工作流系统中基于场所的分布式授权模型研究*)

於光灿 李瑞轩 卢正鼎 宋 伟 唐 卓
(华中科技大学计算机学院 武汉 430074)

摘 要 对现有工作流授权模型进行简要分析,提出工作流系统中基于场所的分布式授权模型,新模型适应于多种现有授权模型不能支持的应用场景。将对工作流的授权分为两个步骤:第一步,为工作流中的活动选择执行场所,可以直接指定或者通过数据驱动的方式为活动选择执行场所;第二步,场所管理员根据场所的安全策略,为本场所负责执行的活动指定具体的执行者,可以通过授权规则或直接指定的方式,为活动选定执行者,实现对工作流系统的分布式授权。

关键词 工作流,场所,授权模型

Research on Locale Based Distributed Authorization Model in Workflow System

YU Guang-can LI Rui-xuan LU Zheng-ding SONG Wei TANG Zhuo
(College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract Some existing workflow authorization models ware briefly analyzed, a locale based distributed authorization model applied to workflow system was proposed, The new model can support some application scenes that others models can't support. The authorization of workflow system is divided into two steps. The first step is to choose executing locales for activities of workflow, both directly assigning method and data driven method can be used in the step. The second step is that administrators of locales assign executers to activities based on security policies of locales, both authorization rules and directly assigning methods can be used in the second step. Through the two steps, the distributed authorization of workflow system is implemented.

Keywords Workflow, Locale, Authorization model

1 介绍

工作流是由一组相互之间具有依赖关系的活动组成,可以表达较为复杂的业务流程,在电子商务、电子政务以及制造业等领域有着广泛的应用前景。工作流管理联盟规范[1]中指出,虽然不同的工作流管理系统具有不同的应用范围和不同的实施方式,但是从比较高的层次上来抽象考察工作流管理系统,所有的工作流管理系统都具有以下3种功能:

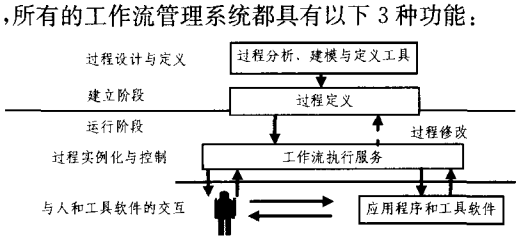

图1 工作流系统特征

(1)建立时期(build-time)功能,通过使用分析、建模工具,把业务过程转化为形式化的、计算机可以处理的过程模型,也可以称为过程定义。

(2)运行时期(run-time)控制功能,解释已定义好的模型,激活相应的人或应用程序,控制功能由工作流引擎(Engine)来实现。

(3)运行时期与用户(通过工具软件)和应用程序交互,实现对工作流中活动的执行。这些功能之间的关系如图1所示。

要使工作流得到更广泛应用的一个重要前提是解决好授权问题。现有的应用于工作流系统的授权模型一般采取以下两种实现方式[2-4]:第一种方式为在过程设计与定义的同时进行授权模型的配置,即所定义的过程模型中包含完整的授权信息,过程模型由工作流执行服务实例化并解释执行,根据过程模型中的授权信息结合组织模型选择授权用户执行相应的活动;第二种方式在过程设计与定义时对授权模型进行初步配置,在过程模型实例化之前通过映射工具将过程模型中的授权信息与组织模型中相关信息进行映射,从而完成对授权模型的配置。这两种方式的共同点就是在过程模型实例化之前,其对应的授权模型已经配置完成,当过程模型由工作流执行服务实例化并开始执行之后,就不能重新配置授权信息。一个典型的工作流过程模型如图2所示,表示一个退税业务流程,各个活动的意义及其授权信息如下:"支票准备"活动由普通职员根据税收情况准备退税支票,"支票预审"活动由副经理进行初步审核,该活动必须由两个不同的副经理分别执

*)国家自然科学基金项目(60403027,60773191,70771043),国家高技术研究发展计划(863计划)项目(2007AA01Z403),中国博士后科学基金项目(20060400846),湖北省自然科学基金项目(2005ABA258),软件工程国家重点实验室开放基金项目(SKLSE05-07),华为科技基金项目(YBIN2006089)。於光灿 博士研究生,主要研究领域为分布式计算、分布式系统安全;李瑞轩 博士,副教授,主要研究领域为分布式异构系统、分布式系统安全;卢正鼎 教授,博士生导师,主要研究领域为分布式计算、软件集成环境、数据库系统、信息安全;宋 伟 博士研究生,主要研究领域为分布式异构系统及安全;唐 卓 博士研究生,主要研究领域为分布式异构系统及安全。

行,“支票复核”由经理综合上述两个副经理的初步审核结果对支票进行复核并得出同意或不同意的结论,最后由职员根据支票复核的结果签发或取消该支票。该过程模型的一个显著特点就是在该过程模型被执行(实例化)之前,对各个活动的授权可以明确确定,比如,“支票准备”活动的执行权限被明确授予职员,而“支票复核”活动被授予经理。

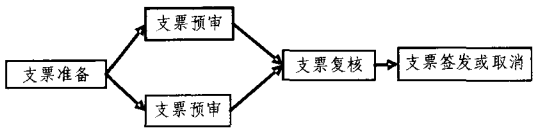

图 2　工作流表示的退税流程

但是,在有些情况下,过程模型在被执行之前不能确定其所包括活动的授权信息。比如多个政府部门联合行政审批工作流程,以申请开餐馆的行政审批流程为例(图 3),整个流程跨多个组织和部门,在如此大范围内实行统一的授权策略是比较困难的事情。一个有效的解决方案是在流程的建立时期只是指定各个活动的执行部门(或场所),当流程被解释执行时,各部门获得相关活动的控制权后,部门再根据其本部门的授权策略对活动进行二次授权,最终确定活动执行者。这里提出场所的概念,可以理解为部门或小组的协同工作的组空间,用户可以在场所里感知其被授权执行的活动,场所的管理员可以进行授权操作,比如指定某人完成某个具体的活动等,其具体含意下文将详细介绍。即对工作流的授权分为二步,第一步指定活动的执行场所,第二步在对应的执行场所对活动进行授权,因为每个场所的管理员和授权策略可能不同,从而实现分布式的授权。这种二步授权方法还适用于一些不太规范的活动,即活动不便于使用授权规则进行自动授权,而需要部门领导或管理员根据各种因素进行综合判断,最后人工对活动进行授权。

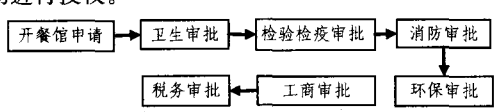

图 3　开餐馆行政审批流程

上述授权方法适用于许多授权较复杂的业务流程场景,但是应用该方法有一个前提条件,那就是在流程开始执行之前,必须确定各个活动的执行场所。有些情况下,在流程开始执行之前,各个活动的执行场所无法确定,必须根据流程执行情况动态地选择活动的执行场所,或者称之为数据驱动的授权,比如政府部门的公文流转系统,流程相对比较固定,但是后续活动的执行场所往往要根据前面活动的执行结果而确定。一个发文业务流程示例如图 4 所示,对于教育部门的发文流程,公文拟制活动可在教育部门的某个科(室)场所被执行,审批活动可以与公文拟制活动使用相同的场所,不过由科(室)领导执行,签发活动在教育部门领导办公场所被执行,传阅活动可能在一个专门的文件传阅场所被组织领导(如市领导)执行,发文活动为一个自动节点,由应用程序自动执行。而其他部门,如农业、卫生等部门,发文流程与图 4 相同,不过活动的执行场所显然与教育部门不同,这就需要数据驱动的授权。数据驱动的授权就是在流程执行过程中,根据流程中定义的控制数据确定活动的执行场所,这是授权的第一步。活动的执行场所确定后,再根据上述的二次授权方法,确定活动的最终执行者。数据驱动授权的最大优势在于动态确定活动的执行场所,使得一个业务流程可以得到最大限度的共享。比如假设政府各部门都采用图 4 所示的发文业务流程,但是由于活动的执行场所及授权方法不同,各个部门只能定制本部门的业务流程及其授权方法,不利于部门之间协作处理公文,而且当业务流程因实际需要更改时,各个部门定制的所有业务流程都要同步修改,工作量较大。利用数据驱动授权选择活动的执行场所,在场所中定义各部门对活动的授权策略实现二次授权,能够灵活实现绝大部分业务流程的授权问题。

在实际工作中,一些活动可能由多个用户共同参与完成,比如图 4 所示的发文业务流程示例中的公文拟制活动,就可能由多人共同参与,因为一些重要的公文往往由多人共同完成。工作流的授权模型就需要考虑到这种可能由多人参与同一个活动的授权情况。对于多人参与同一个活动又有以下 3 种情况要给予充分考虑:第一种情况是多人完全共享与活动相关的所有数据,比如多人参与的公文拟制活动,参与拟制的所有人员就可完全共享正在拟制的公文;第二种情况是多人部分共享活动相关数据,比如多人参与的投票表决活动,投票内容或目标等数据应该共享,而各人的投票数据(赞成或反对等)则可能需要保密,不能共享;第三种情况是完全不能共享数据,这种情况在实际工作中发生的可能性很小。下面就针对上述所提出的几种授权场景给出工作流系统中基于场所的分布式授权模型。

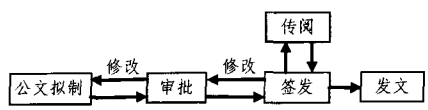

图 4　发文业务流程示例

2　授权模型

为了定义工作流系统中的授权模型,首先引入场所的概念。场所的概念来源于 Fitzpatrick 的基于场所的框架理论[5,6],场所可以理解为可视化的组空间,为用户组提供交互的空间及各种交互手段。因为场所是一个组空间,用户组在场所内通过交互以完成共同的任务,所以场所内需要控制策略以实现对诸如“加入场所”等组操作的控制;另外,对于“场所的创建”等操作的控制不可能在场所内的控制策略中加以实现,所以必须有一个控制策略在场所之外,先于场所而存在,用于确定场所的创建者、控制者。在实际应用中,访问控制往往要考虑到环境的因素,为了满足这一需求,文中引入场所上下文的概念。场所的上下文类似于 Unix 操作系统中环境变量的概念,包括一个名称/值对的集合,场所的上下文提供当前场所的状态信息,如当前时间、会议是否已经开始等。场所的上下文在场所模板中定义,而在不同的具体的场所中,这些上下文变量被赋予不同的值,以表示这些场所所处的状态。场所和场所模板的关系是多对一的关系,即可根据一个场所模板可创建多个场所。场所模板中定义场所内的全局性访问控制策略,定义存在于场所之外的先于场所而存在的一些必要的控制策略,如场所上下文、场所的创建者、控制者等;而场所一经创建,场所的控制者可根据场所的协作需要,在场所全局访问控制策略约束下制定本场所内部的一些自主访问控制策略,实现全局强制性访问控制和场所中的自主访问控制相结合。

在本文提出的工作流系统授权模型中,场所是工作流中活动的执行空间,发挥着重要的作用。授权分为两步:第一

步，是为活动选择执行场所，可以直接指定或者通过数据驱动的方式为活动选择执行场所；第二步是场所管理员根据场所的安全策略为由本场所负责执行的活动指定具体的执行者，可以通过授权规则或直接指定的方式为活动选定执行者。另外，场所的组加入策略也是授权模型不可缺少的重要前提，场所的组加入策略确定哪些人员可以进入场所，场所管理员在这些进入场所的人员中为活动选择执行者。下面就场所的组策略、工作流活动的场所选择及场所内对活动的授权三个方面分别进行描述。在此之前，先对访问控制的主体(在此指用户)进行简单描述。

设用户的集合为 U，用户的基本形式为 $i(h_1,h_2,\cdots,h_n)$，其中 i 为用户的标识符，标识符唯一标识一个用户，$h_1,h_2,\cdots,h_n$ 为用户的属性，用户 i 所有属性的集合称为用户 i 的属性集，记为 $i.A$，属性的个数可以为零个也可以为多个，属性可以包括用户所在部门、岗位、角色、姓名、能力和工龄等相关信息，i_1(姓名="张三"，岗位="××科长"，工龄="10")为标识符为 i_1 用户的示例。用户的属性由属性机构(Attribute Authority，AA)负责维护，属性机构如何为用户指定属性不在本文讨论范围之内，我们只是假设由属性机构管理的所有用户属性集合为 A，任何用户的属性集都是集合 A 的子集，如 $\forall i\in U, i.A\subseteq A$。

2.1 场所的组加入策略

场所的组加入策略是场所管理员根据场所模板的全局性安全策略和本场所的一些具体的安全性需求，而制订的一组规则，用于控制用户加入场所。设场所的集合为 L，$\forall l\in L$，允许加入场所 l 的用户的集合记为 $l.U$，最简单的组加入策略是为场所管理员直接指定允许加入场所的用户，如场所 l 的管理员直接确定集合 $l.U=\{i_1,i_2,i_3\}$，即用户 i_1，i_2 和 i_3 允许加入场所 l，其他用户则被禁止加入。这种直接指定用户的方式虽然简单，但是不便于管理，结合直接指定用户与根据用户的属性自动确定可以加入场所的用户是一个比较好的选择，既方便管理也符合实际工作环境。$\forall l\in L$，场所 l 的组加入策略由三个部件组成：

(1)场所管理员明确允许加入的用户集合，记为 $l.aU$；

(2)场所管理员明确禁止加入的用户集合，记为 $l.dU$；

(3)由属性规则 ρ 允许加入的用户集合，记为 $l.\rho U$，$l.\rho U$ 中的用户为满足属性规则 ρ 的所有用户。设 ρ 为(部门="××科" and 员工性质≠"临时工")，则 $l.\rho U$ 中的用户为"××科"的正式员工。

综合上述3个集合，最终允许加入场所的用户集合 $l.U=(l.aU\cup l.\rho U)-l.dU$，从中可以看出只要是管理员明确禁止加入的用户，即使存在于另外两个允许加入场所的集合中也会被禁止进入场所，以保证场所的安全。

2.2 工作流中活动执行场所的选择

活动是在场所中被执行，为活动选择执行场所是工作流访问控制的第一步。如上文所述，为活动选择执行场所有二种方式，第一种方式是直接指定，第二种方式是数据驱动，下面就这二种方式进行描述。

工作流规范的集合记为 W，$\forall w\in W$，工作流规范 w 的活动集合记为 $w.T$，$<_w$ 可以看作是活动 $w.T$ 的偏序集，如果 $\forall t,t'\in w.T$ 且 $t<_w t'$，则在工作流 w 的任何实例中 t 先于 t' 执行。一般而言，工作流中除了包含一些活动中要处理的业务数据，还包括一些控制数据，这些控制数据可用于控制流程的执行和动态授权。本文只关注用于控制动态授权的数据。$\forall t\in w.T$，活动 t 的控制数据记为 $t.cD$，本文中 $t.cD$ 可用于数据驱动的动态场所选择。

直接为活动指定执行场所的定义为：

$\forall w\in W,(w.T,w.A),w.A\subseteq w.T\times L$

设 $(t,l)\in w.A$，表示指定工作流 w 中的活动 t 执行场所为 l。这种直接指定的方式为静态方法，实现起来较为简单，但是必须在工作流被实例化之前完成，一旦被实例化则无法重新为活动指定新的执行场所。

数据驱动的动态场所选择的定义为：

$\forall w\in W,(w.T,w.f),w.f:w.T\rightarrow L$

其中 $w.f$ 表示工作流 w 的动态场所选择函数，设 $\forall t\in w.T$，函数 f 根据活动 t 的控制数据 $t.cD$ 选择该活动的执行场所。一般而言，控制数据 $t.cD$ 有两种使用方式：第一种方式为一个工作流中的控制数据在初始化时被指定，在工作流的执行过程中保持不变，如上述图4中描述的各部门共享使用的发文使用流程，如果用户在公文拟制活动中设定控制数据为"教育局"，则后继的活动通过配置文件或代码自动选择教育局发文相关场所；第二种方式前一个活动为后一个活动指定执行场所，在这种方式中，活动执行者对工作流的控制能力更强。

2.3 场所中活动的执行者选择

在完成工作流授权的第一步(为活动选择执行场所)后，场所获得对活动的控制权，场所管理员根据场所的安全策略，为本场所负责执行的活动指定具体的执行者。根据活动的性质，为活动指定执行者主要有以下两种方式：第一种方式，场所管理员将工作流的活动指定给执行者，被指定的执行者获得工作流所有实例对应活动的执行权限，这种方式适应于工作流的业务流程比较规范，管理员可以根据组织分工将工作流的所有实例对应活动一次性指定给相关执行者；第二种方式管理员将工作流的某个实例的某个活动指定给执行者，被指定的执行者只获得工作流一个实例对应活动的执行权限，这种方式适应于工作流的业务流程不很规范，管理员只能根据工作流实例的具体情况将实例的活动指定给执行者，但是这种方式人对工作流的控制能力最强，能够适应于最复杂的授权情况。

设 $\forall w\in W$，$\forall t\in w.T$，工作流活动执行者指定策略由以下6个部件组成：

(1)场所管理员明确指定可以执行活动 t 的用户集合，记为 $t.aU$；

(2)场所管理员明确禁止执行活动 t 的用户集合，记为 $t.dU$；

(3)由属性规则 $t.\eta$ 确定可以执行活动 t 的用户集合，记为 $t.\eta U$。$t.\eta U$ 中的用户为满足属性规则 $t.\eta$ 的所有用户的集合，设 $t.\eta$ 为能力>"一般"，则 $t.\eta U$ 中包括那些"能力"属性为"强"的人员，特别值得注意的是属性规则 $t.\eta$ 的应用范围是已经加入场所的人员，这些人员在进入场所之间已经通过场所的组加入策略的过滤。

(4)申请执行活动 t 的用户的集合，记为 $t.rU$。因为在有些应用场景中，管理员根据场所内用户的申请对活动进行分配，因此本文定义用户对活动的请求集合。管理员可以对用户的申请进行批准操作，获得批准的请求用户将被存入集合 $t.aU$。

综合上述4个部件，最终被允许执行活动 t 的用户的集合 $t.U=(t.aU\cup t.\eta U)-t.dU$。

在对活动进行授权的工作中，除了确定活动的执行者这

一基本步聚外，还要考虑优先级问题，多个用户符合执行活动的条件，到底由哪个(些)用户最终执行该活动。优先级是一个比较综合性的问题，除了要考虑用户的固有属性信息，如岗位、能力等，还要考虑到用户在场所中的一些具体情况，比如，用户的工作量、所承担活动的完成情况等，综合这些元素，给出用户的优先级。

(5)定义集合 $t.U$ 上的偏序集$>$表示用户的优先级，$\forall u_1, u_2 \in t.U$，如果 $u_1 > u_2$，则用户 u_1 的优先级大于用户 u_2 的优先级，如果 $u_1 \not> u_2$ 并且 $u_2 \not> u_1$，则用户 u_1, u_2 有着相同的优先级。

上文所述的方法将工作流的活动指定给执行者，被指定的执行者可能获得工作流所有实例对应活动的执行权限。为了方便描述，我们将工作流实例的活动称为活动实例，当场所获得活动实例的控制权后，根据上述规则，自动选定执行者以执行活动实例。但是，如果工作流的业务流程不很规范，管理员无法在场所取得活动实例的控制权之前制定上述规则，而必须在场所取得活动实例的控制权之后，根据活动实例的具体情况将活动实例指定给执行者。

(6)用 $w.I$ 表示工作流 w 的所有实例，$\forall i \in w.I$，设 ti 为工作流实例 i 的某个活动实例，则场所管理员为活动实例 ti 指定的用户集合记为 $ti.U$，之所以指定一个用户集合，是考虑到有些活动可能是多用户参与活动。

除了为活动选择执行者，工作流系统授权模型另一个重要部分是授权约束。常用约束包括职责分离约束和职责绑定约束，职责分离约束指某两个活动必须由两个不同的用户执行，职责绑定约束指某两个活动必须由同一个用户执行。Crampton 在文献[7,8]中提出约束模型，该模型的基本形式为$(D,(t,t'),\rho)$，其中 $D \subseteq U$，$\rho \in Rel(U)$，$Rel(U) = 2^{U \times U}$，t 为先序活动，t' 为后序活动，设用户 u, u' 分别执行活动 t, t'，约束$(D,(t,t'),\rho)$得到满足的条件是$(u,u') \in \rho$。如$(\{张三\},(t_1, t_2),\neq)$表示用户张三如果已经执行任务 t_1 则不允许执行任务 t_2。由于该约束模型表达能力强、适应范围广，所以本文所采用该模型以实现授权约束。

3 系统原型

原型系统采用 B/S 架构，系统结构如图 5 所示。工作流子系统由工作流引擎、授权模块、流程定义工具和工作流控制台等部分组成，我们采用 Jboss 的 Java 开源项目 jBMP[9] 作为工作流系统的基础部分，在 jBMP 基础上进行二次开发。jBMP 是采用面向图形编程技术的轻量级工作流产品，包括工作流引擎(java 库文件)、基于 eclipse 插件的流程定义工具和控制台 Web 应用程序等部件，并提供开放的授权接口。工作流子系统的授权模块由策略管理点、管理执行点和策略库组成，工作流子系统的安全管理员在策略管理点完成上述授权的第一步，即为活动指定执行场所，制定的授权规则存放在策略库中，策略执行点根据策略库中的授权规则允许或禁止用户对活动的访问。如上文所述，为活动指定执行场所分为直接指定和数据驱动两种情况，直接指定方式可以在流程定义中实现也可以在策略库中实现，而数据驱动方式必须在策略库中实现，所以流程定义工具和策略库都要引用场所模型。因为对场所的管理不是本文的重点，所以图 5 的系统结构中只是给出了场所模型，没有给出具体的场所管理组件。

场所子系统由组应用程序和授权模块组成，组应用程序包括活动执行程序、即时通信、视频会议和电子白板等多用户协作应用系统，授权模块由策略管理点、组加入执行点和组加入策略库等组成，场所管理员在策略管理点完成以下两项任务：第一项任务为制定组加入策略，以确定可以加入场所的用户，因为制定组加入策略要考虑到用户的属性信息，所以策略管理点要引用组织模型，制定的组加入策略存放在组加入策略库中，组加入执行点根据组加入策略库中的规则允许或禁止用户进入场所并执行相关的组应用程序；第二项任务为完成对工作流授权的第二步，即为活动或活动实例选择执行者，通过 Web Services 访问工作流子系统授权模块的策略库，场所管理员只能管理策略库中那些被指派到本场所的活动，实现分布式授权，对于那些被指派给场所的，而场所管理员没有在策略库中给出对应的用户选择规则的活动，当有新的活动实例产生时，工作流控制台通知对应场所的策略管理点，再由场所的管理员为该活动实例选择执行者。

因为整个系统由工作流子系统和多个场所子系统组成，各子系统物理上是分布的，逻辑上也相对较独立。我们采用单点登录方法来实现统一的用户认证，各子系统共享组织模型和用户信息，系统采用 Yale 的 java 开源项目 CAS 实现单点登录[10]。

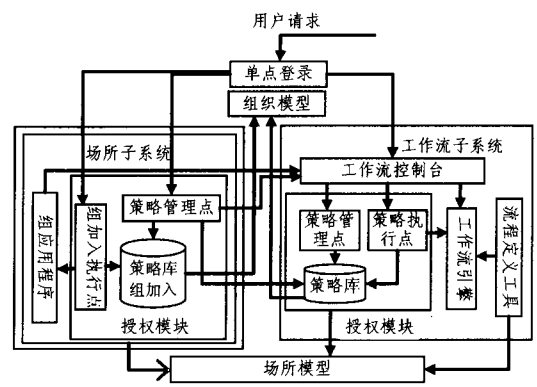

图 5 原型系统结构图

结束语 本文对现有工作流系统授权模型进行了简要分析，给出几种适用于实际工作环境，但是现有授权模型并不能很好地支持工作流系统授权情景，在此基础上提出工作流系统中基于场所的分布式授权模型。我们首次将场所的概念应用于工作流系统的授权模型中，授权分为两步：第一步为活动选择执行场所，该项操作由工作流子系统安全管理员完成，可以直接指定或者通过数据驱动的方式为活动选择执行场所；第二步场所管理员根据场所的安全策略，为由本场所负责执行的活动指定具体的执行者，该项操作由各场所的安全管理员完成，可以通过授权规则或直接指定的方式为活动选定执行者，从而实现对工作流系统的分布式授权。模型不仅适用于工作流业务流程比较规范，可以通过授权规则的方式为活动选择执行者的情况，还适应于工作流的业务流程不很规范，管理员只能根据工作流实例的具体情况将实例的活动指定给执行者的情况，充分体现人对工作流的控制能力，使得模型适应于复杂的授权情景。文中最后出了系统原型，对本文提出的模型进行了实际验证。

参 考 文 献

[1] Workflow Management Coalition. The workflow reference model [Z]. Document Number TC20021003, Jan. 1995(1)

(下转第 266 页)

迁 complete，即整个过程的 BPEL 代码注释为 complete，因此，我们提供的迭代方法可以把工作流网转换为可读的 BPEL 模板代码。

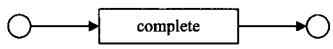

图 6 简化流(flow)结构为一个变迁后的工作流网

结束语 本文给出了有色 Petri 网协作模型的 BPEL 代码实现算法：首先，把有色 Petri 网协作模型转换为结构 WF-net 模型；接着，使用归纳法把 WF-net 模型的子结构替换成标注有 BPEL 代码的变迁。

在案例研究中运用转换算法，首先根据电话机故障修理的需求，创建了该案例的有色 Petri 网协作模型；接着经过分析和简化之后把原始的有色 Petri 网协作模型转换为结构 WF-net 模型；最后，实现了 WF-net 模型的 BPEL 代码实现，显示了转换算法的实用性。

由于主要的目标是产生可读的、易维护的 BPEL 代码，因此没有针对复杂的完全转换(例如，没有使用事件处理结构而是识别典型的 BPEL 结构)。

未来的工作将引入辅助变量或者借鉴程序设计方法学，把无结构的 WF-net 模型转换为 BPEL 代码。

参 考 文 献

[1] Arkin A，Askary S，Bloch B，et al. Web Services Business Process Execution Language Version 2.0. Working Draft. WS-BPEL TC OASIS，May 2005

[2] Jensen K. An Introduction to the Theoretical Aspects of Coloured Petri Nets. http://www.daimi.au.dk/~kjensen/papers_books/rex.pdf

[3] Object Management Group. Unified Modeling Language(UML) 2.1.1. http://www.omg.org/technology/documents/formal/uml.htm

[4] 袁崇义. Petri 网原理与应用[M]. 北京：电子工出版社，2005

[5] http://is.tm.tue.nl/~cgunther/dev/prom/

[6] CPN Group，University of Aarhus，Denmark. CPN Tools Home Page. http://wiki.daimi.au.dk/cpntools/

[7] Ullman J D. Elements of ML Programming. Prentice-Hall，1998

(上接第 235 页)

[2] Bertino E，Ferrari E，Atluri V. The specification and enforcement of authorization constraints in workflow management systems. ACM Transactions on Information and System Security，1999，2(1)：65-104

[3] Wainer J，Barthelmess P，Kumar A. W-RBAC-A workflow security model incorporating controlled overriding of constraints. International Journal of Cooperative Information Systems，2003，12(4)：455-486

[4] Enhydra Shark. http://www.enhydra.org/workflow/shark/index.html

[5] Fitzpatrick G. The Locales Framework：understanding and Designing for Cooperative Work. Australia：Ph.D. Thesis. The Univ. of Queensland，1999

[6] Fitzpatrick G，Mansfield T，Kaplan S. Locales Framework Exploring foundations for collaboration support // IEEE Proceedings of OzCH1. Hamilton，New Zealand，1996：34-41

[7] Crampton J. A reference monitor for workflow systems with constrained task execution // Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies (SACMAT 2005). Stockholm，Sweden，2005：38-47

[8] Crampton J. An algebraic approach to the analysis of constrained workflow systems // Proceedings of the 3rd Workshop on Foundations of Computer Security. Turku，Finland，2004：61-74

[9] Jboss jBMP. http://www.jboss.com/products/jbpm

[10] Yale CAS. http://www.ja-sig.org/products/cas/index.html

(上接第 250 页)

[2] Henricksen K，Indulska J. A Software Engineering Framework for Context-aware Pervasive Computing // Pervasive Computing and Communications，2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference. 2004：77-86

[3] Brown P J，Jones G J F. Context-aware retrieval：exploring a new environment for information retrieval and information filtering. Personal and Ubiquitous Computing，2001，5(4)：253-263

[4] Pascoe J. The stick-e note architecture：extending the interface beyond the user // Proceedings of International Conference on Intelligent User Interfaces. Orlando，USA，1997：261-264

[5] Ranganathan A，Chetan S. Olympus--A High-level Programming Model for Pervasive Computing Environments // Pervasive Computing and Communications，2005. PerCom 2005. Third IEEE International Conference. 2005：7-16

[6] Keays R，Rakotonirainy A. Context-oriented Programming // Proceedings of the 3rd ACM International Workshop. San Diego，CA，USA，2003

[7] Rakotonirainy A. Context-oriented Programming for pervasive space // Proceedings of the ACM Dynamic Languages Symposium. 2005

[8] Román M，Hess C，Cerqueira R，et al. Gaia：A middleware infrastructure to enable active space. IEEE Pervasive Computing Magazine，2002，1：74-83

[9] Capra L，Emmerich W，Mascolo C. CARISMA：Context-aware Reflective Middleware System for Mobile Applications. IEEE transactions on software engineering，2003

[10] Capra L，Emmerich W，Mascolo C. A Micro-economic Approach to Conflict Resolution in Mobile Computing // SIGSOFT. 2002

[11] Arntzen I M. A Programmable Structure for Pervasive Computing // Proceedings of the IEEE/ACS International Conference on Pervasive Services. 2004