

# Process for the Enactment of Workflow Using Role Based Access Control

John Barkley  
National Institute of Standards and Technology  
Gaithersburg MD 20899  
(301) 975-3346  
jbarkley@nist.gov

September 9, 1996

## Abstract

Role Based Access Control is an access control mechanism whose use is increasing in organizations. The primary reason for this is RBAC's ease of administration as compared to other access control mechanisms. The use of workflow technology is also increasing within organizations. Workflow technology is the means by which business processes are automated. Workflows consist of a set of activities carried out in a predefined order. As such, access control becomes an integral part in the enactment of a workflow. A process for the enactment of workflow using RBAC is presented. This process provides not only access control for each activity in a workflow but also the proper sequencing of activities as specified in the process definition for the business process which the workflow automates.

## 1 Introduction

Role Based Access Control is an access control mechanism whose use is increasing in organizations. The primary reason for this is RBAC's ease of administration as compared to other access control mechanisms. With RBAC, access is based on a user's role within an organization. Consequently, access control administration is at a level of abstraction that is natural to the way that organizations typically conduct business.

Administrators typically think of their organizations in terms of the roles adopted by individuals and the access permitted to those roles. Other access control mechanisms require that administrators translate their normal organizational view into the access control mechanism. With RBAC, an administrator's organizational view is the access control mechanism.

The use of workflow technology is also increasing within organizations. Workflow technology is the means by which business processes are automated. A business process involves the transfer of one or more documents, information, or tasks between participants according to a set of

procedural rules in order to achieve business goals. Workflow technology consists of a set of tools to define and manage business processes. A workflow is the complete or partial automation of a business process.

Workflows consist of a set of activities carried out in a predefined order. As such, access control becomes an integral part in the enactment of a workflow. Each activity requires privileged operations, the access to which is restricted to authorized user(s) who participate in that activity. Moreover, the privileged operations permitted to a user may change as a workflow is processed. For example, an activity involving the purchase of an article of equipment is only permitted to a user until the purchase has been completed whereupon the permission for that user to purchase the equipment is removed. Such requirements for administering access as the workflow progresses suggests the use of RBAC as the access control mechanism.

RBAC can also be used as a means of insuring that the activities which make up a workflow are carried out in the correct sequence. This paper presents a process for the enactment of workflow using RBAC. This process provides not only access control for each activity in a workflow but also the proper sequencing of activities as specified in the process definition for the business process which the workflow automates.

## 2 Terminology

When referring to RBAC concepts, this paper uses terminology taken from the Sandhu paper[1]. The RBAC model used is the  $RBAC_0$  model from the Sandhu paper. This model is generally considered to be the minimal functionality required in order for an access control mechanism to be called an RBAC mechanism.

When referring to workflow concepts, this paper uses terminology taken from the Workflow Management Coalition Specification[2]. In this section, the definition of terms used in this paper and taken from the sources listed above are presented.

### 2.1 RBAC

**Access Control** The process of limiting access to the resources of a system only to authorized programs, processes, or other systems (in a network).

**Object** A passive entity that contains or receives information.

**Permissions** A description of the type of authorized interactions a subject can have with an object.

**Resource** Anything used or consumed while performing a function. The categories of resources are time, information, objects, or processors.

**Role** A job function within an organization that describes the authority and responsibility conferred on a user assigned to the role. In this paper, the term *role* also refers to an abstraction which is created to identify the function of an activity within a business process.

**Session** A mapping between a user and an activated subset of the set of roles to which the user is assigned.

**Subject** An active entity, generally in the form of a person, process, or device, that causes information to flow among objects or changes the system state.

**User** Any person who interacts directly with a computer system. In this paper, the term *user* also refers to a computer process which may or may not represent a person.

## 2.2 Workflow

**Activity** A description of a piece of work that forms one logical step within a process. An activity is typically the smallest unit of work which is scheduled by a workflow engine during process enactment (e.g. using transition and pre/post-conditions), although one activity may result in several work items being assigned (to a workflow participant)

**AND-Split** A point within the workflow where a single thread of control splits into two or more parallel activities.

**AND-Join** A point in the workflow where two or more parallel executing activities converge into a single common thread of control.

**Business Process** A set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships.

**Invoked Application** An invoked application is a workflow application that is invoked by the workflow management system to automate an activity, fully or in part, or to support a workflow participant in processing a workitem.

**Parallel Routing** A segment of a process instance under enactment by a workflow management system, where two or more activity instances are executing in parallel within the workflow, giving rise to multiple threads of control.

**Process Definition** The representation of a business process in a form which supports automated manipulation, such as modeling, or enactment by a workflow management system. The process definition consists of a network of activities and their relationships, criteria to indicate the start and termination of the process, and information about the individual activities, such as participants, associated IT applications and data, etc.

**Sequential Routing** A segment of a process instance under enactment by a workflow management system, in which several activities are executed in sequence under a single thread of execution. (No -split or -join conditions occur during sequential routing.)

Perform invoked application  $IA_{A_{j,k}}(wf)$  which automates activity  $A_{j,k}$   
Send completion message to  $P4EW/R(wf)$  indicating success or error

Figure 1: Operation  $OP_{A_{j,k}}(wf)$  associated with each activity  $A_{j,k}$

**Workflow** The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.

**Workflow Management System** A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications.

### 3 Process Description

In a system that supports RBAC, the role is the means by which access to a resource is determined. In RBAC, access to a resource by a user is permitted only if:

1. the permission required for access to the resource is assigned to a role; and
2. that role is assigned to the user requesting access to the resource; and,
3. that role is activated in the user's session.

In addition to a role's use for access control, a role may be used to refer to the set of operations to which the permission(s) associated with that role grants access. Some implementations of RBAC make use of this concept by presenting the role as a menu choice to the user.

Roles can be used in still another way. Because a role is the means by which access to a resource can be enforced, assignment of a permission to perform an operation and the removal of such an assignment can be used as a means to sequence a set of operations. The sequencing of operations is the fundamental behavior required to support workflow. Thus, an RBAC mechanism can be used as a means of implementing workflow.

#### 3.1 Brief

The process definition for a business process can be partitioned into sequential routing segments and parallel routing segments. A sequential routing segment has one or more activities which must proceed in a strictly sequential manner. A parallel routing segment has two or more activities which can proceed in parallel. The workflow specified by a process definition is managed by a workflow management system which enacts each segment in the order specified by that process definition. An RBAC system can form the basis for the enactment of workflow, i.e., an RBAC system may be used as the basis for a workflow management system.

Decompose  $wf$  into sequential and parallel segments  $S_j$

For each segment  $S_j$  of  $wf$ :

If  $S_j$  is a sequential routing segment:

    Create  $ROLE_{S_j}$  in the RBAC System

    For each activity  $A_{j,k}, k = 1, \dots, N_{S_j}$  in  $S_j$ :

        Assign permission to perform operation  $OP_{A_{j,k}}(wf)$  to  $ROLE_{S_j}$

        Assign  $ROLE_{S_j}$  to  $USER_{A_{j,k}}$

        Enable the capability for  $OP_{A_{j,k}}(wf)$  to be activated

        Sleep, resuming at next line when completion message received from  $OP_{A_{j,k}}(wf)$

        Remove assignment of  $ROLE_{S_j}$  from  $USER_{A_{j,k}}$

        Remove permission to perform operation  $OP_{A_{j,k}}(wf)$  from  $ROLE_{S_j}$

        Disable the capability for  $OP_{A_{j,k}}(wf)$  to be activated

        If completion message indicated error:

            notify  $P4EW/R(wf)$  administrator or terminate

    Remove  $ROLE_{S_j}$  from the RBAC System

If  $S_j$  is a parallel routing segment:

    For each activity  $A_{j,k}, k = 1, \dots, N_{S_j}$  in  $S_j$ :

        Create  $ROLE_{A_{j,k}}$  in the RBAC System

        Assign permission to perform operation  $OP_{A_{j,k}}(wf)$  to  $ROLE_{A_{j,k}}$

        Assign  $ROLE_{A_{j,k}}$  to  $USER_{A_{j,k}}$

        Enable the capability for  $OP_{A_{j,k}}(wf)$  to be activated

    while not all  $OP_{A_{j,k}}(wf)$  completed:

        Sleep, resuming at next line when completion message received from any  $OP_{A_{j,k}}(wf)$

        Remove assignment of  $ROLE_{A_{j,k}}$  from  $USER_{A_{j,k}}$

        Remove permission to perform operation  $OP_{A_{j,k}}(wf)$  from  $ROLE_{A_{j,k}}$

        Remove  $ROLE_{A_{j,k}}$  from the RBAC System

        Disable the capability for  $OP_{A_{j,k}}(wf)$  to be activated

    If completion message indicated error:

        For all  $OP_{A_{j,k}}(wf)$  still active:

            Terminate  $OP_{A_{j,k}}(wf)$

            Remove assignment of  $ROLE_{A_{j,k}}$  from  $USER_{A_{j,k}}$

            Remove permission to perform operation  $OP_{A_{j,k}}(wf)$  from  $ROLE_{A_{j,k}}$

            Remove  $ROLE_{A_{j,k}}$  from the RBAC System

            Disable the capability for  $OP_{A_{j,k}}(wf)$  to be activated

        notify  $P4EW/R(wf)$  administrator or terminate

    Record the completion of  $OP_{A_{j,k}}(wf)$

Figure 2:  $P4EW/R(wf)$  - Process for the enactment of workflow using RBAC

The process for using RBAC to enact workflow is to first partition a workflow into sequential routing and parallel routing segments. The segments are processed in the order specified in the process definition for the workflow.

Given a sequential routing segment, a role unique to that segment is created. For each activity in turn in the segment, permission to perform that activity is assigned to the role and the role is assigned to a user who performs the activity. The activity is initiated as a result of these assignments and the role unique to the segment becomes activated in the user's session for that activity. Once an activity has completed, the assignments related to that activity are removed and assignments made for the next activity in the sequential routing segment. The role unique to the segment is passed between the activities as sort of a *token* as a means of processing the activities sequentially in the order specified by the process definition. When all activities in the sequential routing segment have been completed in the order specified, the next segment in the workflow is processed.

Given a parallel routing segment, for all activities in that segment, a role unique to each activity in the segment is created. Permission to perform each activity is assigned to the unique role for that activity and the role unique to that activity is assigned to the user who performs the activity. Once these assignments have been made to all activities in the parallel routing segment, all of these activities are enabled for activation such that they may all execute in parallel. All activities are activated in a manner such that each activity's unique role is activated for each activity's session. When all activities in the segment have completed in any order, the next segment in the workflow is processed.

### 3.2 Detailed

Given a business process automated by workflow  $WF$ , the process for the enactment of workflow using RBAC,  $P4EW/R(wf)$ , conducts the performance of  $WF$ . The parameter  $wf$  for  $P4EW/R(wf)$  contains all of the information about the workflow to be performed. For the workflow  $WF$ ,  $WF$  is passed as the argument to  $P4EW/R(wf)$ .

Associated with each activity  $A_{j,k}$  of  $WF$  is an operation  $OP_{A_{j,k}}(WF)$  and a  $USER_{A_{j,k}}$  where  $j$  is the segment number and  $k$  is the activity number within segment  $j$ . As shown in figure 1,  $OP_{A_{j,k}}(wf)$  performs the invoked application  $IA_{A_{j,k}}(wf)$  associated with activity  $A_{j,k}$  and upon completion, sends a message with appropriate completion information to  $P4EW/R(wf)$ . The parameter  $wf$  in  $OP_{A_{j,k}}(wf)$  contains all information required by the invoked application to process the specific workflow  $wf$ . For the workflow  $WF$ ,  $WF$  becomes the argument to  $OP_{A_{j,k}}(wf)$ .

$OP_{A_{j,k}}(wf)$  encapsulates the invoked application for  $A_{j,k}$  so that the invoked application's access control and execution can be enacted by  $P4EW/R(wf)$ . Figure 2 shows the process by which  $P4EW/R(WF)$  enacts workflow  $WF$ .  $P4EW/R(wf)$  makes use of the administrative tools of the underlying RBAC system.

Very often an activity in a business process is associated with a human user. In the figures, the term  $USER_{A_{j,k}}$  refers to this human user or where there is no human user,  $USER_{A_{j,k}}$  refers to the computer process involved in performing the activity. The term  $USER_{A_{j,k}}$  ultimately references the "owner" of a process within a computer. The owner of a computer process is

identified with a unique “user ID” or “username” within the operating system.

In practice, within an RBAC system implementation, owners of processes are the ultimate recipients of role assignments regardless of whether the process owners are associated with human users or computer processes. This must be the case since computer processes represent human users. This use of the term *user* is somewhat at variance with the terminology in Sandhu[1] but is consistent with RBAC concepts in that, in practice, a role is always ultimately associated with a computer process, the owner of which (i.e., the “user”) may or may not be a human user.

When  $P4EW/R(WF)$  begins, it decomposes  $WF$  into segments, some of which are sequential routing segments and some of which are parallel routing segments. If  $S_j$  is a sequential routing segment, it creates the role  $ROLE_{S_j}$  which is unique in the RBAC system.  $ROLE_{S_j}$  is removed by  $P4EW/R(WF)$  after sequential routing segment  $S_j$  is processed. This is to ensure that  $ROLE_{S_j}$  does not conflict with other roles within the RBAC system and only exists within the RBAC system while  $S_j$  is processed. At this point,  $ROLE_{S_j}$  is defined but has no permissions assigned and is not assigned to any users.

For each activity  $A_{j,k}$  in  $S_j$  taken in the order specified in the process definition for the business process automate by workflow  $WF$ :

1. permission to perform operation  $OP_{A_{j,k}}(WF)$  is assigned to  $ROLE_{S_j}$ ; and,
2.  $ROLE_{S_j}$  is assigned to  $USER_{A_{j,k}}$ ; and,
3. the conditions necessary for  $OP_{A_{j,k}}(WF)$  to be activated are established.

At this point,  $OP_{A_{j,k}}(WF)$  has been enabled for activation. The activation of  $OP_{A_{j,k}}(WF)$  can occur in several different ways depending on the nature of the activity. If there is a human user associated with  $OP_{A_{j,k}}(WF)$ , an entry referring to  $OP_{A_{j,k}}(WF)$  might be added to the user’s menu, in which case,  $OP_{A_{j,k}}(WF)$  is activated by a human interaction. In order for this type of activation to be successful,  $ROLE_{S_j}$  must be active within the user’s session.

If there is no human user involved,  $OP_{A_{j,k}}(WF)$  might be scheduled to execute or  $P4EW/R(WF)$  may directly activate  $OP_{A_{j,k}}(WF)$ . Regardless of the way in which  $OP_{A_{j,k}}(WF)$  is activated,  $USER_{A_{j,k}}$  is the owner of the computer process performing  $OP_{A_{j,k}}(WF)$ .

$OP_{A_{j,k}}(WF)$  begins execution successfully because:

1.  $USER_{A_{j,k}}$  is the owner of the computer process performing  $OP_{A_{j,k}}(WF)$ ; and,
2.  $USER_{A_{j,k}}$  has been assigned  $ROLE_{S_j}$ ; and,
3.  $ROLE_{S_j}$  is active in  $USER_{A_{j,k}}$ ’s session; and,
4.  $ROLE_{S_j}$  has been assigned permission to execute  $OP_{A_{j,k}}(WF)$ .

Because only  $ROLE_{S_j}$  has been assigned permission to perform  $OP_{A_{j,k}}(WF)$  and only  $USER_{A_{j,k}}$  has been assigned  $ROLE_{S_j}$ , the RBAC system ensures that only  $USER_{A_{j,k}}$  is able to access operation  $OP_{A_{j,k}}(WF)$ .

Having activated  $OP_{A_{j,k}}(WF)$ ,  $P4EW/R(WF)$  sleeps until it receives a message from  $OP_{A_{j,k}}(WF)$  indicating completion. When  $P4EW/R(WF)$  resumes after a completion message from  $OP_{A_{j,k}}(WF)$  regardless of whether the message indicated an error, it:

1. removes the assignment of  $ROLE_{S_j}$  from  $USER_{A_{j,k}}$ ; and,
2. removes the permission to perform operation  $OP_{A_{j,k}}(WF)$  from  $ROLE_{S_j}$ ; and,
3. removes the capability for  $OP_{A_{j,k}}(WF)$  to be activated.

If completion was unsuccessful,  $P4EW/R(WF)$  notifies its administrator for further instructions or terminates.

Each activity  $A_{j,k}$  in sequential routing segment is processed in the above manner. When all activities a sequential routing segment have been processed,  $P4EW/R(WF)$  removes  $ROLE_{S_j}$  from the RBAC system and proceeds to the next segment.

If  $S_j$  is a parallel routing segment, then for each  $A_{j,k}$ ,  $P4EW/R(WF)$ :

1. creates a unique role  $ROLE_{A_{j,k}}$ ; and,
2. assigns permission to perform  $OP_{A_{j,k}}(WF)$  to that role; and,
3. assigns that role to  $USER_{A_{j,k}}$ ; and,
4. enables an activation capability for  $OP_{A_{j,k}}(WF)$ .

For a parallel routing segment, a separate role unique to each activity  $A_{j,k}$  must be created since all  $A_{j,k}$  execute virtually simultaneously. If there were only one unique role for the parallel routing segment, as is the case for a sequential routing segment, any  $USER_{A_{j,k}}$  associated with the parallel segment could access any  $OP_{A_{j,k}}(WF)$  associated with the parallel routing segment since all permission/role assignments are made to all  $OP_{A_{j,k}}(WF)$  and all  $USER_{A_{j,k}}$  at the same time. In a sequential routing segment, the permission/role assignments are only made to a single  $OP_{A_{j,k}}(WF)$  and  $USER_{A_{j,k}}$  at a time.

Once all of the  $OP_{A_{j,k}}(WF)$  in a parallel routing segment have been enabled for activation,  $P4EW/R(WF)$  sleeps. As each message from an  $OP_{A_{j,k}}(WF)$  arrives<sup>1</sup>,  $P4EW/R(WF)$ :

1. removes the permission/role assignments associated with  $OP_{A_{j,k}}(WF)$ ; and,
2. removes the capability for  $OP_{A_{j,k}}(WF)$  to be activated; and,
3. removes the unique role  $ROLE_{A_{j,k}}$  from the RBAC system; and,
4. records the completion of  $OP_{A_{j,k}}(WF)$ .

When all  $OP_{A_{j,k}}(WF)$  in the parallel segment have completed successfully,  $P4EW/R(WF)$  proceeds to the next segment.

A completion with an error implies that  $P4EW/R(WF)$  must terminate all  $OP_{A_{j,k}}(WF)$  that are still active and remove the roles and role/permission assignments that were made on behalf of the remaining active operations  $OP_{A_{j,k}}(WF)$ .  $P4EW/R(WF)$  then generates a notification to an administrator or terminates the workflow engine.

When all segments in  $WF$  have been performed successfully in the order specified,  $P4EW/R(WF)$  terminates.  $WF$  has successfully completed.

---

<sup>1</sup>Note that  $P4EW/R(wf)$  must execute in an environment which supports messaging in such a way that messages are queued rather than being lost if the process is unable to immediately accept the message.

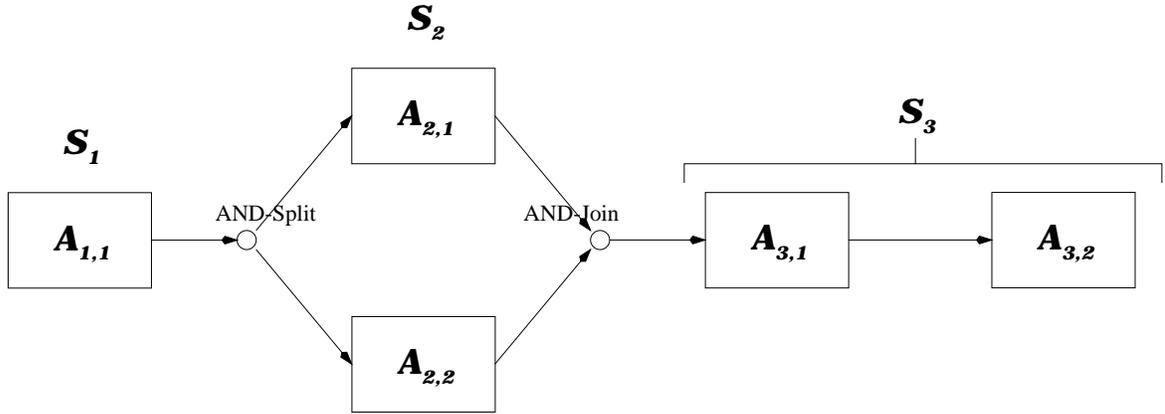


Figure 3: Workflow example: *WPR* - generation/approval of purchase request

## 4 Example

The generation and approval of a purchase request is a simple example of a business process. This business process is automated by the workflow illustrated in figure 3. It consists of:

### Sequential Routing Segment $S_1$ :

**Activity  $A_{1,1}$**  A member of a project team, the requisitioner, initiates a purchase request (PR) by creating an electronic PR form and digitally signing the form. The form now goes to the other two members of the project team for their digital signatures indicating their approval.

### Parallel Routing Segment $S_2$ :

**Activity  $A_{2,1}$**  Second member of the project team digitally signs.

**Activity  $A_{2,2}$**  Third member of the project team digitally signs.

### Sequential Routing Segment $S_3$ :

**Activity  $A_{3,1}$**  The project manager digitally signs indicating approval.

**Activity  $A_{3,2}$**  The division manager digitally signs indicating approval.

Once all of the signatures have been obtained, the PR form goes to the purchasing department.

This workflow example has a parallel routing segment and involves human interaction. Consequently, there can be more than one possible sequence of events that takes place as this workflow is processed. The following presents one possible sequence of events as the process of enacting workflow using RBAC is applied in this example.

**First member** of the project team selects a menu item to purchase a widget. This action initiates  $P4EW/R(WPR)$  to enact the workflow  $WPR$  for the first member's generation and approval of a purchase request for a widget. The notation " $WPR$ " uniquely identifies the workflow for this specific purchase request initiated by the first member of the project team and thus,  $WPR$  becomes the argument to  $P4EW/R(wf)$ .

$P4EW/R(WPR)$  creates role  $ROLE_{S_1}$ , assigns permission to perform  $OP_{A_{1,1}}(WPR)$  to role  $ROLE_{S_1}$ , assigns role  $ROLE_{S_1}$  to the first member of the project team, activates  $ROLE_{S_1}$  in the first member's session, activates  $OP_{A_{1,1}}(WPR)$ , and sleeps.  $OP_{A_{1,1}}(WPR)$  performs the invoked application that automates activity  $A_{1,1}$ .

**First member** of the project team fills in the purchase request form and signs it under the direction of the invoked application  $IA_{A_{1,1}}$  initiated by  $OP_{A_{1,1}}(WPR)$ . The successful completion of  $IA_{A_{1,1}}$  results in  $OP_{A_{1,1}}(WPR)$  sending a successful completion message to  $P4EW/R(WPR)$ .

$P4EW/R(WPR)$  is awakened. As a result of receiving the successful completion message from  $OP_{A_{1,1}}(WPR)$ , it removes role  $ROLE_{S_1}$  and its associated assignments. It now creates role  $ROLE_{A_{2,1}}$ , makes the assignments necessary for the second member of the project team to perform  $OP_{A_{2,1}}(WPR)$ , and adds an entry for  $OP_{A_{2,1}}(WPR)$  to the second member's menu. When the second member selects  $OP_{A_{2,1}}(WPR)$  from the menu,  $OP_{A_{2,1}}(WPR)$  is activated. In addition, it creates role  $ROLE_{A_{2,2}}$ , makes the assignments necessary for the third member of the project team to perform  $OP_{A_{2,2}}(WPR)$ , and adds an entry to select  $OP_{A_{2,2}}(WPR)$  to the third member's menu. When the third member selects  $OP_{A_{2,2}}(WPR)$  from the menu,  $OP_{A_{2,2}}(WPR)$  is activated.  $P4EW/R(WPR)$  sleeps.

**Third member** of the project team chooses the added menu entry for  $OP_{A_{2,2}}(WPR)$  in a session, reviews the purchase request for the widget, and signs. This action causes  $OP_{A_{2,2}}(WPR)$  to send a successful completion message to  $P4EW/R(WPR)$ .

$P4EW/R(WPR)$  is awakened by the receipt of the message indicating the successful completion of  $OP_{A_{2,2}}(WPR)$ , removes role  $ROLE_{A_{2,2}}$  and its associated assignments from the RBAC system, and records the completion of  $OP_{A_{2,2}}(WPR)$  for  $WPR$ .  $P4EW/R(WPR)$  sleeps.

**Second member** of the project team chooses the added menu entry for  $OP_{A_{2,1}}(WPR)$  in a session, reviews the purchase request for the widget and signs. This action causes  $OP_{A_{2,1}}(WPR)$  to send a successful completion message to  $P4EW/R(WPR)$ .

$P4EW/R(WPR)$  is awakened by the receipt of the message indicating the successful completion of  $OP_{A_{2,1}}(WPR)$ , removes role  $ROLE_{A_{2,1}}$  and its associated assignments from the RBAC system, and records the completion of  $OP_{A_{2,1}}(WPR)$  for  $WPR$ . Since both activities of segment  $S_2$  have now successfully completed,  $P4EW/R(WPR)$  creates role  $ROLE_{S_3}$ , assigns permission to perform  $OP_{A_{3,1}}(WPR)$  to role  $ROLE_{S_3}$ , assigns role  $ROLE_{S_3}$  to the project manager of the project team, adds an entry to select  $OP_{A_{3,1}}(WPR)$  to the project

manager's menu, and sleeps. When the project manager selects  $OP_{A_{3,1}}(WPR)$  from the menu,  $OP_{A_{3,1}}(WPR)$  is activated.

**Project manager** of the project team chooses the added menu entry for  $OP_{A_{3,1}}(WPR)$  in a session, reviews the purchase request for the widget and signs. This action causes  $OP_{A_{3,1}}(WPR)$  to send a successful completion message to  $P4EW/R(WPR)$ .

$P4EW/R(WPR)$  is awakened by the receipt of the message indicating the successful completion of  $OP_{A_{3,1}}(WPR)$  and removes  $ROLE_{S_3}$ 's associated assignments,  $P4EW/R(WPR)$  now assigns permission to perform  $OP_{A_{3,2}}(WPR)$  to role  $ROLE_{S_3}$ , assigns role  $ROLE_{S_3}$  to the division manager, adds an entry to select  $OP_{A_{3,2}}(WPR)$  to the division manager's menu, and sleeps. When the division manager selects  $OP_{A_{3,2}}(WPR)$  from the menu,  $OP_{A_{3,2}}(WPR)$  is activated.

**Division manager** chooses the added menu entry  $OP_{A_{3,2}}(WPR)$  in a session, reviews the purchase request for the widget and signs. This action causes  $OP_{A_{3,2}}(WPR)$  to send a successful completion message to  $P4EW/R(WPR)$ .

$P4EW/R(WPR)$  is awakened. As a result of receiving the successful completion message from  $OP_{A_{3,2}}(WPR)$ , it removes role  $ROLE_{S_3}$  and its associated assignments from the RBAC system.

The successful completion of workflow  $WPR$  causes  $P4EW/R(WPR)$  to terminate and be removed.

## References

- [1] R. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role Based Access Control Models. *IEEE Computer*, 29(2), February 1996.
- [2] Workflow Management Coalition Specification: Terminology and Glossary. WFMC-TC-1011 2.0, Workflow Management Coalition, June 1996. World Wide Web URL - <http://www.aiai.ed.ac.uk/WfMC/DOCS/glossary/glossary.html>.