统中,跨越多个软件模块中,它们的实现代码也会分布在不同的软件模块。

在基于面向对象的事务工作流应用系统的实现过程中,多种需求关注点及其实现的模块(构件、类)之间的对应中,软件代码混杂、分散,导致软件开发过程的可追踪性差、开发效率低、代码的重用性不好、代码量不高、软件系统的演变进化困难等,直接导致系统的开发难度大、维护性差。面向方面方法就是针对这些问题而产生的。

一般来讲,面向方面的软件开发方法包括方面分解、关注实现和方面编织三个步骤[1],如图 1 所示。

· 方面分解(Aspectual decomposition):通过对系统需求的分解来识别出哪些是横切于全系统的关注,哪些是系统的公共关注,并从横切于系统中的关注分解出模块级的关注;

· 关注实现(Concern implementation):分别实现每一个关注;

· 方面编织(Aspectual weaving):根据实现关注的每一方面的规定,把不同的功能模块单元编织或集成在一起,形成方面满足应用包含的所有关注及功能需求的最终系统。

一个成熟的工作流软件产品既是一个工作流应用软件开发和部署平台,又是工作流应用软件运行平台。它不但要提

*)本技术成果受"陕西省自然科学基金"(项目编号:2005F46)和"西安科技攻关计划"(项目编号:GG05022)项目资助。董云卫 博士,副教授,研究兴趣:工作流技术、软件测试技术。

维普资讯 http://www.cqvip.com

供快速开发工作流应用软件所需的工具，供应用软件开发商基于工作流平台进行工作流应用系统的二次开发；同时，还要一个用工作流应用系统运行必须的支撑环境。根据 WfMC 定义的参考模型，一个工作流产品从功能上可以分为构造阶段功能域、运行阶段控制功能域和运行交互阶段功能域等三部分。在工作流系统中应用需求关注有业务流程、业务活动和参与者等，在构造阶段就是要分别实现工作流的关注，并对它们进行编织。在运行阶段和交互阶段，工作流根据对业务活动、参与者和业务流程实现关注编织后的建立的关系在工作流引擎的解析下执行。

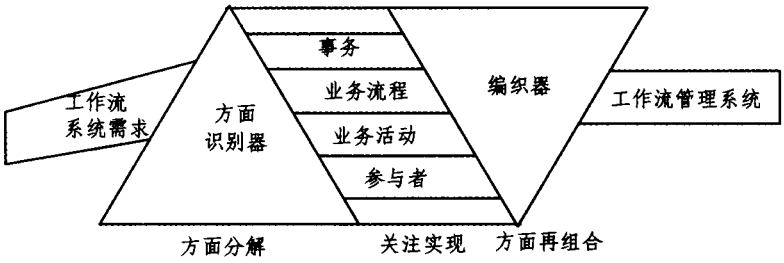

图 1　面向方面的软件开发过程

对于事务工作流管理系统来说，在构造阶段还需包括事务的内部关系和事务外部关系的结构的定义，不同事务之间协同执行的预定义等。在运行执行阶段需要提供事务执行管理机制，如事务的操作原语的调用、执行协议的解析、事务的调度等等。在运行交互阶段还需要包括控制台对事务的手工管理，如取消一个正在执行的事务、对失败事务的人工处理等。

2　基于 AOSD 的工作流管理系统体系结构

一个工作流应用系统的需求常常横切于各工作流程中，比如事务管理的需求关注横切于每一个工作流应用及其每一个工作流活动中。同样地，工作流流程关注（状态转移、相关数据变化和信牌传递等）横切到每一个工作流程中以及组成流程的每一个活动中。在工作流执行过程中每一个工作流或工作流活动都有一个执行主体即参与者，同样每一个工作流事务也有一个参与者，它既是工作流事务的执行者，又是实现事务人机交互管理的接口，如实现对工作流应用中的数据的查询、修改或中止一个工作流事务的执行等。参与者属性也是事务工作流的关注，它也横切于每一个工作流程中、工作流活动、工作流事务中。因此，我们可以把事务工作流管理系统的需求分解为：流程关注、活动关注、参与者关注和事务关注等四个基本的需求关注。

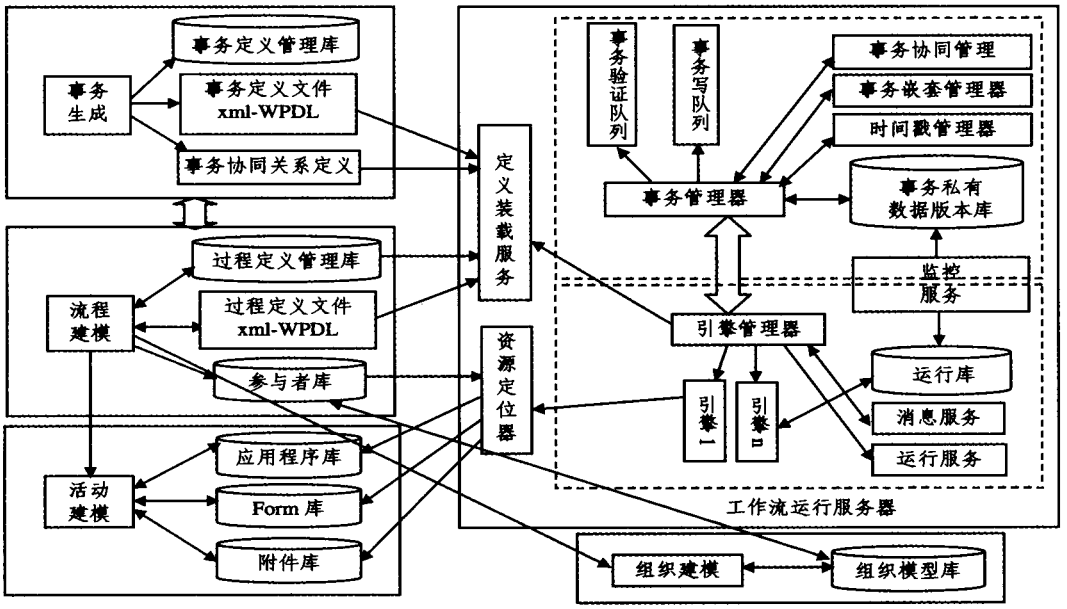

图 2　基于方面的工作流管理系统的体系架构

工作流过程功能或具体应用都是通过多个工作流活动来完成的，在具体实现中我们往往用软件构件来实现作流活动，以增加系统的复用性。在表示工作流程实现的构件中，它具有工作流程的控制属性、参业者属性和事务属性，从需求的角度看它们是多维的，如果我们仅采用面向对象的方法，把这些关注都以类或对象方式分别封装到每一个构件中，就会出现两方面的困难：一是开发难度增大，在每一个实现的活动功能构件（类或对象）中都要有各种关注的属性和方法，而这些关注的属性和方法交织在一起，增加了系统的分析和设计难度。另一方面，工作流系统的流程、参与者、事务管理以及工作流活动执行的业务操作都是为企业应用服务的，就要求它们能够随业务需求的改变而调整，如果把这些系统关注都在一个类或对象中实现，系统维护能力就大大减低了，因为某一方面关注细小的改动都需要对系统的低层结构（组成构建的类或对象）进行修改。采用面向方面的软件开发方法可以把事务工作流中的系统需求的不同方面关注分离出来，并独立表示

维普资讯 http://www.cqvip.com

为事务关注、流程关注、参与者关注和工作流活动等，并利用不同的设计方法和开发工具来对这些关注分别分析、建模。这些相对独立的关注借助相应的工具分别实现后，通过编织机制实现事务工作流的定义构造，该定义就是对不同关注实现(以程序方式)的组装或集成。编织后每一个工作流及其活动中有关流程、参与者、事务管理的结构信息都组织在一起，并建立了有机的联系，形成了反映事务工作流各需求关注的工作流定义。这种采用面向方面的软件开发实现的软件体系结构如图 2 所示。

3 工作流系统关注的实现

在基于 AOSD 开发系统时，可以利用现有的设计和编程语言分别实现构件和方面，我们可以遵循以下的原则[4]：

• 如果一段程序能够明确地封装在一起，就把它作为一个构件(如对象、方法、过程或 API 等)，构件作为系统功能的基本单元，便于访问、管理和组装。

• 如果一段程序不能够明确地封装在一起，就把它作为方面，方面不是系统功能的基本单元，而是影响系统性能的属性。

3.1 业务流程方面实现

业务流程是横切于系统各子系统或功能的关注点，可以用工作流来描述，它反映了业务活动过程中控制逻辑，包括开始和中止条件、各个工作环节及相互之间的控制流关系等，可以用 Petri 网理论等方法对其建模，将其表示成为能够完全或部分由计算机自动执行的业务过程，在此过程中，文档、信息或任务按照预定的规则传递。

在文[3]中提出了基于 Petri 网的信驱动模型来对业务过程建模，制定严格描述业务流程的语义、语法规则以及业务流程控制的相关描述信息。这种建模方式可以用开发工具实现，业务过程用 XML 语言表示。

3.2 参与者方面的实现

参与者也是横切于全系统的一个关注点，首先考虑执行一个活动是由人来执行还是由计算机完成，其次一人可能执行多个功能，一个功能也可能被多人使用，并且人在系统中的角色可能经常改变。因此需要把业务执行者这一个关注点和业务流程分离，通过建立企业的组织机构是对参与者建模，并分解为三种基本元素：组织单位、角色和参与者。

一个活动的参与者(执行者)可能有三种情况：人员、单位和角色。这三种信息存放在一个企业的组织模型数据库中，供系统编织时使用。参与者是一个人、组织单位和角色的集合运算表达式。参与者是与过程相关的，而人、组织单位和角色是独立于过程定义的，只有满足参与者表达式的人、组织单位和角色才能执行该活动。

3.3 业务活动构件的实现

业务活动是业务流程最小工作单元，表示了业务的具体功能。对应业务流程中一个逻辑步骤或环节的工作任务，一般分为手工操作和自动处理两类。通过对业务活动建模，可以分离活动的内容和活动的控制逻辑(业务流程关注)，这样活动的内容可以是一个用户界面，也可以是一个构件(如 COST 构件，编译后的执行程序或子业务流程)。当业务内容改变时，只需替换具体的功能构件，无需改变业务流程方面的实现。而活动构件的开发可以用 Java、C++ 等开发工具实现。

3.4 工作流事务的实现

工作流事务的建立和生成是建立在工作流过程建模的基础上的，它继承了事务过程建模中有关过程定义的本属性，如过程标识、过程的活动组成、活动之间的依赖关系、子过程的串行、并行执行关系以及过程的启动者、过程中每一活动的参业者、过程中的各种数据的定义等等。工作流事务的实现就是在过程建模的基础上定义每一个过程中的事务组成，即建立一个非事务工作流定义中所包含的活动与事务工作流定义中相应事务的对应关系，根据工作流的活动的状态转移关系(信牌的传递)和事务的执行规则来建立事务之间嵌套事务内部的执行依赖关系。文[4]中给出了一种乐观嵌套工作流事务模型来实现工作流事务管理。

4 工作流系统编织

工作流系统编织就是对工作流应用中各种构建进行集成，而不是创造新的构件，其任务就是在建立工作流务流程、活动、参与者与事务等关注实现之间的关联。工作流系统编织工作分别在流程建模和事务生成过程中完成的，如图 2 所示，在工作流建模时，工作流系统编织过程要为工作流业务流程设置它需要执行的具体工作流活动的名称、标识号、活动功能执行的目录解析地址、指定工作流活动的参与者。在事务生成时，工作流系统编织就是在已定义好的流程模型的基础上，还要定义事务与工作流活动的对应关系、工作流事务的组成的嵌套关系、并发事务执行依赖关系和事务之间的协同关系等，这些和事务有关的属性都需要在事务工作流执行前，根据工作流程的静态结构(工作流程定义)来对事务特性进行定义，以便于工作流系统的执行。

5 工作流系统执行

编织后的工作流程建立了各关注实现之间、关注与构件之间的关联，实现了业务流程标识、活动标识、参者和事务的操作之间的映射。工作流程在工作流引擎的支持下被加载、启动和运行。启动一个业务流程时先要对它进行实例化，启动的流程根据事先定义的活动依赖关系执行活动。工作流活动启动时也要进行实例化，根据编织时制定的参与者表达式计算出执行活动的具体参与者，每一个业务流程的实例都有一个唯一的标识，并且其中的每一个活动实例也有自己的标识。

事务工作流定义在运行时需要工作流引擎来负责解释工作流的各种操作语义，同时还需要事务管理器来解释事务管理的控制协议，并提供事务运行的管理机制，如时间戳管理、事务私有数据版本管理等。事务管理器和工作流引擎之间共同组成了事务工作流的运行平台，它提供事务工作流应用执行的支撑环境。事务工作流定义被装载入工作流运行平台中被实例化后，被解释成能够被工作流引擎和事务管理器支持的事务原语操作进行执行。

在事务工作流的执行过程中，事务的静态结构对应的是工作流活动或活动组的静态结构，而工作流事务的一次执行对应的则是活动或活动组的一个实例，也就是说，具有相同静态结构的工作流事务的每一次执行的场景(Scenario)和结果都有可能不相同。因此，在工作流应用系统在执行过程中需要动态地确定每一个事务的执行过程中的每一个属性，如执行的子事务、事务的调度方式、事务的执行者等等。

相关工作及结论 目前有许多工作流管理系统的商业软件，比较著名的国外工作流软件产品主要有 IBM 公司的 flowmark[5]、BEA 公司的 WebLogic Integration[6] 和 Ultimus 公司的 Ultimus[7] 工作流产品，这些系统都是纯粹用面向对象的方法实现的，基于这些产品开发工作流应用软件的维护性和扩展性都比较差，当应用系统局部功能需要改变时就需要对应用重新开发。

本文提出的事务工作流管理系统的实现方法，从软件开

(下转第 274 页)

维普资讯 http://www.cqvip.com

4 实验结果分析

在matlab的环境下，对输入G中V_a和V_d的个数作为观察对象，对B_i和C_i在0～5内随即取浮点数值，分别用两种算法取每十次随机试验结果的平均值，得出数据绘图如图7、8所示。

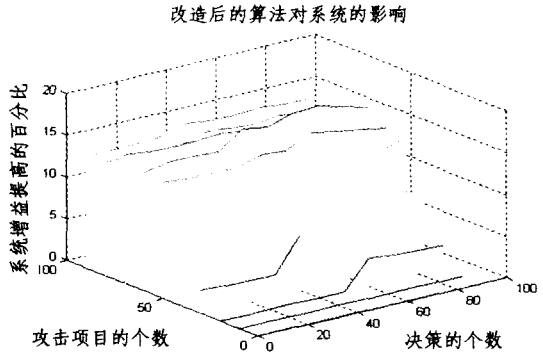

图7 决策和攻击项个数对新决策选择算法的影响

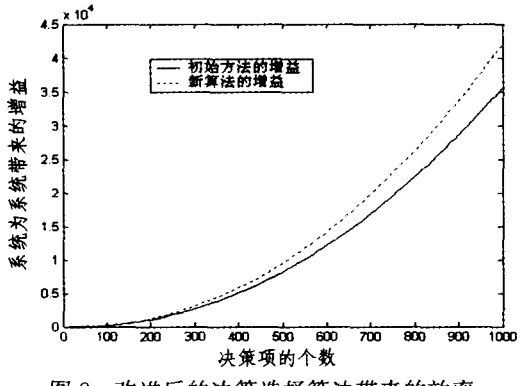

图8 改进后的决策选择算法带来的效率

图7中，取决策的个数从10到100，攻击的个数从10到100，由10次随机试验的结果平均得到，以NDC决策选择算法的比原有决策选择算法的增益提高的百分比考察本文算法对系统的增益，可以知道，

1. 随着攻击项和决策项个数的增加系统增益明显增加。

2. 决策项个数相同的条件下，攻击项目的增加对系统增益影响较大。攻击项目的个数一定时，决策个数的增加对系统效能的增加不会有很大影响。

3. 决策项目个数的增加在数量较少的情况下对系统影响增长较快。决策项个数大于攻击项个数的时候系统的增益有所提高。

4. 当系统中的攻击项和决策项的个数达到50以后，系统增益稳定在10%以上。

现实中可以使用的决策个数很多，再分析上述结果的基础上，图8选择考察本算法中决策项的个数增加对系统增益的影响，选择每5次随机试验的平均结果。纵坐标表示选择的决策集合对系统整体的增益。随着决策个数的增加，选择的决策集合能够更大地增加系统性能。可以看出，随着影响因素的增加，也就是决策集中决策的数目和攻击的数目的增加，系统的增益呈现上升趋势。并且维持比例不断增加。当系统中的影响因素达到500以上的时候，系统增益在15%左右。

所以整个改进的算法可以很大程度上提高决策集的效果，使得选择的决策集为系统带来更大的利益。

总结 针对风险分析纷繁复杂，难于掌握和控制，本文在改进已有的风险分析模型的基础上引入图重写系统，提供了攻击步骤以及决策和攻击之间关系的描述，潜在地避免了威胁的传播，使得其简单高效，便于观易于管理。提高了系统的整体性，提供了统一的视图。同样利用图重写系统，在改进二分图的最小覆盖算法的基础上，利用图着色法获得了更优决策集选择方法。随着系统决策个数增加能够增大算法带来的增益提高比例，并保证在系统中决策个数达到一定数目时有较大的增益提高。使得系统在相同的时间复杂度内大幅度的提高了可获得的利益。为风险分析提供了一个很好的方法，有效提高了系统的整体效能。

参 考 文 献

1 Rothmaier G, Pohl A, Krumm H. Analyzing Network Management Effects with SPIN and cTLA. IFIP world computer conference. Toulouse, France, 2004
2 Anderson A, Longley D, Kwok Lam-for. Security Modelling for Organisations. ACM Conference on Computer and Communications Security. Fairfax, Virginia, USA 1994
3 Kwok Lam-for. Longley Dennis. Security Modelling for Risk Analysis. IFIP world computer conference. Toulouse, France, 2004
4 Hamdi M, Bourdriga N. Algebraic Specification of Network Security Risk Management. First ACM Workshop on Formal Methods in Security Engineering, Washington D. C. 2003
5 Hamdi M, Bourdriga N. An Abstract reduction model for computer security risk. IFIP world computer conference. Toulouse, France, 2004
6 Corradini A, Gadducci F. Categorical rewriting of term-like structures. Electronic Notes in Theoretical Computer Science 51. Elsevier Science B. V. 2002. 3～6
7 Blostein D, Fahmy H, Grbavec A. Practical Use of Graph Rewriting: [technical report]. Kingston, Ontario, Canada: Queens University. 1995
8 Plump D. Term Graph Rewriting. In Handbook of Graph Grammars and Computing by Graph Transformation, World Scientific, 1999, 2(1): 3～61
9 Kruskal J B. Well-quasi-ordering, the Tree Theorem, and Vazsonyi's conjecture. Transactions of the American Mathematical Society, 95, May 1960. 210～225
10 杨继峰，孙永强. 项重写的图实现. 计算机工程，1998(4)：3～6
11 Bar-yehuda R, Even S. A Local-Ratio Theorem for approximating the weighted vertex cover problem. Annals of Discrete Mathematics, 1985. 27～46

(上接第262页)

发和维护两方面对事务工作流应用系统的关注、关注之间的实现以及关注实现编织过程进行描述。在采用基于面向方面开发的软件体系结构实现的软件系统中，当系统需求的任何一方面关注发生变化时(不论是流程定义、活动定义、事务定义，还是参与者)，只需要对该关注重新实现，然后与其他方面关注已有的实现再一次进行编织，就完成了系统的维护与更新，这种实现和编织都可以借助软件开发辅助工具来实现，不需要对系统的底层结构进行编程。该方法拟在协同工作流管理系统SynchroFLOW-T开发中采用，将极大地提高工作流软件开发的效率，增强工作流应用系统适应需求变化的敏捷度。

参 考 文 献

1 Laddad R. I Want My AOP (JavaWorld). http://www.javaworld.com/javaworld/jw-04-2002/aspect3/jw-0412-aspect3.zip, 2002
2 董云卫，郝克刚. 一种面向方面的软件体系结构. 微机发展，2004，14(6)
3 郝克刚，王斌君，安贵. WPDL中的JOIN语义问题和分区解决方案. 计算机科学，2003，30(7)
4 董云卫，郝克刚. 一种乐观嵌套工作流事务模型. 计算机科学，2005，32(8)
5 Hoffmann M, Mike D S. Ebbers, Image and Workflow Library: Advanced Work- flow Solutions using IBM FlowMark, IBM Workflow Solutions, Boeblingen, Germany, Jan. 199
6 BEA White Paper: Building A Fully Integrated, Extended Enterprise, San Jose, CA U. S. A., Http://www.bea.com, 2003
7 Khan R. Supporting BPM Teams. http://www.ultimus.com/ultwhite/BPM_Ultimus.pdf, 2004