Sysgem AG
Casa Bergenia
Postfach 159
CH-7031 Laax
Switzerland

Tel: +41 (0) 81 921 6853

office@sysgem.com
www.sysgem.com

Functional Design Specification

Sysgem Roles Based Access Control

Authorization sheet

Authors			
Function	**name**	**date**	**signature**
Project Manager	M.K. Schofield		
Design Leader	B.A.L. Jemmett		

Approvals			
Function	**name**	**Date**	**signature**
Customer – Technical Approval	C. Pijnenburg		
Customer – ISD/CS	W. Sonnema		
Customer – Security Manager	W. Rovers		
Customer – Project Manager	R. Peeze Binkhorst		

Reviewers			
Function	**name**	**Date**	**signature**
Sysgem Development	S.J. Brown		
Raxco – Quality Assurance	A. Kuipers		
Organon – Technical Review	V. Weijer		

Authorization			
Function	**name**	**Date**	**signature**
Sysgem Managing Director	J. Marilus		

Contents

Document history

Revision	Name author(s)	Revision description	Revision date
1.0	M Schofield / B Jemmett	Initial version of the document	24 March 2006
1.1	M Schofield / B Jemmett	Draft – Internal Review	09-May-2006
1.2	M Schofield	1st Review Draft (Organon)	12-May-2006

1 Purpose

The purpose of this document is to give a description of the functional design of the Sysgem Role Based Access Control (RBAC) system.

It has been written in response to a set of documents that were produced by NV Organon between January 2005 and December 2005. (See paragraph: Related documents below).

Sysgem RBAC is a product that will be put into 'general availability' to licensed customers of Sysgem AG. While the first target customer will be Organon NV, the functionality is not restricted to this one customer.

RBAC will be used in Organon to automate and simplify the procedure of requesting, authorizing and providing users with accounts and access to applications, standard software and shares.

2 Scope

2.1 RBAC implemented as Web Application

RBAC will potentially be used by all employees of Organon (worldwide). The system is designed so that all employees and other users of the Organon internal IT infrastructure will be able to request roles for themselves.

IT Coordinators (ITCs) use the product to manage user access for the users in the departments for which they are responsible. ITCs have the prime responsibility for approving requests for new membership of Departmental Roles for the departments for which they are responsible.

Application (Resource) Managers are primarily responsible for approving requests for access to the applications/resources for which they are responsible. They may delegate some of this authority to the ITCs.

The User Interface (to request roles; to approve role requests; to enquire on the status of a request and to enter a manual updates to the RBAC database) will all run from a Web Browser window. A Web Server Application will be installed on a Windows Server of the customer's choice and the UI will be invoked by following a URL link to RBAC on that server.

Mocked-up examples of the user interface can be found in later sections in this document.

2.2 RBAC implemented for access to resources in Oss

Organon intends to use Sysgem RBAC, initially, in the locality of Oss (Holland), but eventually intend to manage access to resources on other sites. For this reason, a single instance of an installed RBAC system is designed from the outset to interface to multiple instances of SAcM and multiple instances of SAcM WAR DB & Subscriber DB, in as easy a way as is practical.

2.3 Users of RBAC

Organon intends RBAC to be used not only by internal Organon employees, but also by contactors and other external and remote users who have access to the Organon IT infrastructure. For this reason the layout of the UI and the clarity of help and other descriptive text will be reviewed as the project progresses. Software baselevels will be made available to Organon to install and run on test systems for the purposes of reviewing the UI.

2.4 Resources in RBAC

RBAC is used to manage requests for access to resources and also to manage that access. The resources managed with RBAC in Organon are:

- Shares - access currently requested by Organon C-form
- Applications (including RBAC) - access currently requested by Organon C-form
- Standard software - access currently requested by Organon P-form
- AD accounts - currently requested by Organon B-form.
- Other requests that need to be satisfied manually.

The first functional baselevel release of Sysgem RBAC to Organon will support the requests for resources that need to be satisfied by manual intervention. Such a request might be, for example, for the supply of a piece of hardware (e.g. a workstation). The request has to be approved and once approved, a notification is sent to the appropriate person(s) who will take manual action on the request. Once the request has been satisfied the database will be manually updated to record that the action is complete. If Organon chooses to use this feature for evaluating the control of hardware requests, then this is the nature of this baselevel release.

2.5 Related documents

This (Sysgem) Functional Design Specification refers mainly to the document: ISD-OYT-FDS-01 (Rev 1.4) written by Kees Pijnenburg (Organon) and dated 13-Dec-2005.

ISD-OYT-FDS-01 in turn refers to the following documents, and is a culmination of the requirements and the intent defined in:

- ISD-OYT-Vision-RoleManagement-01.doc (Vision document)
- ISD-OYT-URS-RoleManagement-01 (User Requirements Specification)
- Output from RBAC prototype workshops.

ISD-OYT-FDS-01 is the most recent document produced by Organon and summarizes all requirements for this product found in all other documents.

Sysgem have taken ISD-OYT-FDS-01 to be a statement of the Organon requirements even though the original intention of this documents was to be an internal Organon Functional Design Specification, had the project been developed internally by Organon software engineers.

2.6 Document Structure

This Sysgem Functional Design Specification mimics the document structure of ISD-OYT-FDS-01 exactly. Later documents produced in the same series by Sysgem and Raxco Software (Netherlands), such as Test Specifications, and Test Results will also follow exactly the same structure.

The intention of using the same document structure is to ensure 100% consistency with the requirements, and 100% coverage of functionality and test specifications with the original (ISD-OYT-FDS-01) requirements document.

3 Abbreviations

Abbr.	Description
APS	Application Support
CMDB	Configuration Management Data Base
FDS	Functional Design Specification
HRM	Human Resource Management
ISD	Information Services Department
ITC	IT Coordinator / IT Contact person / Department representative
POC	Proof Of Concept
RBAC	Role Based Access Control
SAcM	Sysgem Account Manager
SDS	System Design Specification
SOP	Standing Operating Procedures
UI	User Interface
URS	User Requirements Specification
WAR	Web Access Request (Database of SAcM)
WTR	Werk Tijden Registratie (Organon's User Registration for External Employees)

4 Definitions

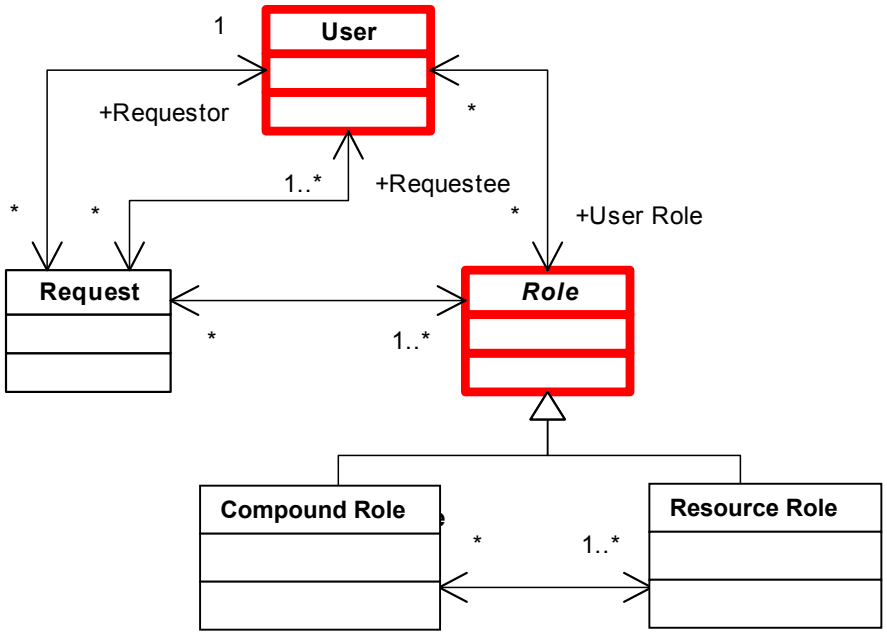
Keyword	Description
Application Manager	A person who is a manager of a specific resource. It would be logical and more consistent with other definitions to call this person a Resource Manager, but Application Manager is commonly used within Organon.
Approver	An authorizer who approved a specific request.
CMDB	CMDB is the database of the software package *Asset Center* of vendor Peregrine. In this application all assets of ISD and assets supported by ISD for the business are maintained. The CMDB maintains assets such as: shares, software applications, standard software, desktops and laptops.
Compound roles	"Roles" can be either "Resource" Roles or "Compound" Roles. Compound Roles may contain multiple Resource Roles and other levels of Compound Roles. (See also "Resource Roles"). However, In the case of Organon, a "Compound Role" will be related to a function in just one organizational department. Examples are: ISD Developer, ISD Project Leader.
Conflicting roles	Two resource roles can be conflicting or mutual exclusive, which means that the resource does not allow a user to have both roles at the same time.
External user	A person that is not an employee of Organon, but is hired by Organon to work on the local site.
Internal user	An employee of Organon that works on the local site.
Local site	Oss, Schaijk and Apeldoorn.
Manual execution	The action that needs to be physically carried out to complete a request of this type. (See also Manual Update and Manual Updater).
Manual update	The process of updating the RBAC database after a request has been manually (physically) satisfied.
Manual updater	A person (mostly the technical application manager) who gets (via a mail) notification of a request for e.g. the creation of an application account, when that application type is not yet supported by SAcM. So, the request has to be executed manually on that application.
Remote site	All sites of Organon with exception of the local sites. For example: Roseland.
Remote user	An employee of Organon that works on a remote site.
Request	A request consists of a list of one or more department- or resource-roles and a list of one or more requestees. A request has one requestor.
Request line	A request line consists of a single department role or a single resource role and a single requestee. When the request is

Keyword	Description
	committed, a request line is created for each and every combination of requestee and role in that request. A reference to the originating request is contained in every request line.
Requestee	The user for whom a request was made. So a user is requestee in relation to a specific request.
Requestor	The user that entered a requested. So a user is requestor in relation to a specific request.
Resource	Application, share, standard software or Windows AD account. (In a later phase, a piece of hardware could also be seen as a resource)
Resource Manager	See "Application Manager"
Resource role	A role or group within a resource. A user can (request to) have access to a resource role. Examples are: Web applications that use local NT groups for security (e.g. Mars with role / group G001), Shares with Groups (e.g. the share FRAG$01D with role / group FRAG&G001), Oracle Applications that use Oracle roles for security, being a member of an AD domain groups for Standard Software distribution.
Role	A department- or resource role.
SAcM	User Management Sysgem in use by User Management team (ISD/CS) which is used to process the Organon B- and C-forms provided by the business (requests), such as access to applications and shares. RBAC will replace these forms; requests for access will be entered and authorized electronically. Subsequently SAcM will automatically process the request.
Sector	Sector within Organon, for example "Research", "Finance, IT & Communications" etc.
Site	Geographic location of a specific department of Organon. E.g. Oss or Roseland.
Standard Software	Standard software is all kind of software that is scripted (either requested by the business or ISD itself) by ISD for software distribution on desktops and laptops. The software will be distributed if requested by an Organon P-form. For example Acrobat Reader.
Technical grant	A resource role contains one or more technical actions, like put a user in an AD group, give a user an Oracle role, give a user an account, etc. Execution of these actions leads to technical grants for that user.
Technical role	See technical grant.
User role	The actual membership of specific users to department- and resource roles. So a specific user can have the role G001 in Mars and the role FRAG&G001 on share FRAG$01D.

Keyword	Description
User type	For security reasons the RBAC users are given types like RBAC manager, Application Manager, ITC, End User, Default User etc.

4.1 Correlation between definitions

The definitions of some keywords are interrelated. The following schema helps to clarify this correlation:



Note the following:

- A Request has one Requestor (who is a user)
- A Request has one or more Requestees (who are users)
- A Request has one or more Requested Roles.
- A Requested Role is always a Compound Role or a Resource Role. (In Organon, the Compound Role will reflect the Department name.)
- A Compound Role has one or more Resource Roles and may also include lower level Compound Roles
- When a user is granted a Compound Role or a Resource Role he is linked / associated to that Role. This link is called a User Role.
- Organon's "Department Roles" will be implemented as RBAC "Compound Roles".
- Organon's "Application Roles" and "Resource Roles" will be implemented as RBAC "Resource Roles".

5 Overview

5.1 Functional aspects

Within this document the following functional aspects are described:

- Requesting
- Authorizing
- Revoking
- Metadata management
- E-mail notification
- Processing
- Auditing
- Security
- Tracking
- Reporting
- Reconciling

These aspects are described in chapter 6.

The data model, its entities, attributes and relationships are described in chapter 7.

The following interfaces are described:

- AD interface
- Role interfaces
- SAcM interface
- Other Interfaces (e.g. CMDB / Asset Center interface)

These interfaces are described in chapter 8.

5.2 Prerequisites

All users of Organon need to have a Windows AD account (on the EMEA domain) to be able to use the IT infrastructure of Organon.

The first step in the use of RBAC for a new employee takes place automatically as a record is placed in the Subscriber DB for this user during the automated SAcM Subscriber Synchronization process. After this new record appears in the Subscriber DB, a new entry is then automatically created in the RBAC “tblUsers” table by the RBAC Reconciliation process.

The next step is invoked by an Organon user (a “Requestor”) who uses RBAC to request that the new User be granted access to a Departmental Role. The Departmental Role (e.g. “ISD Project Leader”) will include a request to add a Windows AD account. This action will replace the submission of an Organon “B-form”.

5.3 Screenshots

This specification includes ‘mock-ups’ of the screens that will be used in RBAC.

The layout and controls of these screens are not fixed and may change during implementation or as a result of development and feedback from Organon.

The screenshots should be viewed with this in mind.

5.4 Traceability matrix

A traceability matrix was included in the document: ISD-OYT-FDS-01 to cross reference sections with the User Requirements Specification and with other documents that provided input to ISD-OYT-FDS-01.

To ensure 100% consistency, and as a traceability matrix between this Sysgem specification and the primary document that was used as input (ISD-OYT-FDS-01), the document paragraph structure of this Sysgem document has been made identical to the Organon ISD-OYT-FDS-01 document. To make a cross reference between the two documents, simply open the two documents on the same paragraph number and the content of the corresponding paragraphs will refer to the same topic.

The original reference codes to the Organon Requirements defined in ISD-OYT-FDS-01 has been retained in this document and is included in square brackets “[“ & “]” throughout the document.

This specification is largely based around those original set of requirements. The summary text of each requirement is included at the start of each section in this Sysgem document and for clarity has been bolded and preceded by the word “**Requirement**:”.

The same approach is also taken with the Sysgem Test Plan and Test Results documents.

6 Function descriptions

6.1 Requests

Requirement: [1.10] Request has one requestor:

Each request has one requestor (the user who enters the request in RBAC). Request entries are linked back to the originating requestor by the field: "Requesting User ID" in "tblRequestHeader".

Requirement: [1.20] Request has requestees and roles:

In RBAC a request consists of a list of one or more users (also referred to as 'requestees') and a list of one or more roles. Roles are of two types:

- Resource roles.
- Compound roles (e.g. named with a 'departmental function'). These consist of resource roles and further levels of compound roles.

6.1.1 New request

Requirement: [1.30] GUI of "new request" uses tabs

See the screenshot (RBAC-mock-up-1b.jpg) on the following page for a mock-up of the user interface style for making an RBAC request.

The tab "Request" is used for entering 'New Requests'. Within this tabbed page there are three panes separated by the following lower level tabs:

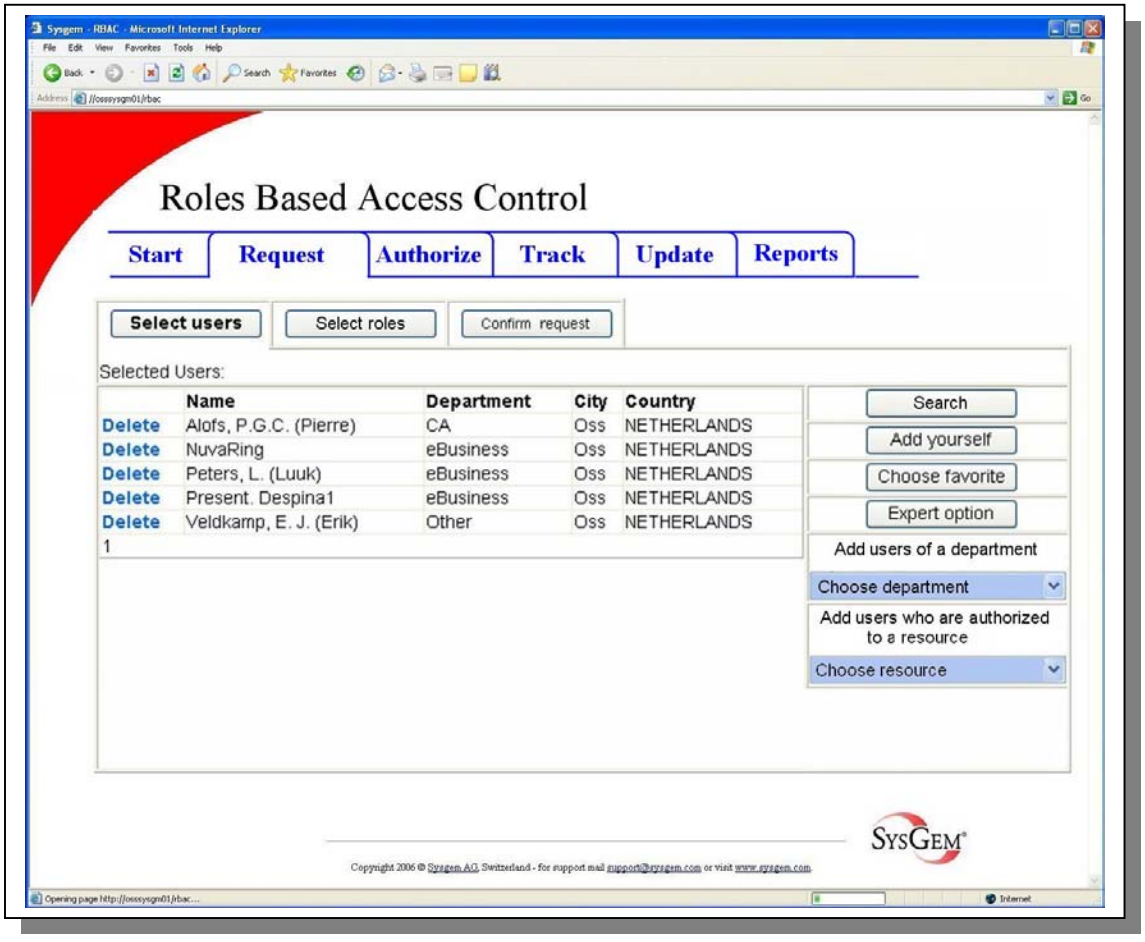
- Select Users
  (i.e. requestees)
- Select Roles
  (to grant to the requestees)
- Confirm request.

6.1.1.1 Tab: Select Users

Requirement: [1.40] add (or delete) requestees to request

The purpose of this pane is:

- to give more information about the users (requestees) that are already selected for a request
- to add or delete users (requestees) easily to the list.



RBAC-mock-up-1b.jpg

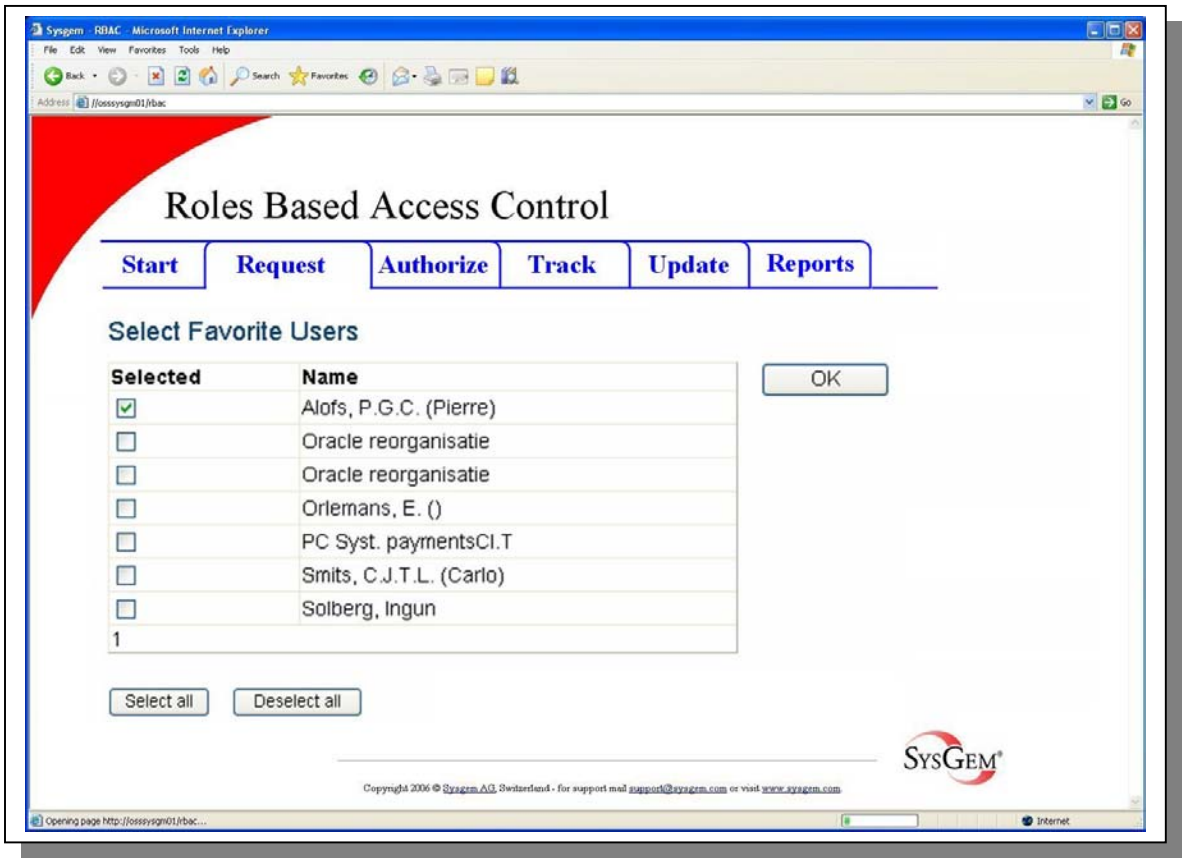
Requirement: [1.50] Add yourself to a request

The button “Add yourself” adds the requestor (authenticated by its Windows account) to the list.

Select Users by Favorites:

Requirement: [1.60] Add (or delete) favorite requestees to request

The button “Choose favorite” shows a list of the ‘favourite users’ of the requestor.



RBAC-mock-up-2.jpg

Items from the favourite users list are selected and easily added to the selection list for this request after pressing the OK button on this screen.

See requirement [1.100] below for creating a personal list of favorites.

Users that are already part of the request are shown as selected when the favourite list is displayed. New selections will be added to the list when the OK button is pressed. Also any existing users that have been deselected from the favourite list will be removed after the OK button is pressed.

The buttons “Select all” and “Deselect all” can be used to select or deselect all Favorite users to / from a request list.

Select Users by Expert Option:

Requirement: [1.80] Expert option to add (or delete) requestees to a request

The button “Expert Option” of tab “Select Users” provides the requestor with copy / paste functionality. This function processes partial text and tries to recognise users whose names start with the same characters as the pasted text. It then adds the full text to the list of users of the request.

Select Users by Department:

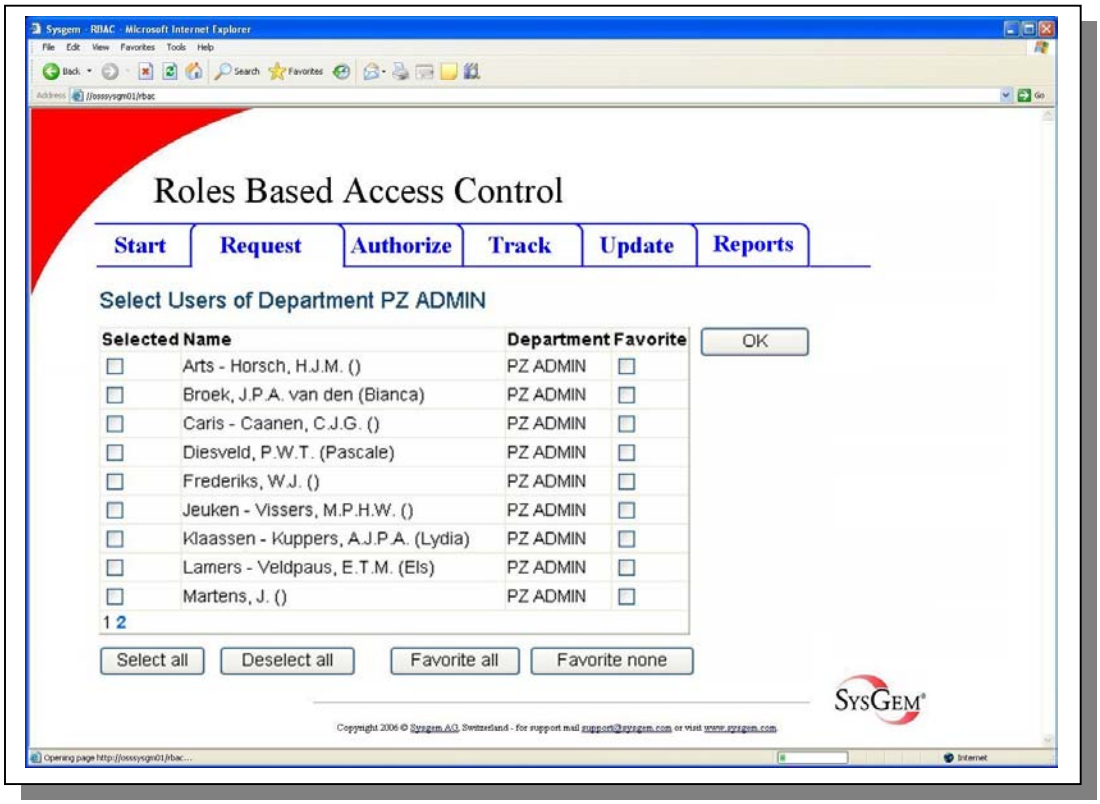
Requirement: [1.90] When adding requestees to a request, filter by department

In the case of IT Coordinators, the dropdown list: “Choose department” provides the requestor with a list of all the departments for which they are IT Coordinators plus the department to which the requestors themselves belong.

Table A2 (tblAuthGroup) and A3 (tblAuthGroupMember) will be used to determine this list for the requestor. The departments for which the requestor has RBAC administrative rights (i.e. the AuthGroup has “IsAdminGroup” flag set to “True”) will be used.

It is also possible to drill down further to get a list of *all* departments in the system.

After choosing a department the following form is shown giving a list of all users in the selected department:



RBAC-mock-up-3.jpg

Requirement: [1.100] Create personal list of favorites

During the process of creating a request, the checkbox in the column “Favorite” can be used to add users to the favorite users list of the requestor.

Personal favorites can also be managed on the main tab “Update” of the RBAC form.

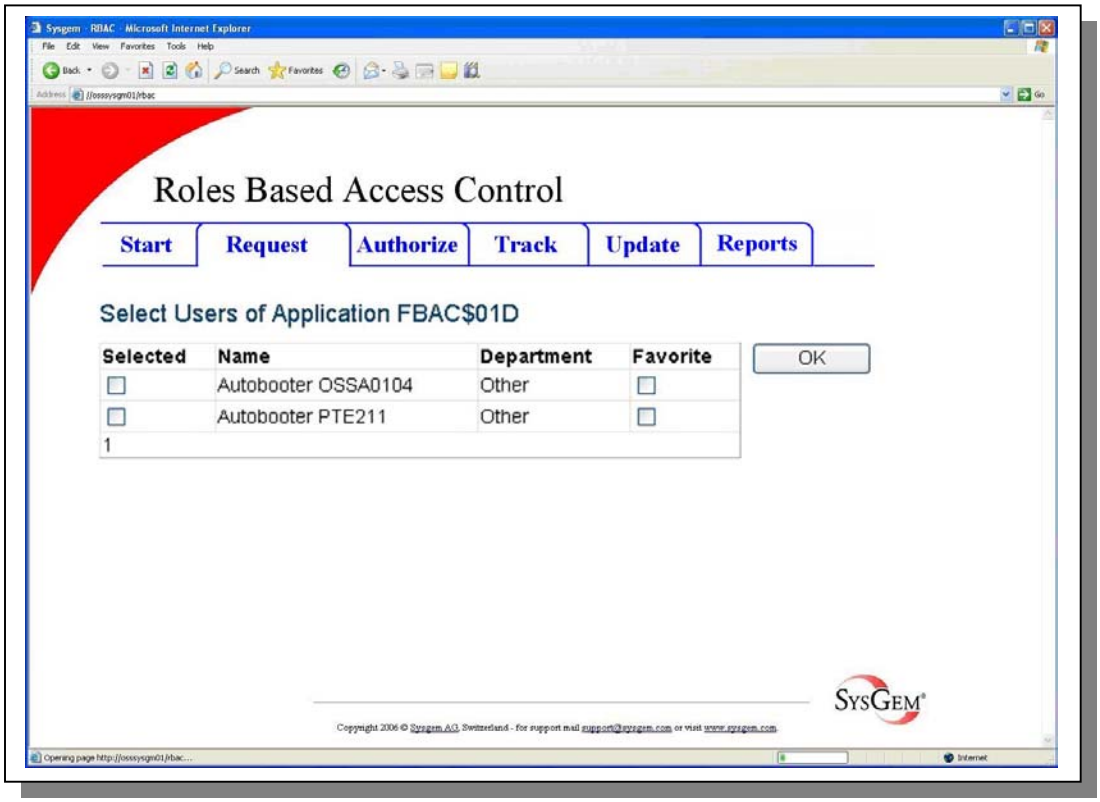
Select Users by Application:

Requirement: [1.110] When adding requestees to request, filter by application

If the requestor is a Resource (Application) Manager, then the “Choose resource” dropdown list on the “Select Users” tab provides a list of all resources (applications) for which the requestor is a Resource (Application) Manager.

If it is required, it is then possible to drill down further and get a list of all resources in the system.

After choosing a resource the following form is shown. It has a similar functionality to the previous form (RBAC_mock-up-3.jpg), and is used to add further users to the request.



RBAC-mock-up-4.jpg

Select Users by Search Option:

Requirement: [1.120] When adding requestees to a request, filter by search option

The button “Search” searches for a user on a partial match of the username. We will also look into the possibility of having a search field on a partial match of a user’s fullname that is held in the Subscriber record.

A list of matching users is then shown with similar functionality as that shown in (RBAC_mock-up-3.jpg), and described for [1.90].

Requirement: [1.130] When adding requestees to a request, filter by share option

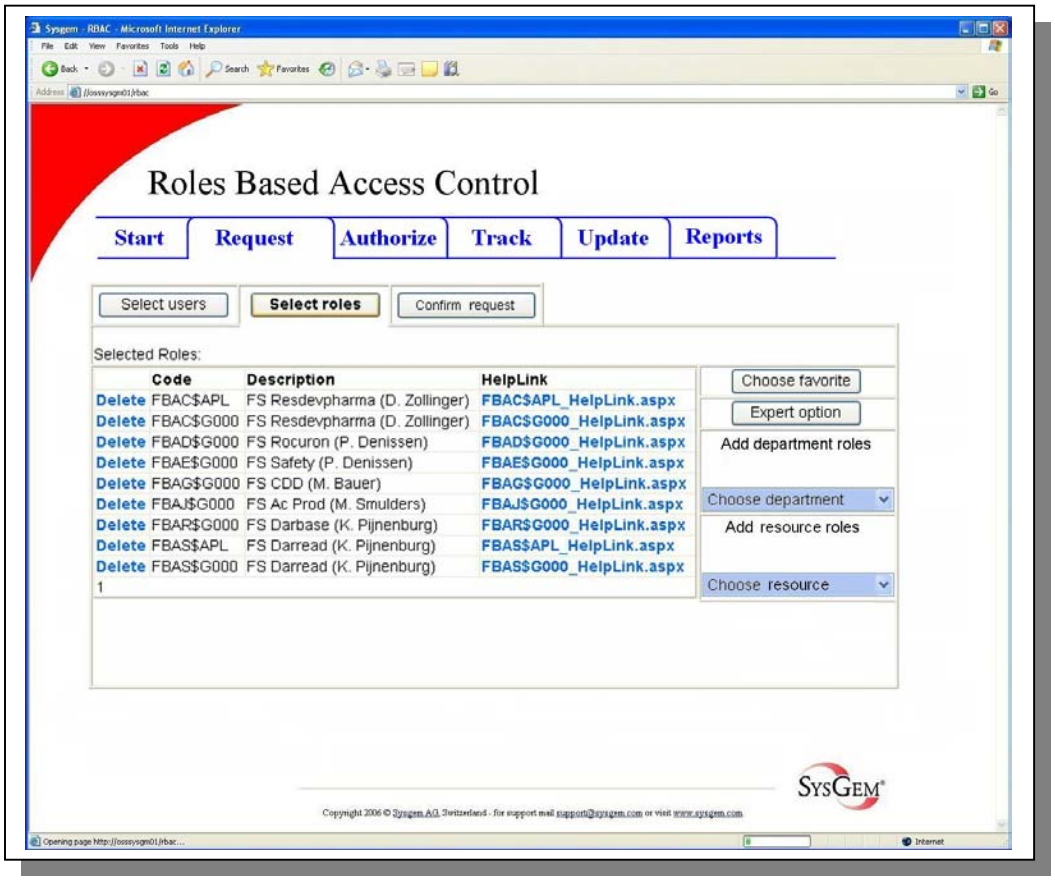
Following a discussion between Mike Schofield (Sysgem) and Kees Pijnenburg (Organon) on 05-Apr-2006 it was decided not to include this feature. Instead a dropdown option giving a list of all resource roles will be provided (combining the requirements of [1.110] and [1.130]).

6.1.1.2 Tab: Select Roles

Requirement: [1.140] Add (or delete) roles on request

The purpose of this page is:

- to give more information about the roles which are already selected, and
- to easily add (or delete) roles to / from the list.



RBAC-mock-up-5a.jpg

The functionality of adding roles is almost the same as that for adding users.

“Choose department” and “Choose role” dropdown lists gives lists of departments or roles specific to that IT Coordinator or Resource Manager, and further lists can be obtained by drilling down further into each list.

Select Roles by Favorites:

Requirement: [1.160] Add (or delete) favorite roles to request

The button “Choose favorite” shows a list of the favorite roles of the requestor. This is handled in a similar way to the favorite user list described earlier.

Select Roles by Expert Option:

Requirement: [1.170] Expert option to add (or delete) roles to request

The button “Expert Option” of tab “Select Roles” provides the requestor with copy / paste functionality. This function processes partial text and tries to recognize roles whose descriptions start with the same characters as the pasted text. Selected roles are then added to the list of roles of the request.

Select Roles by Department:

Requirement: [1.180] When adding roles to request, filter by department

The “Choose department” dropdown list provides the requestor with a list of departments for which he/she is the IT Coordinator, plus the department to which he/she is a member.

Drilling down in this form allows all departments to be available for selection.

After choosing a department, a list of related department roles is shown from which the requestor can select multiple entries.

Select Roles by Application:

Requirement: **[1.190] When adding roles to request, filter by application**

In the case of a Resource (Application) Manager, the “Choose resource” dropdown list provides the requestor with a list of all resource roles linked to this requestor. Drilling down in this form allows all resource roles in the system to be displayed, from which multiple items may be selected and added to the request.

Requirement: [1.200] Adding roles to request, filter by share

Following a discussion between Mike Schofield (Sysgem) and Kees Pijnenburg (Organon) on 05-Apr-2006 it was decided not to include this feature. Instead a dropdown option giving a list of all resource roles will be provided (combining the requirements of [1.190] and [1.200]).

Hyperlink to Help Text per Resource Role:

Requirement: [1.205] Feedback to the user about the resource role

The column “HelpLink” provides the requestor with a list of hyperlinks to the pages with additional information about the resource roles.

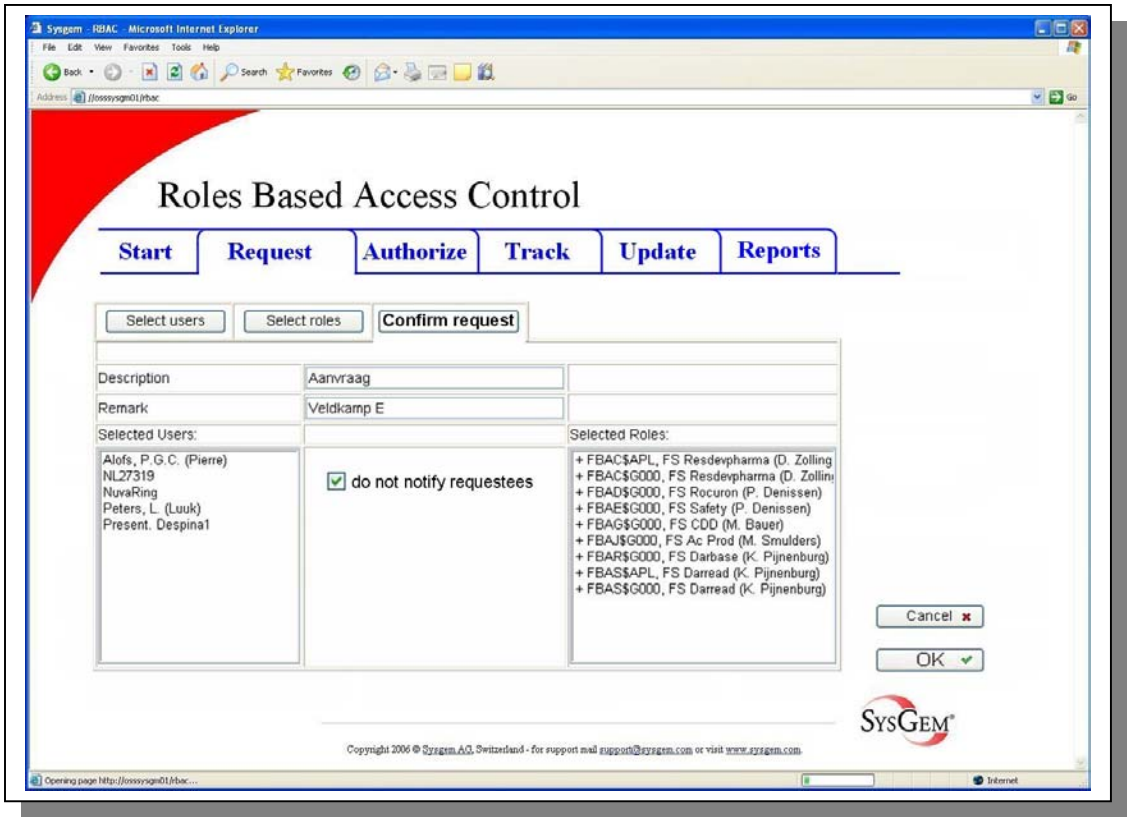
If a role does not have a hyperlink available then an empty field will be shown instead.

6.1.1.3 Tab: Confirm Request (Previously known as: “Overview”)

Requirement: [1.210] Overview of request

The purpose of this function is to enable the requestor:

- to be shown an overview of the entered requests
- to add a description and/or a remark to the request
- to confirm the request.



RBAC-mock-up-6a.jpg

(This tab has been renamed from: “Overview” to: “Confirm request”).

To confirm the request – press the button “OK”.

To cancel the request without saving any changes, press the “Cancel” button.

6.1.1.4 Confirm request

Requirement: [1.220] Confirmed request cannot be changed

Press “OK” on the “Confirm request” page to confirm the request. After confirming the request, the requestor cannot make any further changes, but it may later be rescinded (see also [1.245] and [2.80]).

Requirement: [1.230] Confirmation of request

Once a request is confirmed, a request line is created for each combination of roles and requestees of that request. Each request line is associated with one (and only one) requestee and one (and only one) role. So the request in the previous screenshot for 5 users and 9 roles, results in 45 request lines.

Requirement: [1.240] Confirmed request must have at least one requestee & at least one role

To confirm a request at least one requestee and one role have to be selected and the “OK” button pressed.

Requirement: [1.245] Requests (confirmed and un-confirmed) can be cancelled

As an alternative to confirming a request, the requestor can also cancel a request by pressing the “Cancel” button.

After the request has been confirmed, it may still be “Rescinded” (withdrawn) up to the point that it has been (fully) approved. After it has been approved, it may no longer be rescinded.

6.1.1.5 Requesting access to RBAC with RBAC

Requirement: [1.250] RBAC can be used to request access to RBAC itself

The purpose of this function is to make it easy to control the level access given to RBAC users. RBAC roles are available that grant RBAC privileges so that a user falls into one of the categories defined in paragraph 6.8.2 (Authorization) and as requested in requirement [8.30].

Requests for RBAC roles are approved by an RBAC Manager.

So RBAC can be used to request access to RBAC itself.

6.1.1.6 Requesting Windows AD account

Requirement: [1.260] With RBAC only accounts for common AD users will be requested

Within the Organon network, different types of Windows AD accounts are supported. The following shows a list of some examples:

- Special
- Developer
- APS
- ITC
- Administrator
- Support
- Account (for common users)

RBAC will be used for creating the Windows accounts for common users (only).

Requirement: [1.290] RBAC will execute the Windows AD request before any other request

When a request is made for one or more users that do not yet have a Windows account, RBAC automatically creates the requests for the corresponding Windows accounts before the resource requests are submitted.

RBAC will make sure that the sequencing of requests is appropriate, for example, it will ensure that an account request is executed before the resource requests are executed (see paragraph 6.6.1).

Requirement: [1.300] Option: do not notify requestees

An option to suppress the notification of changes to requestees is an option on the "Confirm Request" screen. See the checkbox on the screen shot: (RBAC-mock-up-6a.jpg) above.

(See also paragraph 6.5 regarding the notification of requestees).

6.1.2 No search / update request functionality

Requirement: [1.310] Confirmed request cannot be changed

Since the implementation uses a Web Server architecture, all data entered on a page will be transmitted to the server whenever there is a page transition. A transaction will thus be accumulated page by page at the server until the "Confirm" – "OK" button has been pressed.

In version 1.0, we will allow a partially completed transaction to be restarted on a subsequent session. In fact this will be the default, so that if the previous "request-transaction" were not confirmed, then it will be re-presented to the user when they next start the form. To abandon the current (or the last partially completed) transaction, the "Cancel" button needs to be pressed, at which point all fields will be emptied to their initialized state.

In a subsequent version we will consider allowing multiple partially completed transactions to be saved and later selected and restarted or cancelled.

6.2 Authorizing

Requirement: [2.10] Confirmed request-lines are the objects of authorization

Confirmed request-lines are the objects that are authorized. A request-line is related to a requestor, a single requestee and a single role.

Roles are of two types:

- Resource roles, which are related to a single resource
- Compound roles, which are related to a department. They consist of resource roles and may consist of further levels of Compound Roles (although in Organon, this is unlikely to be the case).

Requirement: [2.20] Electronic signing needed to authorize (approve or reject) a request-line

A request-line can be approved or rejected. This is called authorization. To check whether the authorizer has authority to authorize (and the logged in user is the same person) he has to enter his password again (this is called electronic signing). It would not be practical for the authorizer to enter his password before authorizing each request-line. Therefore, after entering the password, there will be a configurable period of time that the authorizer can approve or reject requests without having to re-enter his password.

Having to re-enter the password is a configurable option within RBAC. Since Organon has specified this as a requirement, it will always be switched on for the Organon implementation.

Requirement: [2.30] ITC authorizes Department roles and Application Managers resource roles

In general, an IT Coordinator authorizes Departmental Compound roles and Application (Resource) Managers authorize resource roles. However, exceptions are possible (see paragraphs 6.2.3 and 6.2.4).

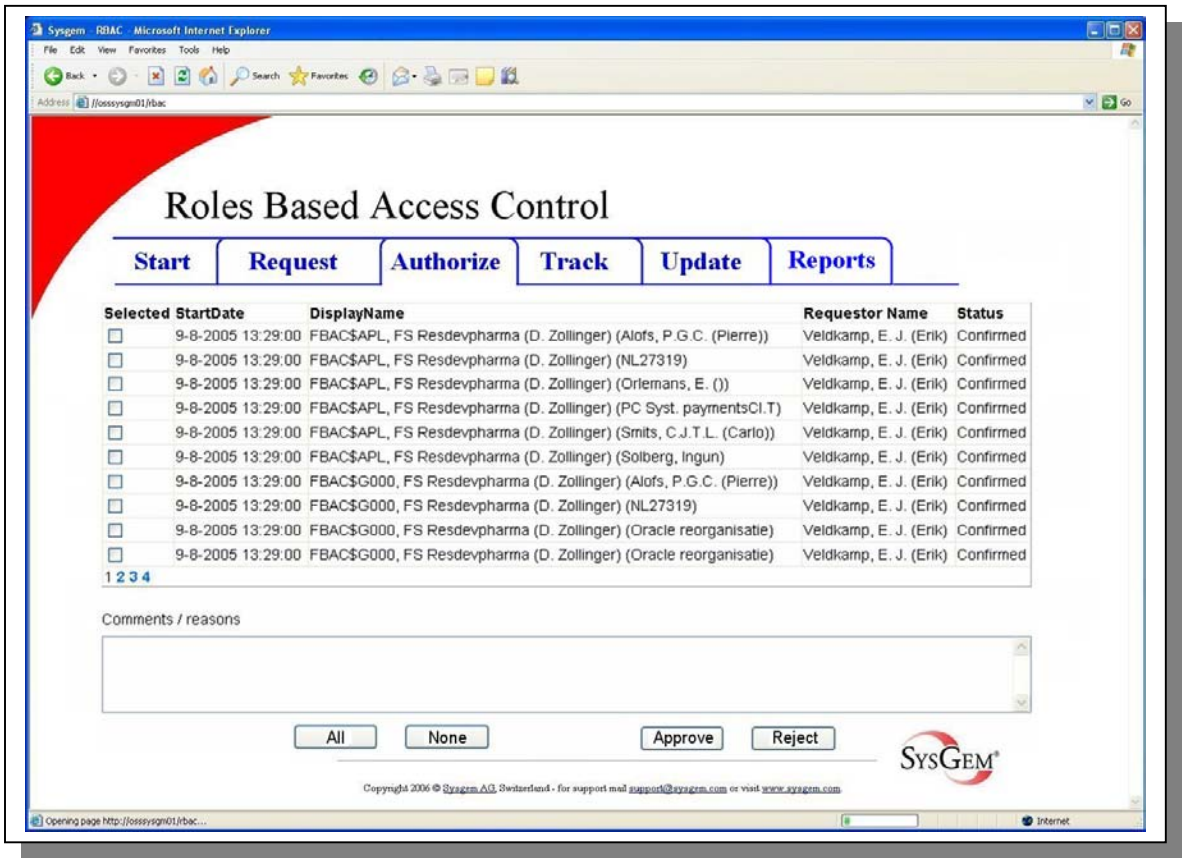
In addition, RBAC will allow (in exceptional circumstances) the combination of an RBAC Manager and a Security Officer to be able to authorize a request. For this, a special login form will be displayed with prompts for two RBAC Usernames and two passwords. The passwords will normally be the Windows login passwords for each user, unless the RBAC profile setting demands an RBAC specific password. After both users have successfully identified themselves as an RBAC Manager and Security Officer, they will be allowed to authorize any outstanding request.

Requirement: [2.40] Requestee may never be the same as either the manual executor or approver

For *Sarbanes and Oxley* purposes, an important general rule is that the requestee may never be the same as either the manual updater or approver. However, the manual updater and approver can be the same person.

Requirement: [2.50] Show request-lines that have to be authorized

The function “Authorize” displays a list with all request-lines that have to be authorized by the current user (i.e. the authorizer).



RBAC-mock-up-7.jpg

Requirement: [2.60] Enter comment when authorizing request-lines

The authorizer can select multiple request-lines and approve (or reject) them by choosing the button “Approve” (or “Reject”). After choosing one of these buttons, the entered comments are stored with all selected lines, and those entries are removed from the displayed list of items still to be approved.

A comment is mandatory when choosing the “Reject” option.

The Buttons “All” and “None” allow all checkboxes on the current displayed page to be selected or deselected.

Requirement: [2.62] when authorizing, mandatory comment for GxP or SoX compliant resources

When the resource is GxP or SoX compliant, entering comment is mandatory for both approval and rejection.

How do we know which resources are GxP or SoX compliant? Do we need to store something in the DB to flag this?

Requirement: [2.65] Add comment and attachment when authorizing a request

Besides the comments it will also be possible to add an attachment to the authorization of a request. This will for instance be used in case of an urgent authorization, to store a mail containing an official order to give a particular user access to a particular resource.

Requirement: [2.70] when authorizer is requestee, the authorizer cannot authorize the request

Note: when the requestee is the current user (the authorizer) the request will be visible in the list, but (because the authorizer and requestee cannot be the same person) the current user cannot approve it.

6.2.1 Authorize resource role

Requirement: [2.75] minimize requests and grants of “fancy applications”

Note: when the requested resource has its “AlsoAuthorizedByITC” flag set, it first has to be authorized by its ITC before any other authorization takes place (as if a department role had been requested).

State Transitions:

The following diagram shows the principal states of transactions in the RBAC and WAR DB:

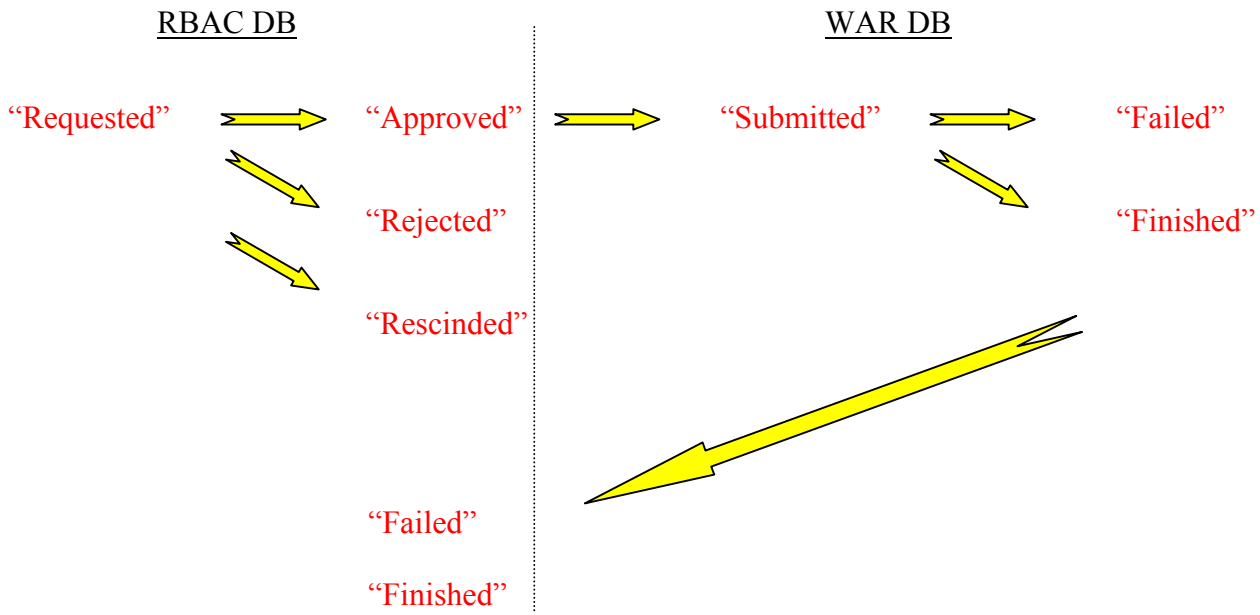


Fig: Main RBAC transaction states

The following diagram shows more detail of all the possible transaction states and their intermediate state transitions during the authorization of a resource role:

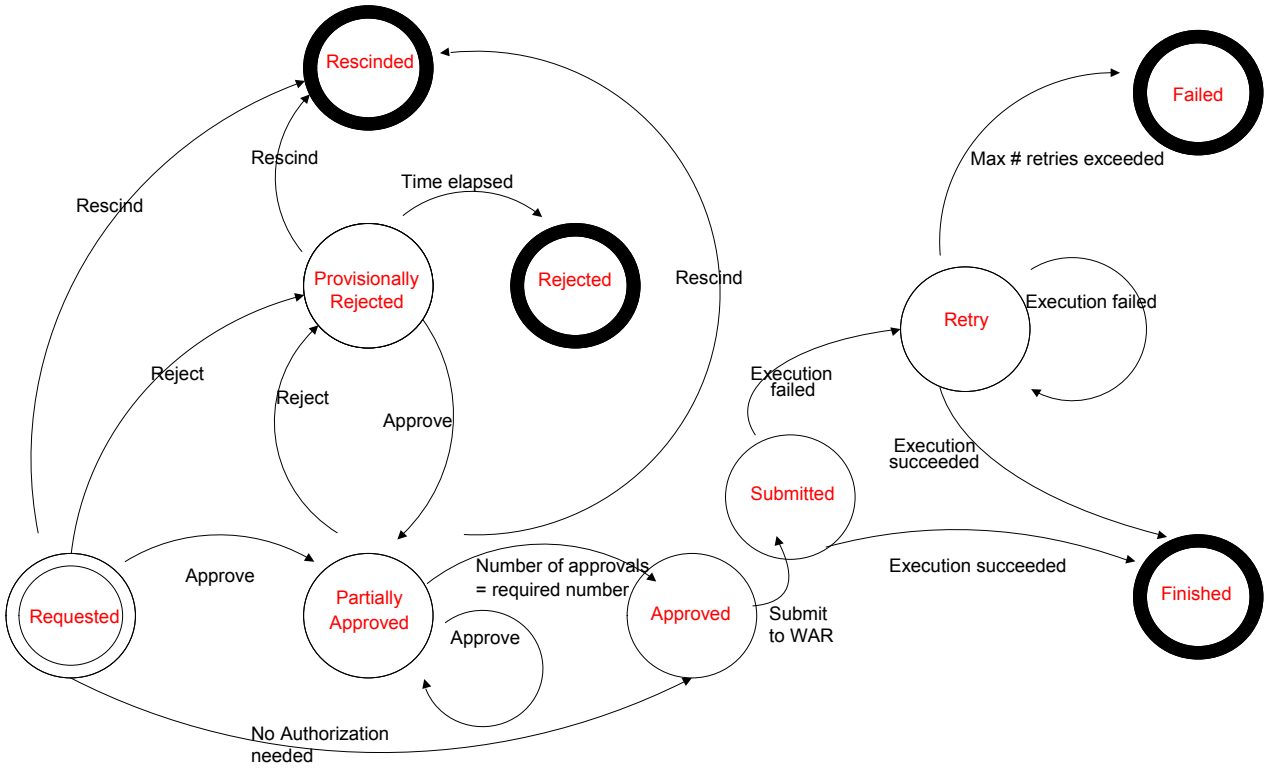


Fig: State Transitions – (Resource Roles)

Requirement: [2.80] Cancelling a request for a resource role

Once a resource role request line is created (after confirmation of its request) its start state is set as “Requested”.

If the requestor wishes to withdraw the request line after he has submitted it, but before it has been approved, he may do so by retrieving a list of all his outstanding requests; selecting the appropriate line(s) and pressing the “Rescind” button.

After the request line has been (fully) approved, it may not be rescinded.

Requirement: [2.100] Authorization of resource role can require multiple persons to approve

Notice that it is possible that one resource role has to be authorized by more than one person. In this case RBAC presents the request to all persons that are on the authorization list of the resource role. Rejection of one of the authorizers leads to rejection of the request line. See paragraph 6.4 for the definition of the authorization list. When all authorizers of the list have approved, the request status becomes “Approved”. Then, after it has been transferred to the WAR DB, the status of the request becomes: “Submitted”.

Requirement: [2.110] Rejected requests for resource roles can still be approved

Also notice that rejected requests can still be approved. This functionality is built in to make sure that the authorizer can approve the request finally, after he discussed the rejection with the requestee or requestor. This avoids the necessity to re-enter the request. Rejected request lines will be visible for the authorizers for a configured period of time (see paragraph 6.4.1). During this time the state is shown as “Provisionally Rejected”. When this period elapses the request line will show the state “Rejected”.

Requirement: [2.120] Some resources do not need authorization

For some resources (like Acrobat Reader) no authorization is needed at all. The “Approved” state is reached from the start state with no authorization action at all.

Requirement: [2.130] Request-lines that are “under execution” cannot be cancelled

While the request line is: “Requested”; “Provisionally Rejected” or “Partially Approved”, the requestor or requestee can still rescind the request. Once the request line has reached the stage of “Submitted” then it cannot be rescinded any more.

The transition of states after the “under execution” state is reached is explained in 6.6.1.

Requirement: [2.150] Only RBAC Managers can authorize request-lines for user types

Only RBAC Managers can authorize request-lines for RBAC user types (e.g. Application Manager, Manual Updater).

6.2.2 Authorize Department Role

The following bullet-points show when ‘department roles’ will be used and when ‘resource roles’ only will be used

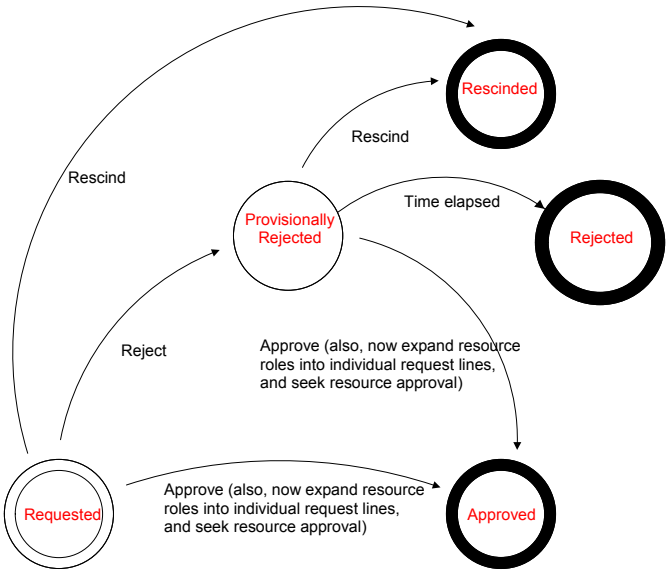
- Department roles are particularly important when a new user joins the company, or a user moves between departments. If a request has been made for a department role, then the approval for the department role membership must be given BEFORE approvals for the underlying resource roles are required or allowed.

- A user who is already a member of a department role (and not in the process of moving between departments) is more likely to have requests for additional resource roles than for new department roles.

Requirement: [2.160] Approved department roles are expanded in their resource roles

Department roles have to be approved before the authorization for the underlying resource roles is requested. As soon as an IT Coordinator approves a department compound role, then it is expanded into the request-lines for the underlying resource roles (and not before). These resource roles then have to be authorized separately by the appropriate Application (Resource) Managers.

The following diagram shows the possible states of the authorization of a department role and their related actions:



Requirement: [2.180] Requests for department roles can be cancelled

Once a department role request line is created (after confirmation of its request) its start state is shown as “Requested”.

If the requestor wishes to withdraw the request line after he has submitted it, but before it has been approved, he may do so by retrieving a list of all his outstanding requests; selecting the appropriate line(s) and pressing the “Rescind” button.

After the request line has been approved, it may not be rescinded.

Requirement: [2.190] Rejected requests for department roles can still be approved

From the start state ‘Requested’ the request line can be rejected or approved (or rescinded).

Requirement: [2.194] Authorization of department role is always done by a single person

A department role only needs **one** approval. A “Provisionally Rejected” request can later be approved up to the point that it is moved into the state of: “Rejected”.

Requirement: [2.197] Approval of a department role does not imply approval of all resource roles

Approval of a department role does not imply approval of all resource roles within that role (unless the ITC has been delegated this authority; see paragraph 6.2.3.2). Approval for the resource roles have to take place separately, after the approval of the department role has been given.

6.2.3 Authorization as bottleneck

Requirement: [2.200] Limit risk of delay caused by authorization process

Because it is done by humans, authorization is a potential bottleneck for the process of handling requests. Humans go on holiday, get sick, or forget things.

To minimize this risk:

- authorization groups,
- delegated authorization
- implicit authorization

…are used.

6.2.3.1 Authorization groups

Requirement: [2.210] Authorization groups provide backups for authorizers

Authorizers need one or more backups because they will not always be available. Multiple users in Authorization groups provide this backup.

6.2.3.1.1 Resource roles

Requirement: [2.215] Authorization groups are related to authorizers and resource roles

The authorizer and his backup(s) are placed in an authorization group; which will be related to a resource with zero or more resource roles. (See paragraph 6.2.3.1.)

Requirement: [2.217] Only one person of an authorization groups has to approve

All users in an Authorization group are allowed to authorize the requests for that particular role, but only one user of the group can / must approve (or reject) the request.

When a role is requested that requires authorization from a member of an authorization group, all members of the group are notified and the request will be in all members authorization screen. When one of the members authorizes the request it will disappear from the authorization screen of the other members, (when these screens are refreshed).

Requirement: [2.218] For a single request a person can only approve in one authorization group

When a requested role has to be authorized by more than one group, it is not permitted for one person to perform the authorization in more than one of these groups.

Requirement: [2.220] Maximum number of authorizers is configurable

The maximum number of authorizers in a group will be configurable by the RBAC manager (see paragraph 6.4.1).

The system will give a warning when active authorization groups contain less than two users.

Requirement: [2.225] Authorization group has one primary-member who can delegate authorization

An authorization group has one primary-member. Only he has the privileges to delegate authorization. (See paragraph 6.2.3.2)

6.2.3.1.2 Department roles

Requirement: [2.230] ITC's of a department are the authorization group for department roles

Each department will have one or more ITC's. These ITC's constitute the authorization group for the department roles that are related to that department.

6.2.3.2 Delegated authorization

Requirement: [2.240] ITC with delegated authorization also approves resource roles when approving a department role

Another possibility for an Application (Resource) Manager is to delegate his privileges for authorizations to an ITC. When the ITC approves a department role he automatically approves the resource roles within it for which he has delegated authority. Delegation is limited to the department roles of departments for which the user is an ITC (and hence, to users within those departments).

Requirement: [2.244] Two ways to delegate authorization

An Application (Resource) Manager can delegate authorization in the following ways:

- When the Application (Resource) Manager is head of the authorization group, his authorization screen will contain a 'Delegate' button (next to the 'Approve' and 'Reject' buttons) and when the requestor is an ITC. The delegate button allows the Application Manager to delegate authority to the requesting ITC.

- The resource role screen also contains a delegate button that displays all ITCs whose department roles contains this resource role. The Application Manager can select the ITC(s) to whom they want to delegate the authority.

Requirement: [2.248] Delegation is specific for a combination of ITC, department and resource role

Authorization is explicitly delegated to a specific ITC (and not to an authorization group), for a specific department (and not for all of the ITC's departments) and for a specific resource role (and not for an entire application).

6.2.3.3 Implicit authorization

Requirement: [2.250] Implicit authorization when requestor can authorize his own request

With implicit authorization the requestor also has authority to authorize his own request. So in this case separate authorization is not required.

Note: An electronic signature (see paragraph 6.4.1) is required from the requestor for implicit authorization.

Some examples:

- An Application (Resource) Manager requests access for a user, for resource roles that are related to his resources (for which he is responsible). Authorization is implicit. No further authorization is required.

- An ITC requests access for a user, for a department role that is related to his department, but he has no delegated authority for the resource roles within it. In this case, the resource roles still have to be authorized by the Application (Resource) Manager.

Requirement: [2.270] Implicit authorization not allowed when the requestee is the requestor.

The system will not allow implicit authorization for requests where the requestee is the requestor.

6.2.4 Multiple authorizations

Requirement: [2.280] Multiple authorization groups can be related to a resource role

Some resource roles need multiple authorizations, i.e. more than one person has to authorize the role. RBAC insists on this by adding multiple authorization groups to a resource role. The following two types of multiple authorizations are supported:

- Parallel (authorization notifications (see paragraph 6.5) are all sent at once)
- Sequential (the authorization notifications are sent sequentially)

A sequence number field is available in the group table to define in which order the groups should authorize the requests.

Requirement: [2.290] Authorizers can be a member of multiple authorization groups

It is possible to put authorizers in more than one group. But when using multiple authorizations, the approvers of the separate groups, must be different users.

6.3 Revoking

Authority for revoking access to roles is less stringent that that for granting access. The following summarizes a set of rules {proposed and agreed between Mike Schofield (Sysgem) and Kees Pijnenburg (Organon) on 25 April 2006}:

1. A Resource Manager can grant or revoke access to the roles for his resource(s).

2. A Department Manager can grant access to the department roles under his control and (when he has delegated authority) to the resource roles underlying those department roles.

3. A Department Manager can always revoke access to both the department roles under his control and the underlying resource roles without any special delegation from the Resource Managers.

4. Resource roles requiring multiple approvals can be revoked by any (single) member of the approval groups.

5. Resource Roles that have been assigned to a user directly (i.e. not as part of a department role) can be revoked by a Department Manager for anyone within the departments he is responsible for.

6. Department Managers can revoke roles even when they refer to Windows AD Accounts.

7. RBAC Managers have the ability to grant and revoke RBAC roles only (i.e. the roles that control privileged access to RBAC itself)

8. Security Managers have the Authority to enter revokes for any user and any role within RBAC.

9. There is no “Approval cycle” for a revoke.

Requirement: [3.10] Granted resource and department role can be revoked

When access to a resource or department role by a user is no longer needed or wanted, these roles can be revoked.

Requirement: [3.15] Revoking is not always done at the same level as authorization

An ITC has authority to revoke a role (either a department role or a resource role) for anyone within the department(s) for which he has ‘Department Manager’ authority.

Requirement: [3.20] ITC’s, Application Managers and RBAC managers can enter revokes

ITCs, Application Managers and RBAC managers have authority to enter revokes.

6.3.1 Entering revokes

The process of entering revokes differs for the different roles of the user.

6.3.1.1 Entering revokes by an ITC (and Security Manager)

Requirement: [3.40] When ITC enters revokes, filter by department of search for user

To enter revokes an ITC first chooses a user. The list of users from which he can select are only the users in the department(s) for which he has responsibility. He can filter / scroll the user list by picking a department from a dropdown list and / or entering a partial username or partial full name.

Requirement: [3.50] Revokes multiple roles of a user in one action

After the user selection, a list is displayed with all active (department and resource) roles of that user. The ITC can now select multiple roles and select the button “revoke roles”.

The user interface and features available to a Security manager is the same as for an ITC except that a Security Manager is not constrained by Departments or Roles. A Security manager has the authority to revoke any user in the RBAC database from any role.

6.3.1.2 Entering revokes by an Application Manager

Requirement: [3.60] When Application Manager enters revokes, filter by resource

To enter revokes an Application Manager first chooses one of his resources from a dropdown list. Now a list is displayed with all active resource roles. The Application Manager selects one resource role and gets a list with all users that are granted this resource role. Now he can select multiple users and select the button “Revoke roles”.

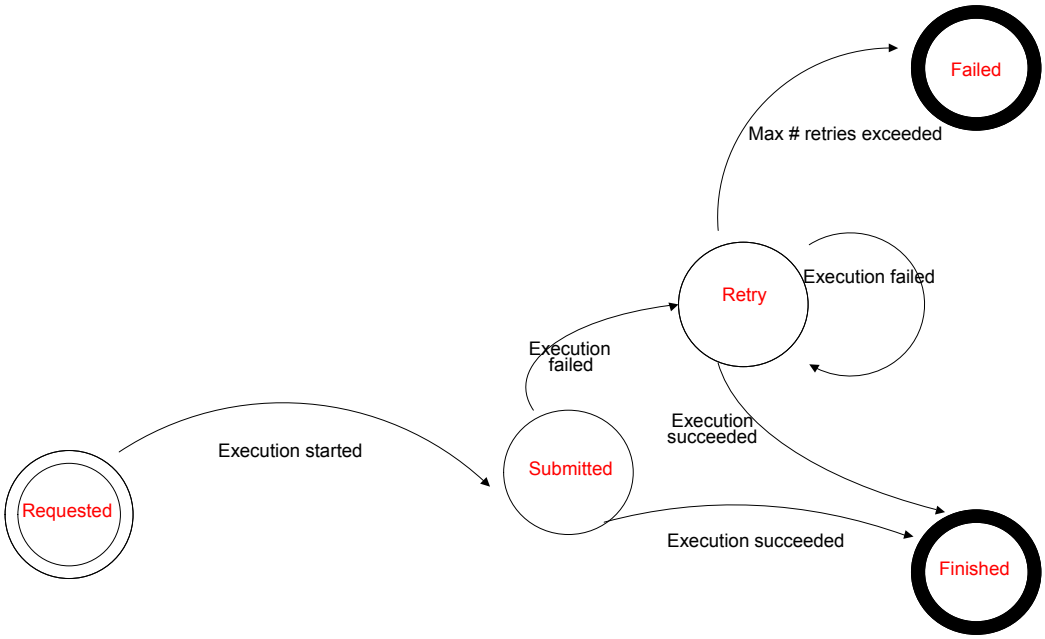
6.3.1.3 Entering RBAC role revokes by an RBAC Manager

Requirement: [3.70] Entering revokes by an RBAC Manager

RBAC roles are implemented as resource roles within RBAC and can be revoked by the RBAC Manager. He first chooses an RBAC role from his list of resource roles after which a list with all users of that role is displayed. Now he can select multiple users and select the button “revoke roles”.

6.3.2 Revoke process

The following schema describes the states of a revoke:



Requirement: [3.80] When revoking a department role a delta is calculated to determine which resource roles to revoke

When a revoke is “under execution” the actions for the underlying resource roles are determined and then executed via the WAR DB. When a revoke contains department roles the underling resource roles will only be revoked when they are not part of a different active department role of the user.

Requirement: [3.90] Revoke action of a resource role are the opposite of the grant actions

The actions associated with a revoke will be the inverse of the actions to grant access. e.g. place user in a group, becomes remove user from a group. Create an account becomes delete or deactivate an account etc.

Revoking a user from a role that removes access to an O/S Account only deactivates the account, it does not delete it.

The rest of the process of executing the revoke is the same as executing a request (see paragraph 6.6 for an explanation of the process and the states)

6.4 Metadata Management

Requirement: [4.10] Metadata can be managed

The RBAC Metadata that can be managed consists of the following data:

- Configuration
- Resources
- Resource roles
- Resource role actions
- Users
- Departments
- Department roles
- Authorization groups
- User roles

Sysgem will provide a generic script for importing bulk data into RBAC tables. This will primarily be used for the initial load of data. It will read data from a text file (in a defined format) and invoke the RBAC API to write that data to the RBAC DB tables. The text file identifies the table(s), the field(s) and the data for inclusion, so a pre-requisite of this tool is that the user fully understands the RBAC DB schema.

The bulk Import script will be written in such a way that it may be invoked from a SEM display or from the WEB UI.

Sysgem will also provide a Web based form where changes can be made interactively. As with the bulk import an understanding of the RBAC DB structure is required. The interactive form for modifying RBAC metadata is intended for entering small amounts of data, such as 'fine-tuning' of data that has been imported with the bulk Import feature.

Great care needs to be taken when using the bulk import tool and before it is used the text import file should first have been verified by testing it on a test system.

All metadata imports will be recorded in the RBAC Audit log.

Requirement: [4.20] End Users, Default Users and Manual Executors have no access to Metadata

Only RBAC users classified as "RBAC Metadata Managers" or "RBAC Superusers" have the authority to define RBAC Metadata.

See also the Security section on "Authorization" for access rights to RBAC features.

6.4.1 Configuration Data

Requirement: [4.30] Configuration of RBAC properties

The following are examples of configurable settings:

- The level (none, error, warning, information etc.) of debug information that has to be logged. The default will be none.
- SMTP server.
- Period of time that rejected request lines will be visible for authorization. After this period they are finally rejected. (See paragraph 6.2.1).
- Period of time after which an authorizer has to enter his password again. (See electronic signing in paragraph 6.2).
- Maximum number of users in an authorization group.
- Others – TBA, as implementation progresses.

Requirement: [4.40] Only RBAC Managers configure RBAC

This function is only accessible by “RBAC Managers” and “RBAC Superusers”.

6.4.2 Resources management

Requirement: [4.50] Types of resources (applications, shares, standard software)

A number of resource types are supported in RBAC (this list is not exhaustive):

- O/S accounts
- Home Directories
- O/S group membership
- E-mail mailboxes
- Share access
- Application accounts
- Application group membership

Requirement: [4.60] Resources are imported from CMDB
… and …
Requirement: [4.80] Shares are imported from CMDB
… and …
Requirement: [4.100] Information that is lacking in the CMDB
… and …
Requirement: [4.110] Date provided by CMDB will not be updated in RBAC

CMDB will be used by Organon to gather information for import into the RBAC metadata using the bulk import tool described earlier in Paragraph 6.4. RBAC itself will not have any knowledge of CMDB.

Requirement: [4.120] Only Application Managers and Metadata Managers have access to resource management

This function is only accessible for Application Managers and Metadata Managers.

6.4.3 Resource roles management

Requirement: [4.130] Information contained by a resource role

A resource role is related to a resource and contains the following information:

- Whether it can be delegated.
- Type of execution (manually or SAcM see paragraph 6.6).
- Overdue time span for sending reminders for authorizing a request and for manual execution of a request.
- Hierarchy group (non conflicting roles are put into different hierarchy groups, conflicting roles are put into the same hierarchy group).
- Level in hierarchy within hierarch group (for conflicting / mutual exclusive roles). See paragraph 6.6.4 for handling of conflicting roles.

Requirement: [4.140] Only Metadata Managers have access to resource role management

This function is only accessible for Metadata Managers.
.

6.4.4 Resource role actions management

Requirement: [4.150] Resource role actions are executed by SAcM to grant a resource role

The actions required to process a role via SAcM are the WAR commands supported by the SAcM WAR Process Requests custom display script. A resource role can have one or more action steps defined and the sequence in which multiple actions are executed is also stored in the action definition for each role.

Requirement: [4.160] Add (or delete) resource role actions

This function will allow the user to add actions to a resource role (using the bulk import or manual import mechanism) or delete actions from a resource role (only available to the manual import mechanism).

Requirement: [4.165] Resource role actions have a specific sequence in which they will be executed

It will also be possible to put the actions in a certain sequence (this will be the sequence in which SAcM will execute them) for each role. For example you need to have first an account on a specific server, subsequently you can get an account for the application (=resource) processing that server.

Examples of actions are:

- Put user x in Active Directory Group y.
- Put user x in Oracle Group y.
- Give user x access to share y.

Requirement: [4.180] Information of a resource role action

Actions will have the following attributes (which can be updated by the user):

- Description
- Type (the actual command to execute)
- Number of retries (in case of failure)
- Time between retries (in case of failure)
- List of Parameter <-> value pairs.
- Source of value (fixed, field x from user, or interactive)

The action “Put user x in Active Directory Group y” will have the following parameter-value pairs:

- User User ID of request-line
- Type of Group Active Directory
- Group Y

Requirement: [4.200] RBAC will compute the delta when actions are updated

After an update of the actions of a resource role RBAC will have to ask SAcM to execute the new actions for the current users of the resource role. To accomplish this RBAC will compute the delta of the old and new actions.

Requirement: [4.205] RBAC will not contain quantified parameters for creating an account

For accounts, the resource action parameters in RBAC will not contain quantified parameters (such as a quota parameters of an account). These parameters will be stored in the SAcM scripts.

Requirement: [4.210] Only Metadata Managers have access to resource role actions

This function is only accessible for Metadata Managers. Application managers can view this information.

Requirement: [4.215] For distributed applications the site of their actions can be overruled

All resources in RBAC are site specific, i.e. RBAC needs a specific target object to set on a specific target agent when a role is applied. If the setting of a single object (e.g. membership of an A/D group) makes a resource globally available on the Organon network, then this is transparent to RBAC. If there are roles requiring actions on multiple agents, then the role will have to be defined as a compound role with multiple (site specific) resource roles within it.

6.4.5 User data management

Requirement: [4.220] RBAC will import the subscriber database of SAcM

The RBAC user list (in tblUsers) will be compiled from the SAcM Subscriber DB. Most fields regarding a user (e.g. full name, telephone number, department) will be obtained by RBAC by reading the Subscriber DB. The majority of the data held in the Subscriber DB will be accessed from that database, i.e. it will not be replicated anywhere else in RBAC.

Requirement: [4.230] Information that the subscriber database must contain

The subscriber database must at least contain the following correct attributes:

- Login Name
- Name
- Last Name
- E-mail address
- Department
- Department No.
- Department CC.
- Sector
- Site

Requirement: [4.240] Registration of new internal employees

SAcM itself receives its user data for internal users of a Human Resource Management (HRM) tool. This means that when an employee is registered by HRM, SAcM and RBAC will soon be aware of it. Initially this new user will not have a Windows AD account (on the EMEA domain). Because all users of Organon need to have this AD account to be able to use the IT infrastructure of Organon, the first step will be to request a Windows AD account for that new user in RBAC.

Requirement: [4.250] Registration of new external employees

For external users (contractors etc) the Subscriber DB is updated from information obtained from the “Werk Tijden Registratie” (WTR) system. (These users initially also don’t have a Windows AD account.) This information is fed into the same SAcM input stream as for [4.240] and as such then this class of user (external employees) becomes transparent to SAcM and RBAC.

Requirement: [4.260] Registration of new remote employees

Remote users are not registered by HRM, but they are registered in the Subscriber DB and therefore RBAC also has access to them. SAcM is instructed to create a “Remote User” Subscriber in the first instance by members of User Admin team. Data is automatically extracted from the Global Catalog List of the Active Directory.

Requirement: [4.270] information that is lacking in the subscriber database

The Subscriber DB does not hold the following information. This is held in other parts of the RBAC DB:

- Roles of users (e.g. ITC's, Application Manager)
- Lists of Application Managers and other users that are needed to authorization a resource role.

Requirement: [4.280] RBAC user types will be requested as resource roles with RBAC

RBAC will be used to request RBAC privileges. Some of the roles within RBAC will be for the purpose of updating RBAC itself.

Requirement: [4.290] Only Default Users has no access to user data management (???)

Only RBAC managers have the authority to approve RBAC role requests.

6.4.6 Department data management

Requirement: [4.300] RBAC will import department data from SAcM

The SAcM Subscriber records contain user's department, and this information is reliably present in the Organon implementation. As part of RBAC Reconciliation, a list of all the departments held in all the Subscriber records (on multiple instances of SAcM) is used to update the RBAC department list, automatically, on a daily basis.

6.4.7 Department roles management

Requirement: [4.310] A department role contain resource roles

A department role is related to a single department and contains one or more resource roles. Department roles cannot be shared between departments.

Requirement: [4.320] Only ITC' and Metadata Managers have access to department role management

This function is only accessible for those people who have the appropriate "RBAC Logon Screen" entry in the RBAC 'tblUsers' data. In Organon it will be ITCs and Metadata Managers and Superuser.

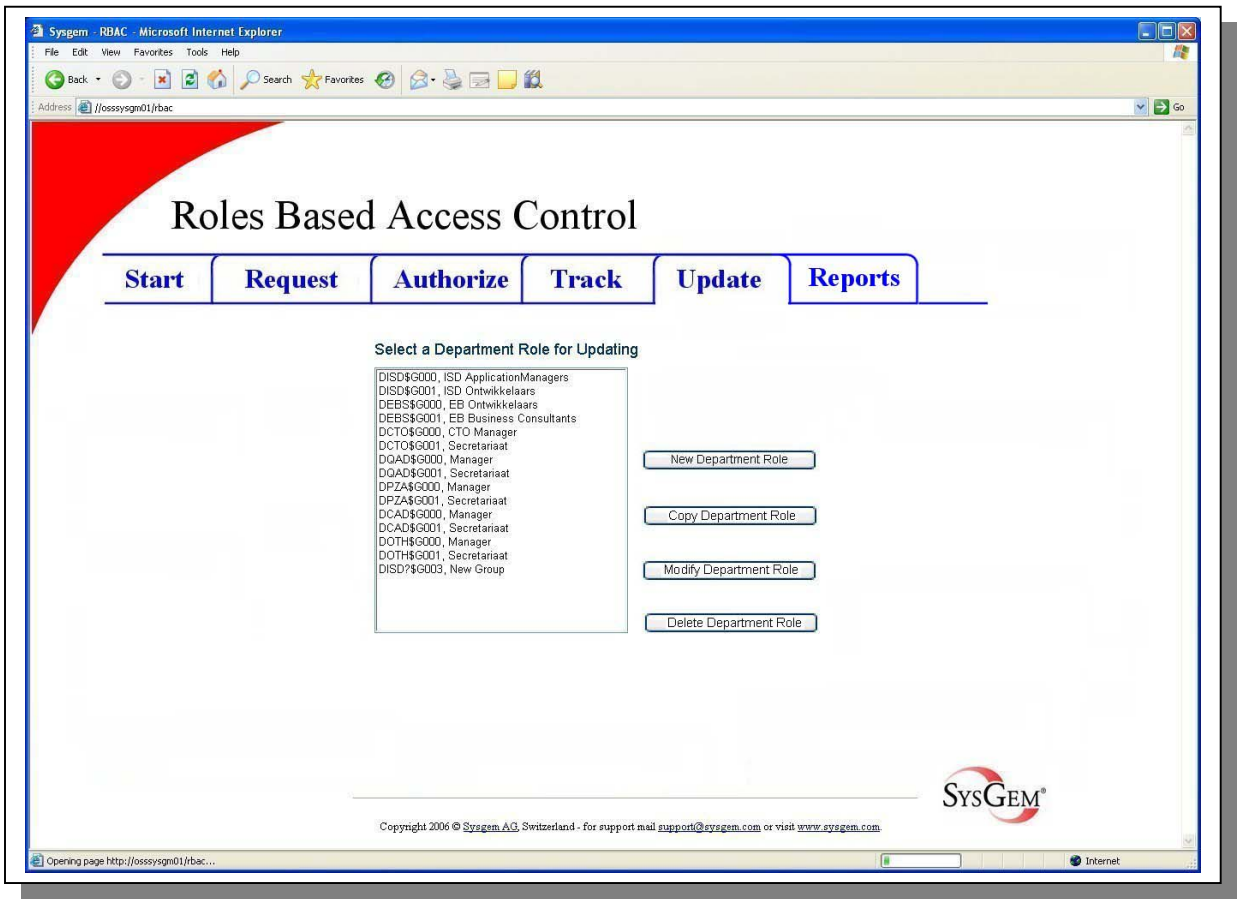
6.4.7.1 Create department role

Requirement: [4.330] Department roles can be created

To create a new department role select the button: “New Department Role” in the “Select Department Role for Updating” form.

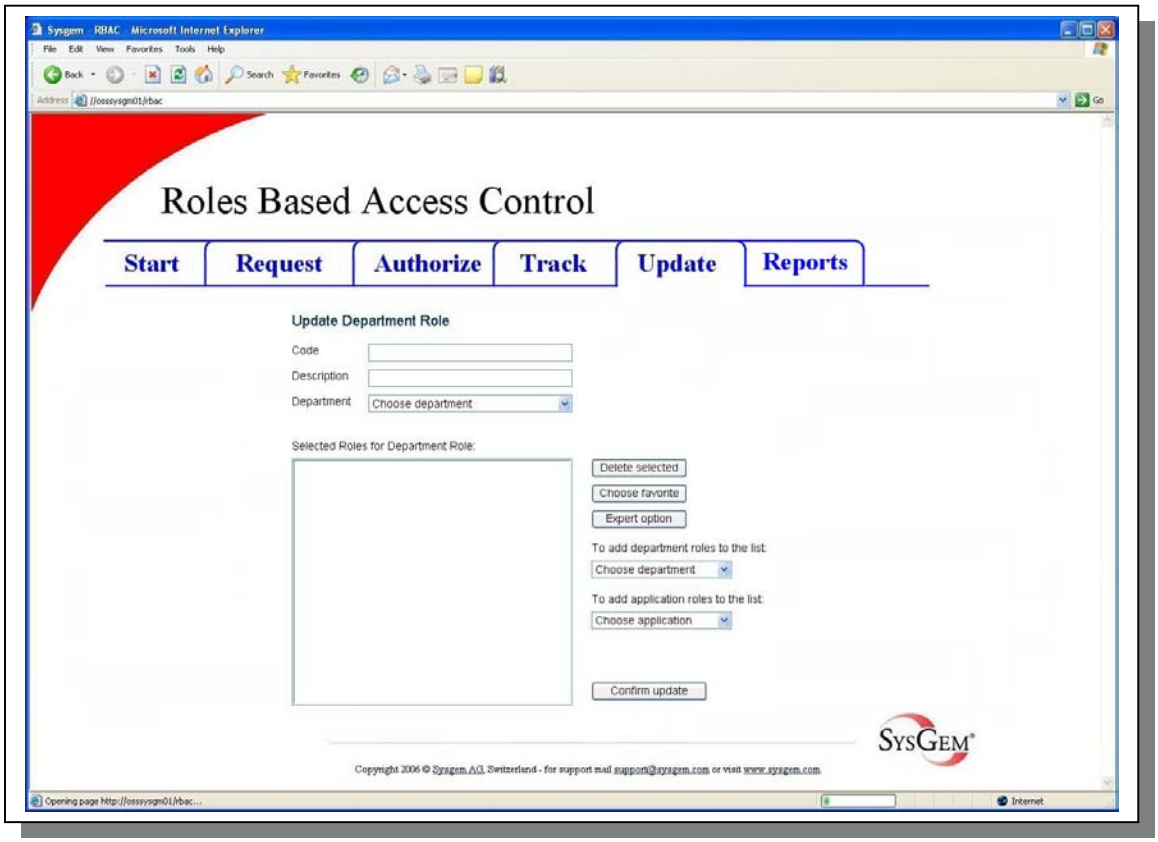
See the following screenshot:

Copy Department Role



RBAC-Mock-up-8.jpg

The following: “Update Department Role“ form appears:



RBAC-Mock-up-9.jpg

The first three controls are used to add a code and description to the department role and to link the department role to a department. The other controls are used for adding or removing roles to the department roles.

Changes to Department Roles, as with all other changes to the RBAC DB are logged to an audit trail.

Requirement: [4.360] Department roles cannot contain department roles

Within Sysgem RBAC it **will** be possible to add compound roles to compound roles. This is contradictory to requirement [4.360]. In the Organon implementation of RBAC, however, it is envisaged that this option will not be used. Organon Department Roles will be a simple, single layer, compound role containing only resource roles at the lower level.

Requirement: [4.365] When adding resource roles to a department role, filter by department or resource

The functionality of adding resource roles to a Compound (Department) Role is almost exactly the same as that for adding users to a request (see paragraph 6.1). The only difference is that the dropdown list with applications contains all resources in the system that

are not department specific and all resources that are specific for the department of the department role.

Requirement: [4.370] Department role are related to one department and at least one resource role

The code, description, name and role fields of the form are mandatory.

6.4.7.2 Update department role

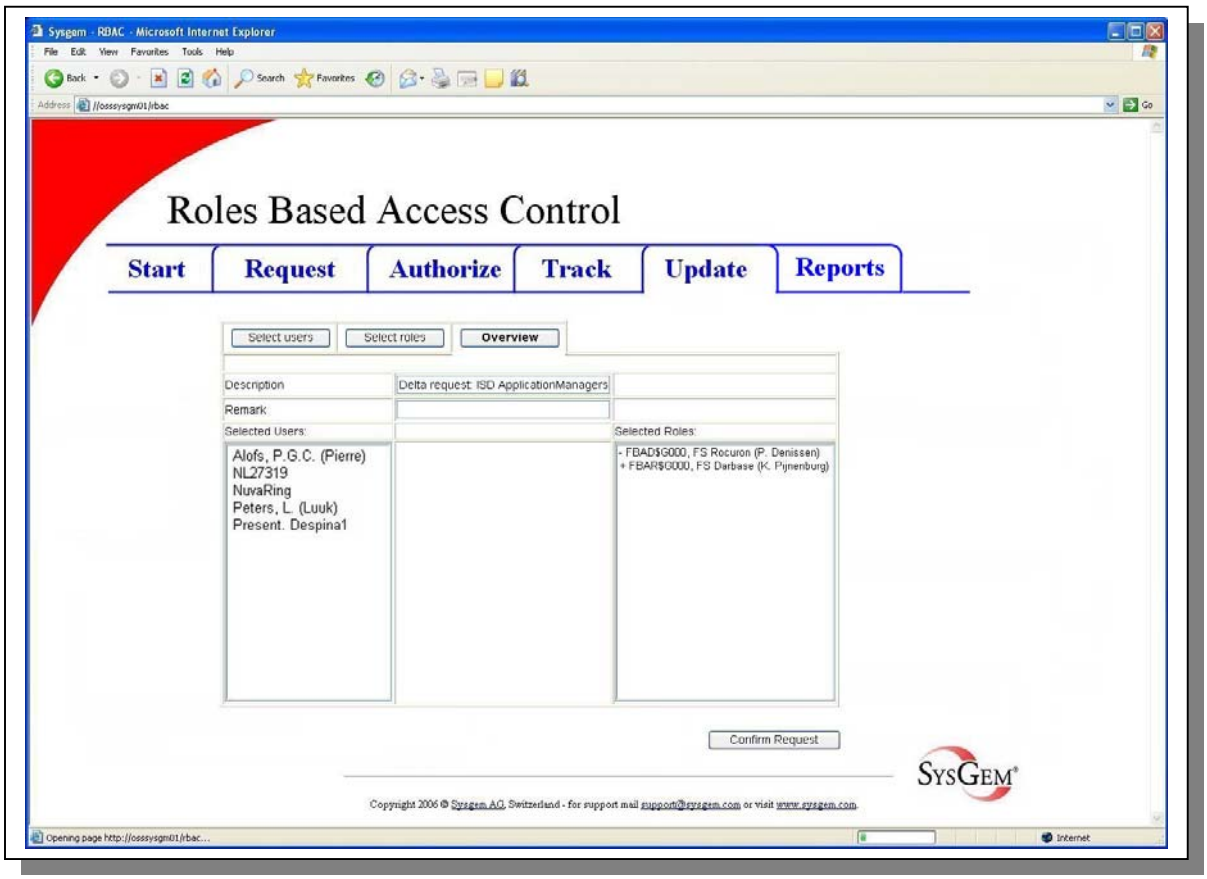
Requirement: [4.380] Department roles can be updated

Updating a Department Role is similar to creating one, except that the owning department cannot be changed.

Requirement: [4.390] A delta-request is computed after updating a department role

The button “Confirm Update” saves the updated department role. Now RBAC computes a so-called delta-request, which contains the difference in roles between the old department role and the new one. WAR transactions are then automatically entered for all users that are already approved for this Department Role.

After confirming, the following form will appear:



RBAC-Mock-up-10.jpg

In this example the delta-request consists of the request for role FBAR$G000 and the revoke of FBAD$G000.

6.4.7.3 Delete department role

Requirement: [4.410] Department roles can be deleted

The form “Select department role for updating” (see the screenshot in paragraph 6.4.7.1) will also allow department roles to be deleted. To delete a department role, all granted resource roles within the department role must first be revoked from all users.

6.4.7.4 Copy department role

Requirement: [4.415] Department roles can be copied

The form “Select department role for updating” (see the screenshot in paragraph 6.4.7.1) will also allow department roles to be copied. This enables ITC’s to base a new department role on an already existing department role. Note: department roles are not interrelated.

6.4.8 Authorization groups

Requirement: [4.420] Authorization groups are related to a single resource

The purpose of authorization groups is described in paragraph 6.2.3.1. Authorization groups are related to a single resource (authorization groups will not be reused to apply on other resources.).

Requirement: [4.425] More than one authorization groups can be associated with a single resource

More than one authorization groups can be associated with a single resource. The authorization groups have a name and users can be added to or removed from the authorization group.

Requirement: [4.430] An authorization group can only be linked to a resource role when the group is linked to the resource

Resource roles are the subject of authorization so the authorization groups of the application must also be linked to the resource roles of the resource. An authorization group can only be linked to a resource role when the group is linked to the resource.

Requirement: [4.440] Authorization groups will be handled in parallel or sequential

When more than one authorization group is related to a resource role, a property of the resource role determines whether the groups will be handled in parallel or sequential. In the latter case the user will be offered functionality to sort the groups.

Requirement: [4.450] Only Metadata Managers have access to authorization group management

This function is only accessible for Metadata Managers.

6.4.9 User roles

Requirement: [4.460] User roles are automatically updated when granting and revoking roles

User roles data describes the link between users and roles. In other words: it describes the actual membership of users to resource roles. Normally this data is updated automatically by RBAC. When a request is approved an executed successfully the link is added, when a revoke is executed the link is deleted.

Requirement: [4.470] Initially user roles have to be entered (manually or interfaced)

The purpose of this function is to initially enter the actual access right of users when reconciliation is not possible for any reason. (For reconciliation of user roles see paragraph 8.4).

6.5 E-mail notification

Requirement: [5.10] Limit risk of delay and inform requestees by e-mail notification

The purpose of E-mail notification is to enable users to keep track of the requests that are of concern to them and to speed up the process of handling requests. Authorization and manual execution are potential bottlenecks for the RBAC process, because they are done by humans. E-mail notifications should help to resolve this bottleneck.

Requirement: [5.20] notifications can be send after confirmation, overdue, authorization and failure

E-mail notifications are sent in the following situations:

- To the Requestor, Requestee and Authorizer when confirmed requests are entered.
- To the Requestor, Requestee and Authorizer when authorization or execution are overdue.
- To the Requestor and Requestee when rejected.
- To the Requestor, Requestee and Application (Resource) Manager when a request action has finished or failed.
- To the Manual Updaters when a resource role that needs manual execution is authorized.
- To the Application Manager and Metadata Manager when execution of a resource role action has failed.

Requirement: [5.30] E-mail notifications will contain a bookmark to the request of interest

E-mail notifications will contain a URL pointing to the request of interest for easy navigation.

Requirement: [5.40] No action of a user is needed to send the notification

RBAC will automatically send the E-mail notifications, triggered by the above events. No action of a user is needed to send the notification. But the user can configure the way E-mail notifications are delivered, and can suppress them if required.

6.5.1 General configuration

Requirement: [5.50] Notification can be per request, per day or none (disabled)

Each user can choose for:

- E-mail notification per request (all notifications for this user will be sent as soon as the event that triggers them occurs). The content of the e-mail message will include a URL that points to this one request transaction.
- Digest per day (all notifications for this user are cached, overnight they are collected and a summary is sent). The content of the e-mail will include a URL for each request

transaction, plus a URL that will give a summary of all new transactions for the day / period.
- None (no notifications are sent). So the user has to take action himself to obtain the information.

Requirement: [5.60] The overdue time span can be configured

Whether a request is overdue or not is determined by the overdue time span attribute of the resource role. RBAC managers or Metadata Managers can configure the overdue time span for sending exception mails to user management for authorizing and manual execution.

Requirement: [5.70] Only RBAC Managers have access to the configuration of the mail server

RBAC Managers can configure which mail server to use.

6.5.2 Detailed configuration

With regard to 'request notifications', a user can have the following functional user types:

- Requestor
- Requestee
- Authorizer
- Manual Updater

These users are made aware of the following transaction state changes:

- Requested
- Rescinded
- Overdue for authorization
- Overdue for execution
- Approved
- Rejected
- Finished
- Failed

Requirement: [5.100] For each combination of functional user types and events, notification can be configured

A user can configure whether he wants an E-mail notification for each combination of these functional user types and transaction state changes.

Exceptions to this rule are:

- Overdue for authorization (is always set to digest)
- Overdue for execution (is always set to digest)
- Execution failed (will always be notified to the Application Manager and Metadata Manager)

Requirement: [5.140] Only the Manual Executor can configure whether he wants notification in relation to a request for manual execution

A Manual Updater can configure whether he wants to be notified about requests for a manual action, using the normal flag states of “On” / “Daily Digest” / “Off”.

Depending on the setting, he will be notified as soon as the request has been authorized.

6.5.3 Request configuration

A Requestor can specify (per request) whether the requestees should be notified or not. When he does request notification, the requestee’s notification settings could overrule the request in which case no notification would be received.

6.6 Execution

Requirement: [6.10] SAcM, Share Management Tool and Manual are used to execute resource roles

When approved, the resource role request-lines of RBAC are submitted to the SAcM WAR DB. This can be done by the following systems:

- SAcM
- Share Management
- Manual Execution

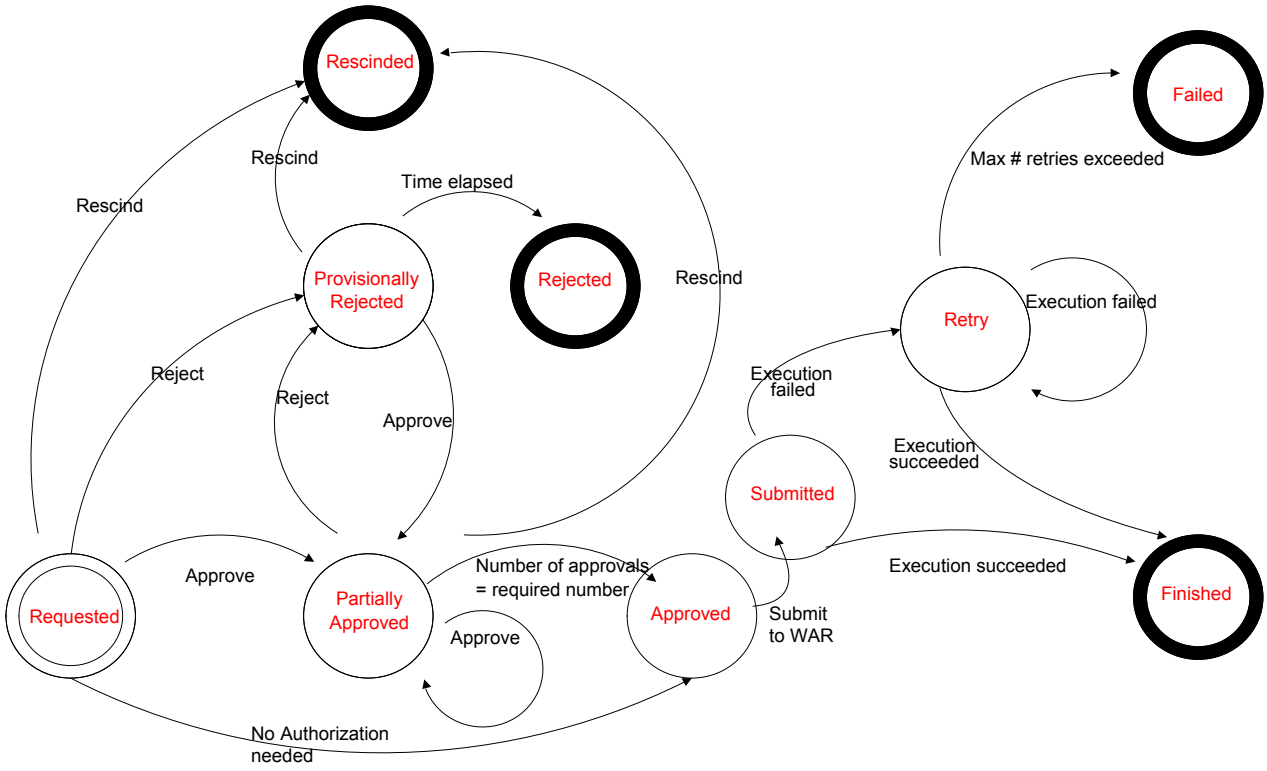
6.6.1 Execution by SAcM

Requirement: [6.20] RBAC places actions in the WAR database of SAcM, SAcM executes the actions

After approval of a request-line for a resource role, RBAC collects the actions related to that resource role and completes the required parameters (See paragraph 6.4.4 for the definition of these actions and parameters). An entry in the SAcM “Web Access Request” (WAR) DB is then written. The request-line now has the status: “Submitted”.

See the following state diagram for the next states of the request-line (this is identical to the diagram shown in Para 6.2.1 – it is repeated here for convenience):

State Transitions:



Requirement: [6.30] When execution fails, RBAC retries after a configurable period

Depending on the nature of the failure, when a WAR action does fail, then instead of changing the status to “Failed” the request line may be resubmitted and the status of the request line changed to “Retry”.

This retry procedure is entirely handled by the WAR Subsystem. The retry policy is determined by a WAR start-up parameter, and the same policy applies to all transaction types.

Requirement: [6.35] When executions fails Application Managers and Metadata Managers are notified

Failure notifications are sent to the Application Manager and Metadata Manager when an execution fails (see paragraph 6.5). RBAC provides them with an overview of the (failed) executions that holds detailed information about the warnings and failures.

When the problem has been (manually) fixed, the status of the transaction may be manually changed to “Submitted” and the thread pickup again from the failed state. But this would only be in extreme cases, since the requestee, requestor, etc have already been notified about the failure.

Sysgem will consider adding an option to RBAC to automatically re-submit a request line, and in this case add additional comment to the transaction to say that this action had been taken, when and by whom.

Requirement: [6.40] RBAC periodically checks the WAR database to retrieve state information

RBAC “knows” whether SAcM executed an action and what state it reached by periodically checking the state of actions in the WAR database.

6.6.2 Execution by Share Management tool

RBAC roles to add (or remove) a user to/from a share performs the following tasks:

- The user is added to (/ removed from) the appropriate (local) group on the appropriate Windows server.
- The user is added to the ShareMgt DB to indicate that he has (/ does not have) access to this share.

There are consequences on the Organon Share Management Tool as the management of individual shares is moved one-by-one from the Tool into RBAC, but these consequences are outside of the scope of this (Sysgem) document. A separate document is required for planning for the changes to the Organon Share management Tool.

There are consequences on the WAR DB “WAR Process Requests” in that scripts need to be added to allow the Organon Share Management DB to be updated as a new command to be called in addition to the update of local shares. See Appendix 4.

6.6.3 Manual execution

Requirement: [6.100] Not all resource roles will be executed automatically

It will be possible to declare that some resource roles will not be executed automatically but instead will result in a request for a “Manual Execution”. This decision may be influenced by:

- Technically to difficult to implement.
- Too much effort to implement.
- Not enough users or requests for a resource to benefit from automatic execution.
- SAcM does not manage the server on which the application is hosted.
- SAcM has to be updated to enable this feature, requiring validation of SAcM or the update.
- Outside the scope of SAcM – e.g. it is a request for a piece of hardware.

Requirement: [6.110] Only Manual Executors are allowed to manually execute resource roles

Authority to update this type of resource role transactions and mark them as “Finished” is a privilege controlled by RBAC, and only those users with the authority of “Manual Updater” or “RBAC Superuser” will be allowed to do this.

Requirement: [6.120] A list with manually executed resource roles is shown to the Manual Executor

When a Manual Updater logs in to RBAC a list with resource roles that have to be executed manually by him will be presented. He has the ability to update the status field on any of these items to:

- Failed (when manual execution was not possible)
- Finished (when manual execution was completed successfully)

Requirement: [6.140] A remark can be added to a manual execution

The Manual Updater will be able to enter a remark (which is mandatory for failures). The user interface will resemble the interface of authorizing request as described in 6.2.

Requirement: [6.150] Initially it is wise to start with manual execution

A customer has the flexibility to declare a resource as requiring “Manual Execution” and can ensure that the correct procedures for requesting and authorizing are correctly in place first. However, the value of this is likely to diminish as more experience is gained with the product.

6.6.4 Execution of conflicting roles

Requirement: [6.160] Resource roles can be conflicting

Resource roles can be mutually exclusive. RBAC will validate requests and if conflicts occur, then error messages will be displayed when the request is confirmed.

Note: Mutually exclusive roles are not, strictly speaking, an issue for Organon. More important for Organon is the use of hierarchical groups of roles, as described in the following section.

See also paragraph 6.4.3.

Requirement: [6.170] RBAC handles conflicting roles by using hierarchy groups

RBAC allows roles to be placed into a hierarchical group during the definition of the metadata. The action taken on the target servers when one of these roles is granted or revoked is dependent on whether there are roles from the same hierarchy group that have already been granted to the user.

Requirement: [6.180] Hierarchy groups are used when granting
…and…
Requirement: [6.190] Hierarchy groups are used when revoking

So, if a role is being GRANTED and there is at least one other role already granted from the same hierarchy then:

- If the already granted role is HIGHER in the hierarchy, then no action will be taken on the target servers. Instead, a record will be added to the Audit log to indicate that the role was granted and that no WAR action was taken because of the presence of the existing, higher level granted role. The RBAC database will now show both the higher and the lower role membership.

- If the already granted role is LOWER in the hierarchy, then RBAC will take the same action on the target servers as if the lower role had been revoked, followed by the action of granting the higher level role. The RBAC database will now show both the higher and the lower role membership.

If a role is being REVOKED and there is at least one other role already granted from the same hierarchy then:

- If the already granted role is HIGHER in the hierarchy, then no action will be taken on the target servers. Instead, a record will be added to the Audit log to indicate that the role was revoked and that no WAR action was taken because of the presence of the existing granted role. The RBAC database will now show that the membership to the lower role has been revoked.

- If the granted role is LOWER in the hierarchy, then RBAC will revoke this (higher) role and then take the same action on the target servers as if the next lower role had just been granted. The RBAC database will now show only the lower role(s) membership.

6.6.5 Re-execution of grants

Requirement: [6.200] Re-execution of grants

As a repair / recovery option against an (undefined) failed situation it is possible to select a set of resource roles authorized to users in the RBAC DB and have them re-applied. The Audit log is appropriately annotated to record that this is this type of transaction and by whom it was made, etc.

[6.210] Selecting resource roles for re-execution

For this purpose there are two types of selection of resource roles:

- Select a request from a list of failed requests
- Select a resource (role) from a list of RBAC users or RBAC roles

Requirement: [6.220] Only RBAC- and Application Managers are allowed to re-execute grants

Only Superuser, RBAC Managers and Application Managers are allowed to re-execute grants. Application Managers are only allowed to re-execute grants on their own applications.

6.7 Auditing

Requirement: [7.10] RBAC keeps an extensive audit trail

One of the advantages of RBAC is its ability to keep an audit trail. For the validated applications of Organon it is not only important to know who has what kind of access now, but also who had access in the past and who authorized this access. With the audit trail it is possible to determine this kind of information.

6.7.1 Logging audit trail

Requirement: [7.20] Information contained in audit trail

The following information will be logged in the audit trail:

- Confirmed requests.
- Approved requests
- Approval or rejection (including implied authorization) of requests.
- Manual execution.
- Completion or failure of an automatic execution.
- Changes in Meta data.

Requirement: [7.30] Optionally RBAC will log the interface data

The data submitted from RBAC to SAcM is always sent via the WAR DB. Records in the WAR database change from the status “Submitted” (before they are processed by SAcM) to “Finished” or “Failed”.

All records in the WAR DB (including the output produced as output from processing on the target systems) are kept indefinitely in the database file. It is a customer option to periodically archive and purge the content of the WAR DB.

Requirement: [7.40] Information contained in log of interfacing

The following fields are contained in the WAR DB:

Command Table:
RequestID
Command
Source
RequestDate
RequestStatus

Parameter Table
RequestID
ParamName
ParamValue

Results Table
RequestID
ResultAgent
ResultContext
ResultFieldName
ResultValue

The content of the WAR “Request ID” field can be used to track the request in the RBAC transaction request tables (e.g. tblRequestLine, whose field “LineID” will correlate to the WAR “RequestID” key) and from that all the information held in the RBAC DB regarding who submitted the request etc. can be obtained.

6.7.2 Accessing audit trail

[7.50] Only RBAC managers have access to audit trail reports

Standard RBAC audit trail reports will be available to RBAC Managers, the RBAC Security Officers and the RBAC Superuser.

At the time of writing, Sysgem are considering that the product: “Sysgem Logfile Concentrator may be used to collate and store log file data and make this available in HTML reports.

6.7.3 Audit security

[7.60] report differences between isst and zoll

Reporting on differences between the actual rights of users and the authorization as stored in RBAC was not in scope for the URS. However, this requirement was later added.

The functionality is implemented within SEM as an extension to SAcM Reconcile. Appendix 2 has details of the Reconciliation tasks.

Only Security Officers and the RBAC Superuser have access to this tool.

If time permits in version 1.0 we will additionally allow reconciliation data to be gathered and displayed in graphical summaries.

6.8 Security

6.8.1 Authentication

Requirement: [8.10] The users are authenticated by their (active) Windows AD account

With the exception of the RBAC "Superuser" (see below), users are identified by their (logged in) Windows AD account.

6.8.2 Authorization

Requirement: [8.20] User must have an active Windows AD account and be a member of the subscriber database of SAcM

All users wishing to access RBAC must have an active Windows AD account and have an entry in the user table of the RBAC database; otherwise access to RBAC is rejected. (Each RBAC user record has a Subscriber record, and the SAcM Subscriber DB is synchronized with the RBAC User table on a daily basis).

The RBAC database and / or AD group memberships contain information defining the RBAC access-permission classifications.

Requirement: [8.24] A difference is made between functional and real user types

RBAC does not draw a distinction between functional and real user types. To access RBAC a user must satisfy [8.20] above; i.e. have a Windows Account and be in the Subscriber DB. Having gained access to RBAC, the user may have additional RBAC privileges as defined in the reply to [8.30] below.

An IT Coordinator does not need to be a member of a department to be the ITC for that department. An ITC will normally be in the department for which they are the ITC - but we do not assume that - the list of departments for which they are the ITC is held as a separate list (in RBAC). AN ITC will, however, normally manage his own department and may also manage several others.

Requirement: [8.27] Real user types are defined by membership of a specific AD group

See reply to [8.24] above.

Requirement: [8.30] User types: ITC, Application Manager, Manual Executor, Requestor, Requestee, Authorizer, Metadata Manager, RBAC Manager, Security officer, End User and Default User

The following categories of RBAC users are provided:

- **IT Coordinator (ITC) (aka Department Manager):** The ITCs are responsible for:

- Authorizing requests for departmental role membership for the departments under their control.

- Authorizing requests for those Resource roles that have been delegated to them by the Resource Managers.

- Managing departmental role metadata, e.g. by associating resource roles with compound roles. (This uses the RBAC Web UI as discussed under section: Department roles management (above))

- Making requests to add / remove users to / from roles. They will most likely use the 'department drop down lists' as their main filtering criteria when selecting both users and roles. Their list of departments is based upon which departments they are linked to. They can, if required, drill down further and make a request for any user in Organon and for any role.

- Taking reports within the confines of the departments that they control. See section: Reporting for more details on the system for reporting.

- **Resource Manager:** Also referred to by Organon as "Application Manager". Resource Managers are responsible for:

  - Authorizing requests for Resource role membership for the Resource roles under their control.

  - Delegating responsibility for request approvals to ITCs. The fact that this privilege has been delegated is recorded in the RBAC DB, and may be revoked at any time by the Resource Manager.

  - Managing the "Resource Roles" metadata under their jurisdiction by associating them with the appropriate Role Actions and Parameters.

  - Making requests to add / remove users to / from roles. They will normally select users and roles by using the Resource dropdown lists as the filter. Similar to ITCs, they can make a request for any user and for any role – and not just the roles they are Resource Managers for!

  - Taking reports within the confines of the resources that they control. See section: Reporting for more details on the system for reporting.

- **Authorizer:** An RBAC Authorizer has permissions to approve or reject requests for particular roles. They can only grant access to the roles that they have permission to authorize. Sometimes, depending on the authorization setting, more than one person has to approve a request line before it is granted.

  Authorization for request approval privileges is granted to an RBAC user by that user being made a member of the appropriate RBAC Authorization Group for the specified role. Only RBAC Managers have the ability to request such changes, and only the RBAC Security Managers have the permission to approve such requests. (See "RBAC

Manager" and "RBAC Security Officer" below)

- **End User:** End-Users are anyone in Organon who has an Active Directory account and an entry in the RBAC Users table. There are three privilege categories of RBAC access for an end user:

  - Ability to make an Enquiry about any outstanding or approved / rejected request relating to themselves.

  - Ability to make a request on behalf of themselves.

  - Ability to make a request on behalf of others.

  Configuration options allow any one of these setting to be the default for RBAC Users.

- **Manual Updater.** A Manual Updater is person who can update the status of a request {plus other fields such as "comment/reason"} for role types that belong to a particular category {suitable for Manual Update} and for which they have permission to update.

  They can change the status of a request to: "Manually completed" or "Failed".

- **RBAC Manager:** The RBAC Manager(s) have responsibility for:

  - Changing access permissions within RBAC. They do so by making an RBAC Request which has to be approved by an RBAC Security Officer.

  - They also have access to a number of configuration options:

    - Configure SMTP server.
    - Configure Timeout time.
    - Configure time that rejected requests will be visible.
    - Configure defaults

- **RBAC Security Officer:** The Security Officer has the following responsibilities:

  - The RBAC Security Officer is the person that approves the above requests from the RBAC Manager

  - They are allowed to browse RBAC Audit Logs and permissions settings.

  - They have the ability to change the second password required for the RBAC Superuser login. (see "RBAC Superuser" below)

  - Taking reports within the scope of the entire RBAC DB. See section: Reporting for more details on the system for reporting.

  - Revoking Access to roles. The RBAC Security Officer has access to the Revoke features whereby any user and any role are available for revoke.

- **RBAC Metadata Manager:** Metadata Managers have access to the following RBAC configuration features.

    - Update resources
    - Update resource roles
    - Update actions of resource roles
    - Update department roles.
    - Configure overdue time.

- **RBAC Superuser**. The RBAC Superuser has the highest RBAC privileges of all the RBAC users. They are primarily responsible for providing RBAC technical Support. They monitor the technical working of the system and make technical changes to "Repair" the system when it is (exceptionally) required. This account type has a separate type of login requiring more than one password.

Further details of the functions provided by these users are described in the appropriate paragraphs of chapter 6.

Note:

At no time is it permitted for anyone to approve a request for themselves, and nor is it permitted for someone to approve the request for someone else who has delegated the approval permissions to them.

Requirement: [8.112] Default Users are only authorized to track requests made for them

An RBAC "End-User" has three RBAC Access categories (See [8.30] above). Under the control of a configuration option, the setting provided by default may be selected as "Enquiry-only" whereby they may only make enquiries (about requests referring to themselves). In Organon terminology, this is a "Default RBAC User".

6.9 Tracking

Requirement: [9.10] With RBAC requestors and requestees can keep track of requests

One of the advantages of RBAC is that requestors and requestees can keep track of requests.

Requirement: [9.20] A difference is made between tracking requests and request-lines

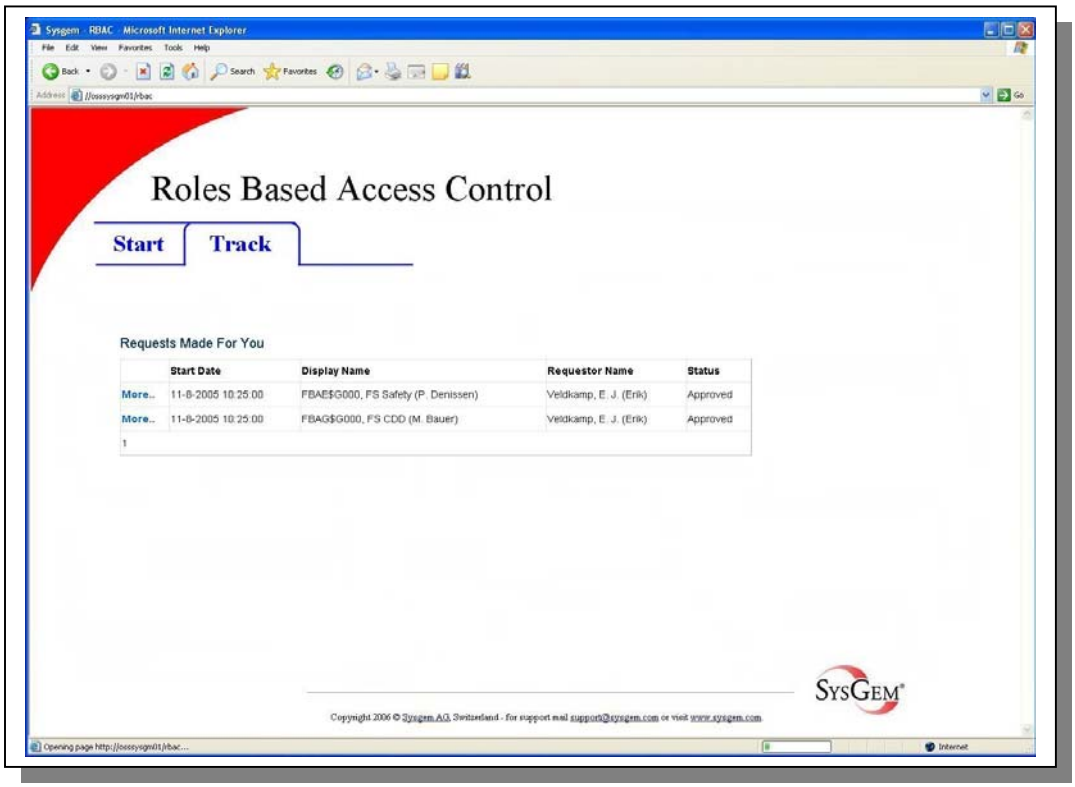
Tracking has the following two levels:

- Request (available to requestor)
- Request-line (available to both requestor and requestee)

6.9.1 Feedback (for requestee) - Tracking

Requirement: [9.30] To track a list with all his requests-lines is shown to the requestee

When the current user is a requestee the “Track” tab will display a list with all his request-lines, as in the following screen shot. If the user does not have any other RBAC privileges then he will not see the “Request”, “Authorize” or “Update” tabs.

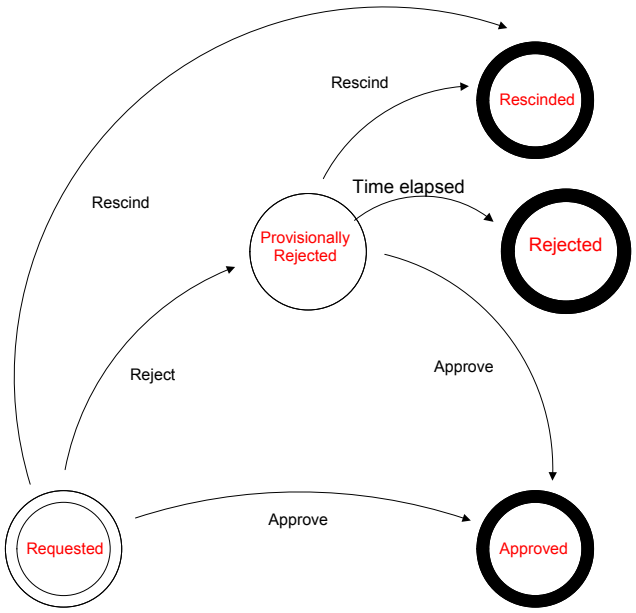


RBAC-Mock-up-11.jpg

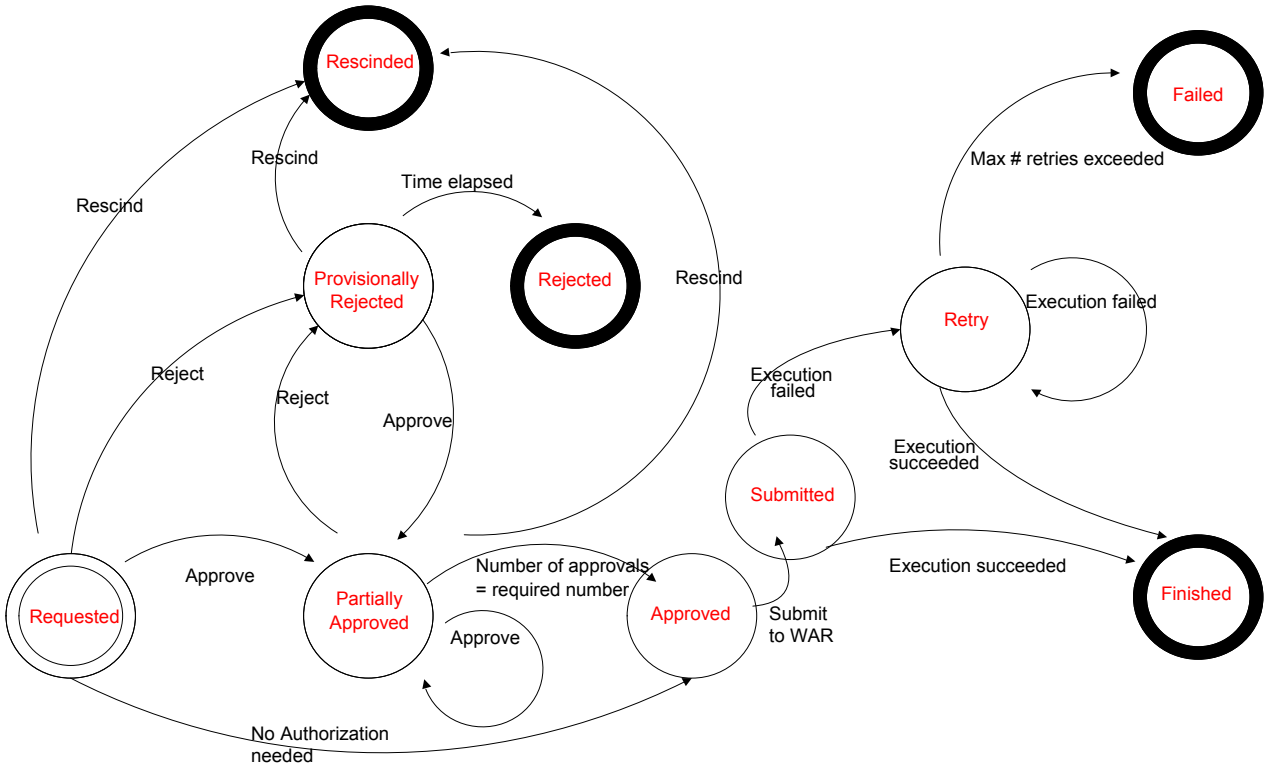
Requirement: [9.40] To track, the functionality number of approvals, approvals needed and failures will be shown to the requestee]

The number of approvals, the number of approvals needed and number of failures will be added as columns to this list.

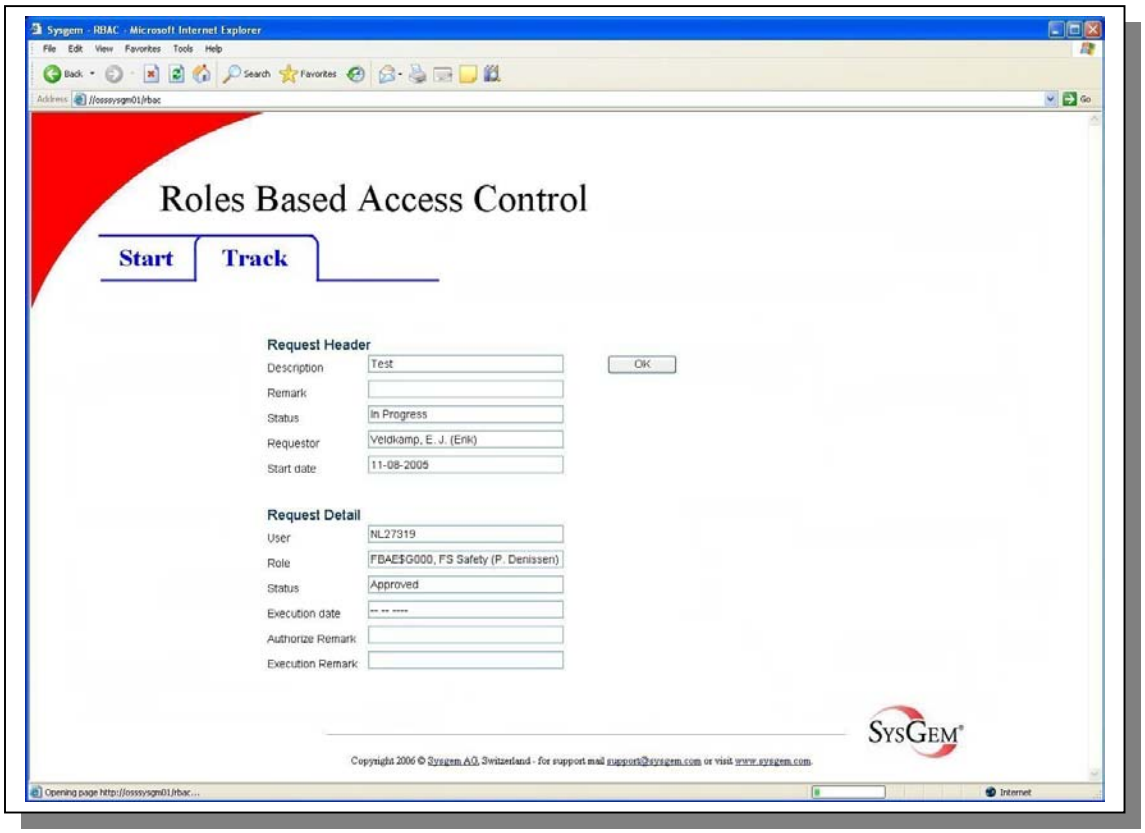
The following state diagram shows the possible status of a **department role request** and their related actions:



The following state diagram shows the possible states of a **resource role request** and their related actions (this is identical to the diagram shown in Para 6.2.1 – it is repeated here for convenience):



By choosing “more” (see the previous screenshot) it is possible to get more detailed information about a request. See the following screenshot for an impression:



RBAC-Mock-up-12.jpg

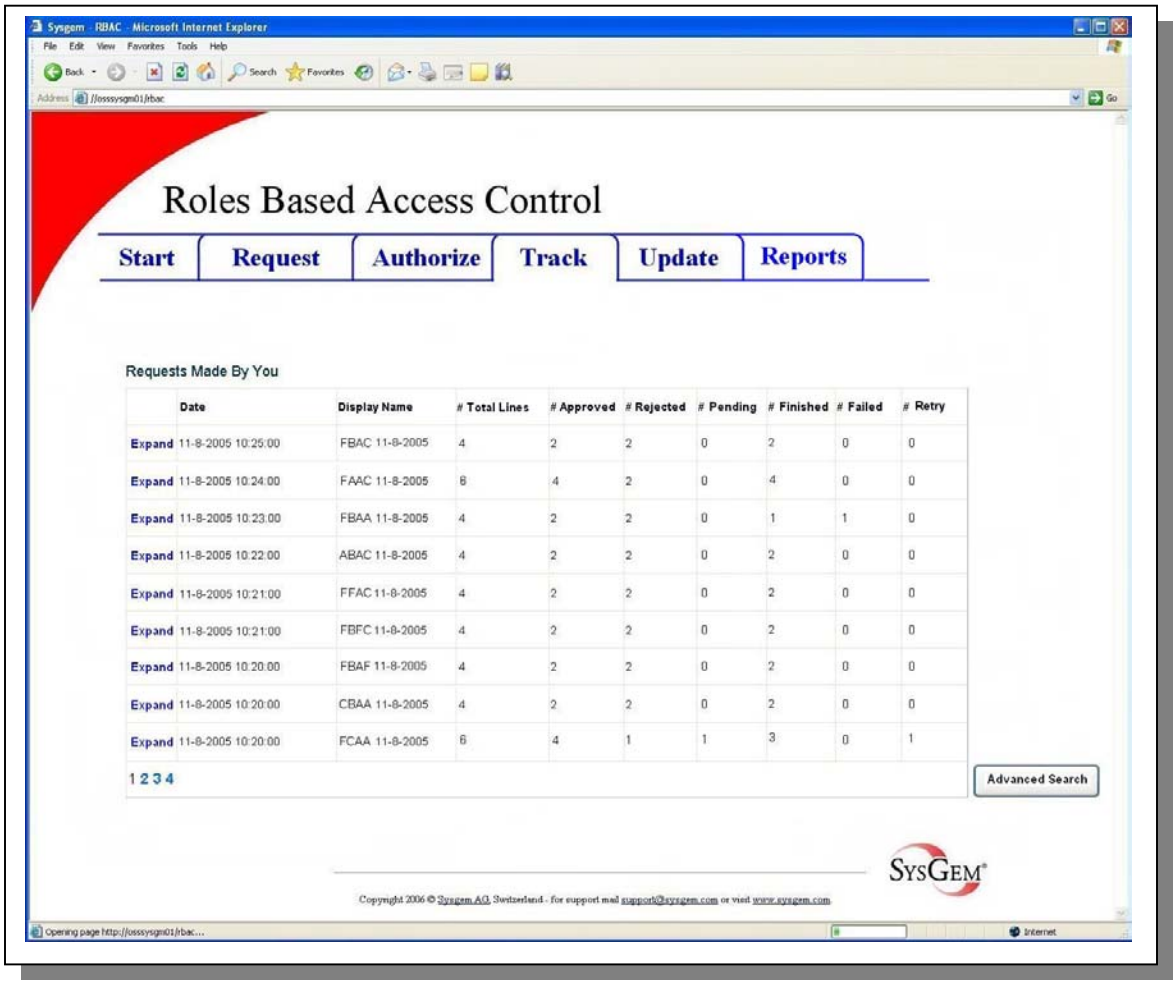
Requirement: [9.70] All users are able to track their requests

Because access can be requested for every user type, all users are entitled to track requests made for them.

6.9.2 Feedback (for requestor) - Tracking

Requirement: [9.80] To track, a list with all his requests is shown to the requestor

When the current user is a requestor the “Track” tab will display a list with all his requests, see the following screen shot:



RBAC-Mock-up-13.jpg

This shows a paginated summary of all the requestor's requests. The list is shown in reverse chronological order.

Associated with each summary line is a button (shown in the example as "**Expand**"). This button will display all the detailed request lines for that one selected request.

It is be possible to filer the complete list of all requests with an "Advanced Search" option that allows: requests to be filtered by Start Date / End Date (of the original submission / confirmation); Text string searches on display Name column; Status column filtering; etc.

Requirement: [9.90] To track, the total number of lines, of approved lines and of rejected lines is shown to the requestor

The "Track Requests" form shows the total number of lines in each request ("#Total Lines"). It shows the number of those lines that have been approved or rejected or still waiting approval ("Pending"). It also show the number of Request Lines that are in the state of "Finished", "Failed" and "Retry" for those lines that have already been submitted to processing by WAR.

Requirement: [9.100] To track, the status of the request is shown to the requestor

The status of a compound role request can be in several states depending on the state of the lower level resource roles. The total number of request lines in each of the different states is shown under the appropriate columns for each request line.

Requirement: [9.120] The requestor can choose to track request-lines

By choosing “Expand” (see the previous screenshot) it is possible to get information about all request lines of a request. This list looks similar to the list shown to the requestee.

Requirement: [9.130] Only default users are not allowed to track request they made

Default Users can track requests that have been made for them. All other users are entitled to track the requests that they made.

6.10 Reporting

Reports are made against:

1. RBAC Audit Log and other log file data (such as the reconcile log data). {i.e. filtered textual reports on static, historical archived data}.

2. Current Active RBAC data. {i.e. statistical charts as well as textual reports on live dynamic data such as roles; departments and users}.

Reports on Static Historical Log Data:

These reports are available to the RBAC Security Officer, the RBAC Manager and the RBAC Superuser.

Audit log data is held indefinitely (it is the customer's responsibility to archive and purge this data using tools provided by RBAC). As such, the reports made against audit data are created interactively, on demand, and usually have filtering applied by date and by other criteria such as by end-user, department, role, or RBAC user.

During the implementation phase, we may also consider the use of the standard Sysgem product: "Sysgem Logfile Concentrator" for audit log reporting. The Logfile Concentrator keeps a separate historical archive of logged events and periodic statistical snapshots of RBAC data. It not only provides web-based (HTML) reports of logged events, but also produces comparison reports showing the differences between the snapshots from the stored data from any two selectable calendar dates.

Reports on Dynamic Active RBAC Data:

For a detailed report on the *current* content of the active RBAC DB, the RBAC reporting tool will generate textual reports that will analyse and report on: live roles data; departments and users; transaction and meta data tables.

The RBAC reporting tool will work within the context of the logged-in RBAC user, and will give access to report types and to data appropriate to the RBAC account privileges for that particular logged-in user.

Reports are generated either automatically on a regular calendar schedule, or are exported on demand by any RBAC user from displayed web pages such as the tracking page, authorization page, request page.

If time permits we will also consider representing statistical data in graphical (chart) format.

In addition to the specific web based reports described above, we will also consider developing customizable Windows based reports using Sysgem Enterprise Manager (SEM) that can also be exported in HTML or a number of other formats.

6.10.1 Report engine

Requirement: [10.10] The report engine will have to integrate with MS SQL server and ASP.NET (C#)

The RBAC reporting tool accesses the RBAC DB structure through the RBAC API, which in turn uses ODBC. In the case of Organon, the ODBC drivers will be MS SQL Server drivers. The reports will be constructed primarily in HTML format using ASP.NET from C#.

Requirement: [10.20] The report engine will have to be able to generate several types of output

The HTML reports will be capable of being exported in the following formats:

- PDF
- CSV
- XLS
- XML

(Can we / do we want to do all these in v1.0? Talk to Kees)

Requirement: [10.30] Only reports relevant for the current user will be shown

Each report is produced in the context of authorized RBAC users. Only report options relevant for the current user are shown.

Requirement: [10.40] The reports of RBAC will allow selection parameters

RBAC reports are filtered using combinations of the following (selection) parameters:

- Resource code
- Department code
- End-User reference (Username, badge number, e-mail address, telephone number)
- RBAC-User reference
- Role ID
- Role type (resource or compound)
- Request ID
- State (Requested; Approved; Rejected; Finished; Failed; Rescinded etc.)
- Transaction dates

Wildcards can be used when entering these parameters.

6.10.2 Printer friendly

Requirement: [10.50] RBAC will provide a printer friendly version of the Web pages

RBAC is a Web based application. Pages that list information (such as the 'Request'; 'Authorize'; 'Track' pages) may be printed using a 'Printer-Friendly' option to format the output as separate reports in the most suitable format for hard copy printing.

6.10.3 Report types

Requirement: [10.60] RBAC will provide various report types

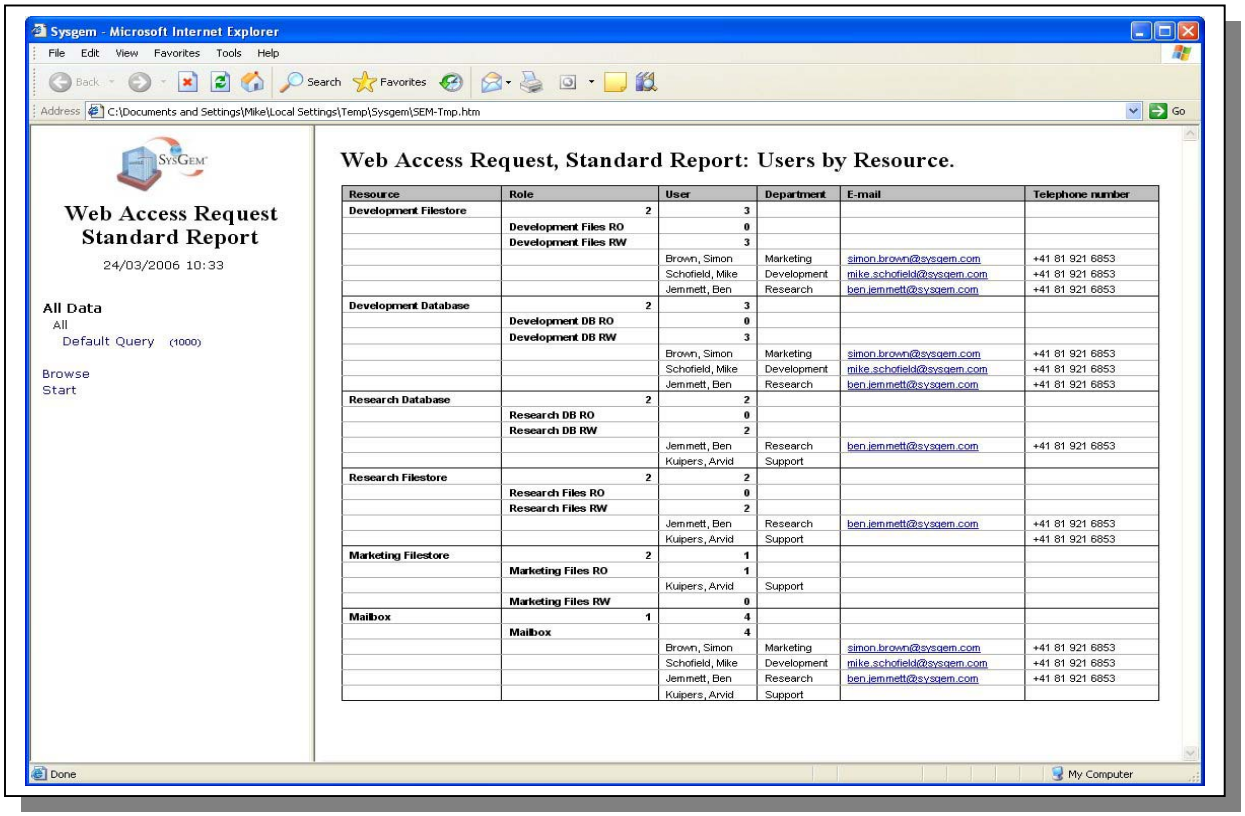
RBAC will at least provide the following reports (this is not intended to be a complete list):

- Summary of auditing data. (e.g. on demand by Security Officer on all transaction data)
- Summary of tracking data. (e.g. on demand by Requestors/Approvers/End Users on their own tracking data)
- Summary of user roles / existing grants. (e.g. on an automatic and regular basis on live roles, departments and users and made available to the Security Officer, or on demand by IT Coordinators/Resource Managers within their own context)
- Summary of performance indicators. (e.g. by IT Coordinators/Resource Managers on the length of time it takes to approve and process requests)
- Summary of security settings and accounts. (e.g. by RBAC Managers on RBAC users, or by Resource Managers on End-User access to resources)
- Ad-hoc reports created by the RBAC Security Officer.

The following mock-up screen shots give examples of three types of report that we intend to produce on an automated and regular basis. Counts of number of roles and users are included in the reports.

Organon should comment on these and make specific requests if additional types of report are required.

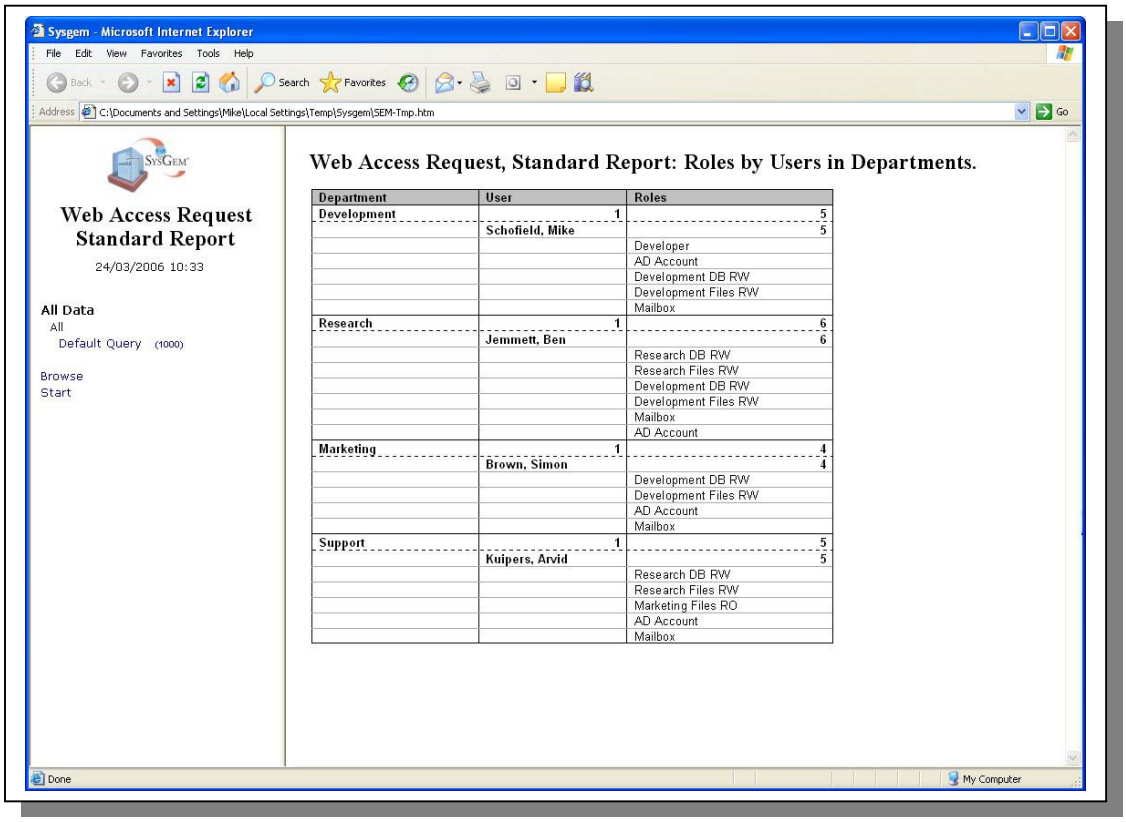
Standard Report 1: Users by Resource:



Resource	Role	User	Department	E-mail	Telephone number
Development Filestore	**2**	**3**			
	Development Files RO	**0**			
	Development Files RW	**3**			
		Brown, Simon	Marketing	simon.brown@sysgem.com	+41 81 921 6853
		Schofield, Mike	Development	mike.schofield@sysgem.com	+41 81 921 6853
		Jemmett, Ben	Research	ben.jemmett@sysgem.com	+41 81 921 6853
Development Database	**2**	**3**			
	Development DB RO	**0**			
	Development DB RW	**3**			
		Brown, Simon	Marketing	simon.brown@sysgem.com	+41 81 921 6853
		Schofield, Mike	Development	mike.schofield@sysgem.com	+41 81 921 6853
		Jemmett, Ben	Research	ben.jemmett@sysgem.com	+41 81 921 6853
Research Database	**2**	**2**			
	Research DB RO	**0**			
	Research DB RW	**2**			
		Jemmett, Ben	Research	ben.jemmett@sysgem.com	+41 81 921 6853
		Kuipers, Arvid	Support		
Research Filestore	**2**	**2**			
	Research Files RO	**0**			
	Research Files RW	**2**			
		Jemmett, Ben	Research	ben.jemmett@sysgem.com	+41 81 921 6853
		Kuipers, Arvid	Support		+41 81 921 6853
Marketing Filestore	**2**	**1**			
	Marketing Files RO	**1**			
		Kuipers, Arvid	Support		
	Marketing Files RW	**0**			
Mailbox	**1**	**4**			
	Mailbox	**4**			
		Brown, Simon	Marketing	simon.brown@sysgem.com	+41 81 921 6853
		Schofield, Mike	Development	mike.schofield@sysgem.com	+41 81 921 6853
		Jemmett, Ben	Research	ben.jemmett@sysgem.com	+41 81 921 6853
		Kuipers, Arvid	Support		

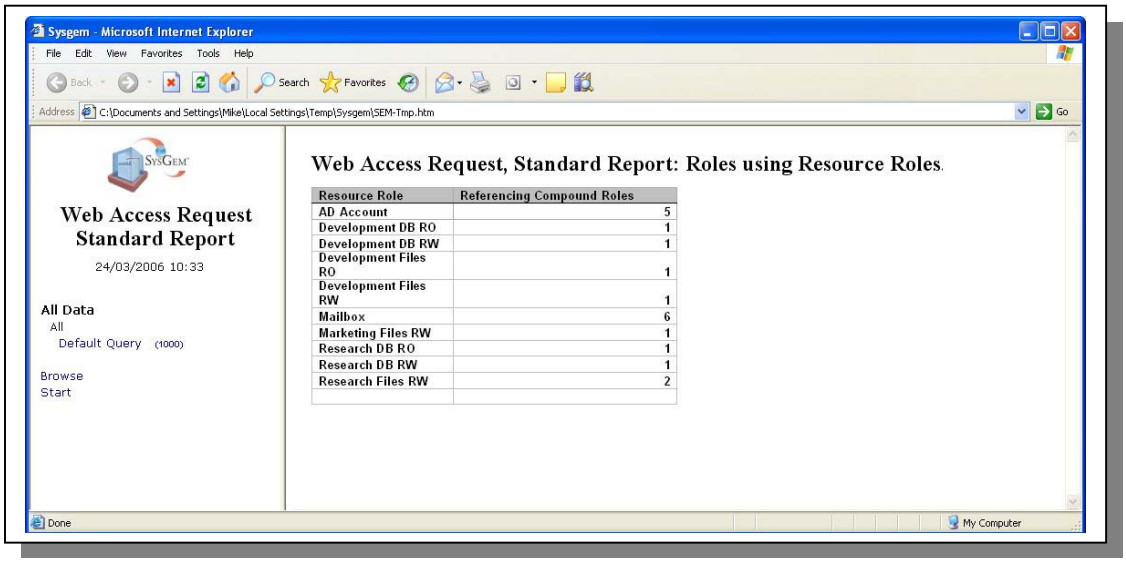
RBAC-Mock-up-Report-1.jpg

Standard Report 2: Roles by Users in Departments:



RBAC-Mock-up-Report-2.jpg

Standard Report 3: Roles Using Resource Roles:



RBAC-Mock-up-Report-3.jpg

6.11 Recovery

In case of disruptions in the infrastructure, the system must continue to provide predictable services. Note that some of the disruptions mentioned below are not within the scope of RBAC.

6.11.1 Unavailability of SAcM agents

If an agent is not available, the WAR command to grant or revoke a role (or related permission) cannot be executed. The command will be marked as 'failed' and reported back to RBAC as such. Note that a planned extension of SAcM is to provide a retry function which would repeat the command a pre-defined number of times (with increasing intervals) before the command will finally fail.

6.11.2 Unavailability of SAcM databases

Requirement: [12.10] Unavailability of SAcM databases

The SAcM Subscriber DB is an integral part of the RBAC DB. If the Subscriber DB or any other RBAC database tables are not available, then RBAC will not be available for making or processing requests until they come online again.

Requirement: [12.20] Unavailability of SAcM databases / new requests

The SAcM WAR database is an integral part of the RBAC DB. Without the WAR DB Request may continue to be entered but transaction will not progress beyond the "Approved" state until the WAR DB becomes available again.

Requirement: [12.30] Unavailability of SAcM databases / status info

The SAcM WAR database is an integral – but discrete part of the RBAC DB. Reporting on the status of transactions will NOT be affected by the absence of the WAR DB except for the WAR "Retry" status, since all other transaction states are recorded in the RBAC tables. If the WAR database is not available, RBAC approved requests (as described in the previous section) cannot be processed, so the status will not progress beyond "Approved" until the WAR DB is available again.

6.11.3 Unavailability of SAcM WAR subsystem

See above. If the WAR subsystem of SAcM is not available, requests entered in the WAR database by RBAC will not be executed. The WAR subsystem will automatically catch up on outstanding requests once the system comes online again.

6.11.4 Unavailability of IIS/RBAC system

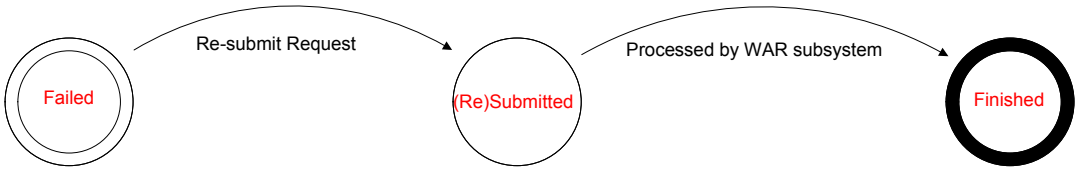
Requirement: [12.40] Unavailability of IIS/RBAC system

If the either the website or the RBAC database are not available, then it is not possible to enter new requests, authorize or execute requests, or get feedback on existing requests. Organon should consider implementing a backup server for disaster recovery purposes.

6.11.5 Failure to execute a request – (Resubmission Procedure)

Requirement: [12.50] Failure to execute a request / hardware failure

If a request fails to execute, e.g. because an Agent was unavailable, then the requestor can resubmit the request. No additional authorization is needed, but the resubmission process requires that additional comment is added to indicate that this is a corrective (resubmitted) action; why it was needed; the identity of the person making the resubmission and a timestamp. This additional information is logged into the audit trail with the transaction.



Requirement: [11.60] Failure to execute a request / incorrect meta data

If a requests fails to execute because of incorrect meta data (e.g. incorrect server name, incorrect share name, incorrect role name, etc), then again, once this problem has been detected and the meta data corrected, the above resubmission procedure is used. This time, it is the Metadata Manager who may request the resubmission without requiring additional authorization. The audit trail will again indicate that this is a corrective (resubmitted) action, why it was required, who made the resubmission and when it happened.

6.11.6 Unavailability of Asset Center

Requirement: [12.70] Unavailability of Asset Center

If the Asset Center is not available, the latest updates in application and server information will not be available. This will not affect existing applications. New applications will not become available until the system / interface comes online again. Direct impact for end-users is low because a number of meta-data activities must take place before actual requests for access to such a new application is possible.

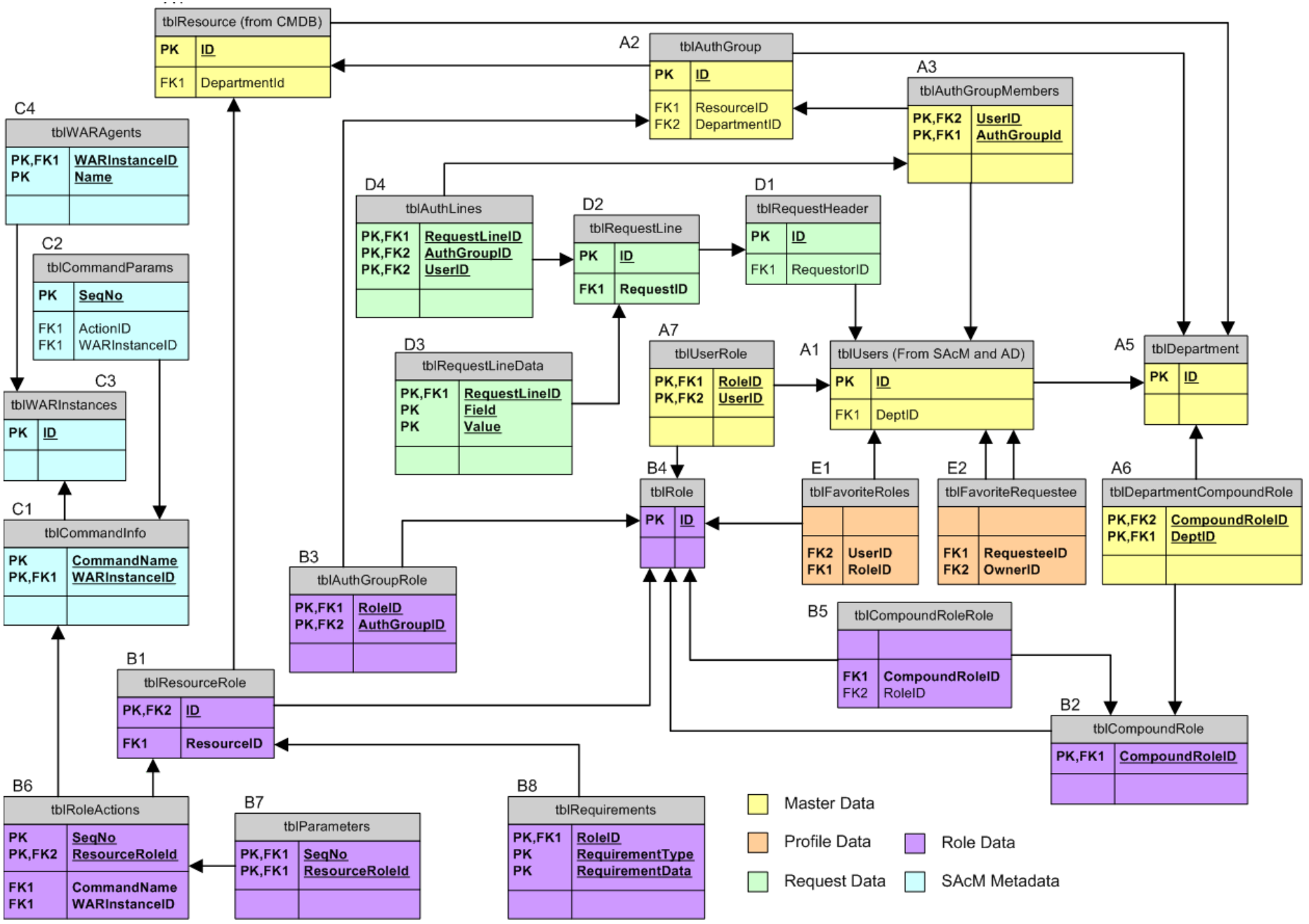
6.11.7 Unavailability of Active Directory

If Active Directory is not available, RBAC will not be available either (as will most of the functionality of the Organon ICT infrastructure).

7 Data collections

7.1 ERD

The Entity Relationship Diagram for the RBAC database (in its current stage of design) is shown on the following page. This diagram may change as the project progresses.


tblResource (from CMDB)
PK ID
FK1 DepartmentId
A2
tblAuthGroup
PK ID
FK1 ResourceID
FK2 DepartmentID
A3
tblAuthGroupMembers
PK,FK2 UserID
PK,FK1 AuthGroupId
C4
tblWARAgents
PK,FK1 WARInstanceID
PK Name
C2
tblCommandParams
PK SeqNo
FK1 ActionID
FK1 WARInstanceID
C3
tblWARInstances
PK ID
C1
tblCommandInfo
PK CommandName
PK,FK1 WARInstanceID
D4
tblAuthLines
PK,FK1 RequestLineID
PK,FK2 AuthGroupID
PK,FK2 UserID
D2
tblRequestLine
PK ID
FK1 RequestID
D1
tblRequestHeader
PK ID
FK1 RequestorID
D3
tblRequestLineData
PK,FK1 RequestLineID
PK Field
PK Value
A7
tblUserRole
PK,FK1 RoleID
PK,FK2 UserID
A1
tblUsers (From SAcM and AD)
PK ID
FK1 DeptID
A5
tblDepartment
PK ID
B4
tblRole
PK ID
E1
tblFavoriteRoles
FK2 UserID
FK1 RoleID
E2
tblFavoriteRequestee
FK1 RequesteeID
FK2 OwnerID
A6
tblDepartmentCompoundRole
PK,FK2 CompoundRoleID
PK,FK1 DeptID
B3
tblAuthGroupRole
PK,FK1 RoleID
PK,FK2 AuthGroupID
B5
tblCompoundRoleRole
FK1 CompoundRoleID
FK2 RoleID
B1
tblResourceRole
PK,FK2 ID
FK1 ResourceID
B2
tblCompoundRole
PK,FK1 CompoundRoleID
B6
tblRoleActions
PK SeqNo
PK,FK2 ResourceRoleId
FK1 CommandName
FK1 WARInstanceID
B7
tblParameters
PK,FK1 SeqNo
PK,FK1 ResourceRoleId
B8
tblRequirements
PK,FK1 RoleID
PK RequirementType
PK RequirementData
Master Data
Profile Data
Role Data
Request Data
SAcM Metadata

7.2 Entity Definitions

Ref.	Table Name	Description
A1	tblUsers:	(List of users – sourced from Active Directory / Subscriber DB)
A2	tblAuthGroup:	(Holds details of each authorization group – name etc.)
A3	tblAuthGroupMember:	(List of users in each authorization group)
A4	tblResource:	(List of resources)
A5	tblDepartment:	(List of departments)
A6	tblDepartmentCompoundRole:	(List of compound roles available to each department)
A7	tblUserRole:	(List of roles currently granted to users)
B1	tblResourceRole:	(Resource-role specific details)
B2	tblCompoundRole:	(Compound-role specific details)
B3	tblAuthGroupRole:	(List of authorization groups required for each role)
B4	tblRole:	(Details common to both resource roles and compound roles)
B5	tblCompoundRoleRole:	(List of roles making up each compound role)
B6	tblRoleActions:	(List of WAR commands to be issued for each resource role)
B7	tblParameters:	(Parameter template for each role action)
B8	tblRequirements:	(List of pre-requisites and conflicting roles for each resource role)
C1	tblCommandInfo:	(List of commands supported by SAcM's WAR interface)
C2	tblCommandParameters:	(List of parameters required for each WAR command)
C3	tblWARInstances	(List of WAR instances available for use)
C4	tblWARAgents	(List of agents available to each WAR instance)
D1	tblRequestHeader:	(Holds overall details of each request – date, requestor, note etc.)
D2	tblRequestLine:	(Holds individual request lines)
D3	tblRequestLineData:	(Data fields for each request line – these correspond initially to input into the user interface, and are created for new lines as the request progresses.)
D4	tblAuthLines:	(List of signatories for each request line – one per authorizing user in each authorization group.)
E1	tblFavoriteRoles:	(List of favorite roles for each RBAC user)
E2	tblFavoriteUsers:	(List of favorite users for each RBAC user)

7.3 Attribute definitions

Below is a list of fields per RBAC table as they are currently defined. This definition is being developed further at this time. The final implemented version will have significant changes.

Master Data:

A1

tblUsers:
RBAC User ID
RBAC Username
Authentication Type
Authentication Data
Department ID
Subscriber Badge Number
RBAC Logon Screen

A2

tblAuthGroup:
AuthGroup ID
Name
Comment
Lead Member RBAC User ID
Is Admin Group
Department ID
Resource ID

A3

tblAuthGroupMember:
AuthGroup ID
RBAC User ID

A4

tblResource:
Resource ID
Department ID
Root OU
Name
Comment

A5

tblDepartment:
Department ID
Name
Comment

A6

tblDepartmentCompoundRole:
Department ID
Compound Role ID
Compound Role Version

A7

tblUserRole:
RBAC User ID
Role ID
Role Version

Role Data:

B1

tblResourceRole:
Resource Role ID
Resource ID
Role Version (always 0)
Manual Executor ID

B2

tblCompoundRole:
Compound Role ID
Role Version

B3

tblAuthGroupRole:
Role ID
Role Version
Role ID

B4

tblRole:
Role ID
Role Version
Version State
Name
Comment
RoleType
ApprovalDeadline
ExecutionDeadline

B5

tblCompoundRoleRole:
Compound Role ID
Compound Role Version
Role ID
Role Version

B6

tblRoleActions:
Resource Role ID
Resource Role Version
WAR Instance ID
Command Name
Execution Order

B7

tblParameters:
Resource Role ID
Resource Role Version
Execution Order
Parameter Name
Parameter Source Type
Parameter Source

B8

tblRequirements:
Resource Role ID
Resource Role Version

Requirement Type
Requirement Role ID
Requirement Role Version

SAcM Metadata

C1

tblCommandInfo:
Command Name
WAR Instance ID

C2

tblCommandParameters:
Command Name
WAR Instance ID
Parameter Name
Parameter Type
Comment

C3

tblWARInstances
WAR Instance ID
Name
DSN

C4

tblWARAgents
WAR Instance ID
Name
FriendlyName
Comment

Request Data:

D1

tblRequestHeader:
Request ID
Requesting User ID
Comment
LastUpdated

D2

tblRequestLine:
Request ID
Line ID
Requested Action
LastUpdated
Status

D3

tblRequestLineData:
Request ID
Line ID
Field Name
Field Value

D4

tblAuthLines:
Request ID
Line ID
AuthGroup ID
User ID
Status
LastUpdated
Comment

Profile Data:

E1

tblFavoriteRoles:
User ID
Role ID
Role Version
Timestamp (Used)
Timestamp (Updated)

E2

tblFavoriteUsers:
User ID
Favorite User ID
Timestamp (Used)
Timestamp (Updated)

8 Interfaces – RBAC Reconciliation with Target Servers

Requirement: [11.10] RBAC will use interfaces to initially fill its database

At the point of deployment, RBAC must be populated with data that reflects the actual state of user accounts at that time. However, Departments will be brought into the system one at a time, so it is not simply a matter of constructing the entire database for the first day of RBAC live-working.

The general principal for the introduction of RBAC is as follows:

- Resource roles for each platform account (Windows/UNIX/VMS) on each server are constructed for all servers that will eventually be brought into the system.
- An initial reconciliation sweep of all target systems will reveal all accounts that are to be introduced into the RBAC DB. This process will be run manually to 'repair' (populate) the RBAC DB, and normally only once - at the start of the project.
- When a new resource role is added into the system, the reconcile procedure will again be run manually, this time to 'repair' the links from the new resource role to the accounts that already possess that resource on the target system.
- When a new Department is added into the system it will be possible for the reconcile process to be run (manually) to repair membership of that Department role to all users who at that time have all the underlying resources that are allocated to the new role. However, IT Coordinators should be aware that this sweep may NOT gather all existing members who should be registered with that Department role, since resource roles can also be removed from a user, even when membership of the Department role itself has not been. So for example, if a Department role has three underlying resource roles, then adding a user to that department role will in fact grant him/her four roles – the department role plus three resources. However, later, one of the resource role memberships could be removed for legitimate reasons, without removing the department role membership. So it cannot be guaranteed that users who should be members of this department role all have all of the underlying resources.
- The reconcile process will be run automatically every day, but the automatic run will make a comparison and will (only) report on discrepancies. It will not automatically repair RBAC role membership as this will defeat the whole object of keeping strict control over the authorization of changes.

8.1 Interface with CMDB

Requirement: [11.30] The CMDB Interface provides RBAC with active resources

Sysgem RBAC will have a generic interface that allows the semi-automatic creation of resource role meta data. It is the responsibility of Organon to extract data from CMDB and feed into RBAC via this defined interface. The quality of the RBAC meta data will therefore (to some extent) be dependent on the quality of the CMDB data.

Requirement: [11.35] All rights to a resource have to be revoked before a resource can be deactivated in CMDB

Before an application can be made inactive in the CMDB, all users have first to be removed from that application in RBAC. Or in other words all rights to the application have to be revoked. Reports shown in paragraph 6.10.3 may be used to show this requirement.

Requirement: [11.40] CMDB is leading in the CMDB Interface

RBAC is not aware of the origin of CMDB data or of the CMDB. Therefore there is no automatic feedback of data from RBAC into CMDB.

Requirement: [11.50] To add a new resource to RBAC, it first should be entered in the CMDB

To add a new resource to RBAC, it first should be entered in the CMDB (also see paragraph 6.4.2). This is a self imposed constraint within Organon.

8.2 Interface with SAcM and Windows Active Directory (user information)

Requirement: [11.60] RBAC will receive information about users and accounts from SAcM and Windows AD

The RBAC DB will be initially populated by and subsequently verified by the RBAC Reconciliation procedure. This is an extension to the original SAcM Reconciliation that reconciled target user accounts with pointers in the Subscriber DB only. The development will extend the scope to cover also the role membership by users that is shown in the RBAC DB.

Requirement: [11.70] SAcM and Windows AD are leading in the interface of the users

The Subscriber DB should be considered as an integral part of the RBAC DB. Existing procedures to synchronize Human Resource data with the Subscriber DB will continue to be in place. Changes that are recognized between the HR data and the Subscriber data will automatically result in an update of the Subscriber data, and depending on the fields will also be updated in the Active Directory. This is no change from the current situation. However, as a result of requirement [11.70], no changes will be pushed back into the Active Directory from any other RBAC component.

Requirement: [11.80] all relevant fields in AD have to be filled in

Because RBAC relies heavily on Windows AD, it is important that all relevant fields in AD have been filled in. This should be an obvious (and obsolete) requirement because this is also in the guidelines from Akzo Nobel. The crucial fields are:

- username,
- display name,
- email address,
- department name,
- department CC,
- department number

Additional information such as:

- title,
- telephone number,
- fax number
- post office box
- country,
- company,
- city
- office

… are useful for display purposes.

All of these fields are updated in the Active Directory as a result of the synchronization of the Subscriber DB with the HR data, when the Active Directory is made to mirror the content of the Subscriber record.

8.3 WAR interface with SAcM (actions)

Requirement: [11.260] The War interface is used to provide SAcM with the desired actions

RBAC implements all its user account actions through SAcM via the SAcM Web Access Request (WAR) DB.

Requirement: [11.270] RBAC places records in the WAR database and polls to see whether SAcM has modified the states of these records

RBAC places records in the WAR database and regularly checks to see whether they have been processed, and what the results of the processing were.

8.4 User role interfaces

Requirement: [11.90] The user role interfaces provides RBAC with information of Resource roles and User roles

The RBAC reconciliation scripts will be run (interactively) to automatically populate all user accounts into the RBAC DB, and subsequently will be run (interactively) to automatically add user membership to other resource role (and if possible – Department role) information.

The process of reconciling resource roles is different for the types of resources (accounts, mailboxes, shares, groups, applications) and will be described in the paragraphs 8.4.1, 8.4.2, 8.4.3.

Requirement: [11.110] The user role interfaces will be logged

Because all grants of access have to be audited, the reconciliation 'repair' of user roles will be logged. The included comment will indicate that the user role was granted by the reconciliation process while 'repairing' membership to the resource role.

Requirement: [11.120] A SOP is needed to assure that when a resource is supported by RBAC, all requests run via RBAC

After integration of a new resource role into RBAC, the requesting and granting access to that role should be done only with RBAC (instead of with a Organon C-form). Updating Organon's Standing Operating Procedures is a separate action required by Organon.

8.4.1 Groups on Shares

Requirement: [11.130] The interface with the Share Management tool provides RBAC with Share User Roles

The Share Management tool contains information about shares, groups on shares and users. This information will be interfaced to RBAC. In RBAC the shares will be represented as resources, the combination of a share and a group will be represented as a resource role; the combinations of a share, a group and a user will be represented as a user having a resource role.

After adding the shares to RBAC, access to shares is given by RBAC. This means that Organon must disable further management of those shares within the Organon Share Management tool.

8.4.2 Standard Software

Requirement: [11.150] Standard Software AD groups all start with OSS_SW

Organon authorizes (and distributes) Standard Software (such as Acrobat Reader) by placing users in corresponding AD groups. As it happens, the names of all these groups start with OSS_SW (but this is transparent to RBAC).

Requirement: [11.160] The interface with AD provides RBAC with Standard Software User Roles

Each Standard Software package has one and only one role (that grants access to the package) therefore the interface to RBAC is quite straightforward. Each AD group corresponding to a Standard Software application is used to create a resource role within RBAC when the meta data is being defined. Similarly, when adding users to the resource roles in RBAC, it is easy to determine that list of users from their AD group membership.

8.4.3 Applications and user roles

Following a discussion between Kees Pijnenburg and Mike Schofield on 09-May-2006 it was decided that assistance in populating “Application roles” would be confined to just two circumstances:

- Obtaining lists of ORACLE role membership information from target ORACLE instances and using this information with a generic input tool to update corresponding roles in RBAC with the same sets of user membership.
- Obtaining lists of Active Directory Group membership and using this information with a generic input tool to update corresponding roles in RBAC with the same sets of user membership.

Requirement: [11.170] Interfacing of Application User Roles has to be researched per application

See paragraph 8.4.3 above.

Requirement: [11.190] The user role interface for applications with a few users can be manual

See paragraph 8.4.3 above.

Requirement: [11.200] The user role interface for applications can link a user to an application role, when the user has all grants needed for that role

See paragraph 8.4.3 above.

Requirement: [11.230] RBAC will need interfaces to AD, Oracle, NT etc. for the application user role interface

See paragraph 8.4.3 above.

Requirement: [11.240] SAcM provides RBAC with all accounts of all users

See paragraph 8.4.3 above.

8.5 Interface of Metadata between production- and test environment

Requirement: [11.250] Interface of Metadata between production- and test environment

To provide a way to interface Metadata between production- and test environments RBAC will be able to export Metadata to XML data and to import XML data to its Metadata.

Only RBAC Managers and RBAC Superusers will have access to this functionality.

9 References

This Sysgem FDS references the Organon document ISD-OYT-FDS-01.DOC (Rev 1.4) that in turn references the following documents internal to Organon:

- ISD-OYT-POCRequests-01.doc
- ISD-OYT-Vision-RoleManagement-01.doc
- ISD-OYT-URS-RoleManagement-01.doc
- ISD-OYT-Design Decisions-01.doc
- ERD-FDS-01.vsd

10 Cross references

Every paragraph in this Sysgem FDS has an equivalent paragraph with the same paragraph numbering as is ISD-OYT-FDS-01.DOC. The relationship between the Organon URS requirements and Organon Functional design specifications are described in the tool Testdirector in project RMS (Role Management System).

11 Appendices

Appendix: 1 – Withdrawn

Appendix 1 to the original Organon FDS has been removed since it highlighted the differences between versions of that document and serves no useful purpose for the history of this Sysgem document.

12 Annex I: Database Access

This annex describes the design of the database access layer shared by all RBAC components. The design is not yet frozen and may/will change as the implementation progresses.

12.1 General Design

In general terms, the database will be accessed via a set of entity classes. Each entity class will relate to one of the entity tables in the database schema, and will provide methods for looking up specific instances of an object, creating new instances of an object, and modifying the attributes of existing objects.

The entity classes will provide access to related entities by providing operations such as “get all request lines” on a request header object, or “get all administrators” on a department object. Many-to-many relationships will also be accessible through classes corresponding to junction tables, which will provide operations to get and set associations from either side of the relationship.

The set of available classes for RBAC data will be:

Entity classes
User
Department
Resource
AuthGroup
AuthLine
RequestLine
RequestLineData
RequestHeader
Role, with two subclasses:
    ResourceRole
    CompoundRole
Requirement
CommandInfo
CommandParam
RoleAction
Parameter
WARInstance
WARAgent

Junction classes
AuthGroupMembers
UserRoles
DepartmentCompoundRoles
CompoundRoleRoles
FavoriteRoles
FavoriteUsers
AuthGroupRoles

Other classes will be available for data used internally by system; these tables will be decided upon when the internal design is complete, but are likely to include:

Web System	**Transaction Processing**	**Reconciliation**
Form	Notification	Account
FormSection	ScheduledTasks	Exception
FormField		ReconciliationReport
ListSource		

The exact design of these classes – member names, operations provided – will be decided as the design progresses, to allow for convenience methods to be included if they appear useful etc. Brief examples of the prototypical implementation for .NET and Perl are shown below.

12.2 .NET implementation

The .NET implementation of the data access layer will be shared by all .NET-based components in the system, including the Web user interface and the back-end processing tasks. Associated objects will be returned in typed lists, allowing the code to be proven type-safe at compile time.

Code accessing the .NET API might resemble:

```
using RBAC;

List<RBAC.User> users = RBAC.User.GetAll();
foreach (User user in users)
{
    HtmlTableRow  row      = new HtmlTableRow();
    HtmlTableCell cellUser = new HtmlTableCell();
    HtmlTableCell cellDept = new HtmlTableCell();

    cellUser.InnerText = user.Username;
    cellDept.InnerText = user.Department.Name;

    row.Cells.Add(cellUser);
    row.Cells.Add(cellDept);
    table.Rows.Add(row);
}
```

12.3 Perl implementation

The Perl implementation will take a similar form to the .NET implementation, with method and class names being identical where possible allowing for language differences. The Perl classes are anticipated to be used mainly be reconciliation tasks and other interfaces with existing SAcM code, which is also Perl based.

Code using the Perl API might take the form:

```
use RBAC::Database;
use RBAC::User;
use RBAC::Role;

my $database = new RBAC::Database();

my $role = $database->findResourceRole($rolename);
my @users = $role->getAllUsersWithRole();
foreach my $user (@users)
{
    my $username = $user->Username();
    print "User $username has role $rolename.\n";
}

$database->close();
```

13 Annex II: Web Front-end

This annex describes the design of the web front-end, which handles all interactions with the system based around a set of form definitions. The design is not yet frozen and may/will change as the implementation progresses.

13.1 Logging in

The initial login page will present the user with a username and password field. These values will be used to authenticate RBAC-authenticated users listed in the tblUsers table. Once a user is found in the table, their initial page is displayed – if no initial page is assigned for the user, they will have no end-user access to RBAC and an error will be displayed.

However, if the user has used NT Integrated Authentication to connect to the application, the user table will be searched for a username equivalent to the user's Active Directory username, as supplied. If one is found, the prompt will be skipped and the user will be presented with their initial page immediately.

13.2 Entry points

There will be two 'entry points' into the web system. The first will be the user login page, as described above. The second will be through links embedded in notification e-mails sent out by the transaction processor. Such links will be to an entry page, and will pass a notification ID in the query string. For example:

http://rbac.sysgem.com/notificationLink.aspx?notification=3279

The page will first check that the specified notification is valid, and then find the RBAC user the notification was sent to. If the user requesting the page is using NT integrated authentication and has the same Active Directory username as the user being notified, then the type of notification is determined and the relevant page is displayed – for instance, the approval page for a request or a page showing the overall progress of a request.

If the user requesting the page is not using NT integrated authentication, or is not using the Active Directory account associated with the user who was notified, the page will display the usual RBAC login form with a message requesting the user logs in using the correct account. Once the correct login is obtained, the notification link is handled as described above.

13.3 Form rendering

Several types of form will be supported:

- Menu forms, providing links to several other forms
- List view forms, listing all rows in a database matching certain criteria and providing the ability to define actions available on each row

- Item view / modify forms, showing a single row in a database in more detail

- Custom forms, which are rendered by custom sections of .NET code but can still use the usual field processing when submitted

The overall form type determines how the field definitions are interpreted. A simple form will contain just an ordered list of fields of various types, whereas a list view form is split into sections with some sections labeled as lists. Each list section repeats for as many entries are selected by the assigned query. One instance of each field within the section is generated per entry.

Various types of field are available:

- Static text fields – used for paragraphs of text etc.

- Free-form text entry fields, in single and multiple line versions

- Multi-select and single-select list boxes

- Links and buttons

Each field has an associated text value (display text in the case of a static text field, and caption for all other fields), a position within the page, and visual style attributes such as bold, italic and highlight.

Editable and variable fields are associated with a data field. These fields reflect the current values in the specified field, and (if editable) are saved back to the field. Typically the data fields are stored in the request line, although various other read-only sources are available (such as the request header, an authorization line, and the role and user data).

List sources have a list data source defined, which can either be a fixed list or derived from the database. Selected entries in the list are written as a set of lines to the request line data.

Links and buttons provide the ability to move to different forms, either directly (saving or discarding any changes to the data on the current form) or after performing an action such as adding the current request line to the current request, or submitting the request.

14 Annex III: Transaction Processing Back-end

The transaction processor runs as a background service, checking on the status of requests in the database and handling them as required. Its major functions are:

- Approval management
- Notification management
- Request submission and progress monitoring
- Scheduling of regular tasks

These are described in more detail below. Please note that the design is not yet frozen and may/will change as the implementation progresses.

14.1 Approval Management

Once a request is marked as submitted by the web interface, it is considered as owned by the transaction processor and no further changes can be made to it by the web interface. The transaction processor examines the details of the request and creates the required set of request lines and associated authorization lines, and marks the request as being in the approval cycle.

The transaction processor handles the state transitions involved with the approval cycle and all other parts of the RBAC system. The ability will be provided for customer-specific code to be called at each state transition, with the custom code able to override any transition or provide supplementary action as needed. This custom code will be run within the .NET system with full access to RBAC's database APIs, and a wrapper module could be provided to support such plug-in code written in Perl.

14.2 Notification Management

The transaction processor is responsible for sending out notification e-mails as and when required, taking into account the user's notification preferences stored in their profile data.

When an event requiring notification occurs, an entry is made in an outstanding events table for each affected user. This entry includes a description of the event, what action is required, and optional data to be used by the web front-end to display further details and/or options. Each entry has a timestamp, and the transaction processor periodically checks for new entries. These are then collected together into an e-mail and sent to the user.

If the user has elected to receive a daily digest, the transaction processor will only gather entries on a configurable schedule; otherwise entries are sent to the user as they are generated with a small delay to allow large batches of simultaneous notifications to be dispatched in a single e-mail.

14.3 Request Submission

Once a request line completes the approvals process successfully, the transaction processor builds the correct entries in the target WAR database. The entries are built using tblRoleActions and tblParameters as a template, and sourcing per-request data for the most part from tblRequestLineData.

15 Annex: IV – Reconcile

TBA – MKS to draft. This will give a detailed description of all the Reconciliation tasks.

16 Annex: V – Required Extensions to the SAcM WAR DB

The following is the list of WAR commands that are currently envisaged (in addition to the already provided set) so as to be able to implement the proposals in this document:

1. add account to global group
2. remove account from global group
3. add account to share group
4. remove account from share group
5. create Oracle user
6. grant role to Oracle user
7. revoke role from Oracle user
8. grant privilege to Oracle user
9. revoke privilege from Oracle user
10. create Unix user
11. add Unix user from Unix group
12. remove Unix user from Unix group