Meta Resource Management System

Use Cases

René Freude (707168) <rene.freude@gmx.de>
Daniel Sadilek (707297) <sadilek@tfh-berlin.de>
Stephan Weiß (706830) <steph.weiss@web.de>

2003-05-23

Revision History

Revision 0.1	2003-05-05	
	Initial public release.	
Revision 0.2	2003-05-23	
	Second release.	

This document specifies the use cases of the MRMS. Additionally it contains example uses of all use cases in one cohesive usage scenario and a glossary of the terms used.

Table of Contents

1. Usage scenario

This usage scenario presents an example environment for the MRMS. It serves as a concrete example from which we abstract the use cases. We have the problem that the MRMS is a very data centric system. So, usually you would begin to create the static analysis model and later derive the dynamic analysis model from it. Unfortunately, the time plan of the course, the MRMS is developed in, does not allow such an approach. Thus, this usage scenario serves also another purpose: Its lists and tables give you an idea of the data that will be managed by the system and helps you (and us) understanding the use cases better.

We assume that the system should be used in a business called Exempler Inc. with approximately fifty employees. In this business there are to manage the employees, their workplaces with computers and tables, the rooms in which the workplaces are and the software licenses installed on the computers. An analysis of these resource types resulted in the following list of attributes that should be saved for the resources:

Table 1. Resource types at Exempler Inc.

resource type	**attributes**
room	number, number of windows and maximum number of workplaces
workplace	description
computer	type of chassis and processor, processor frequency, size of memory and harddisk, monitor manufacturer and make
software license	name of software, type of license, license data, price, expiry date, maximum number of usages
table	material, size
employee	name, address, weekly hours of work, hourly rate

Resources of these types should be organized in the following manner:

- Rooms contain workplaces; the maximum number of workplaces for a specific room is saved as an attribute of the room.
- A workplace belongs to exactly one room.
- A workplace consists of at least one computer and one table but may also have more.
- A computer as well as a table belong to exactly one workplace.
- A computer may contain any number of software licenses.
- One software license may be installed on as many computers as saved in an attribute of the software license.
- Employees work at at least one workplace but may work on more.
- One workplace may be used by any number of employees.

Taking this information we can create a table of link rules:

Table 2. Link rules at Exempler Inc.

resource type 1	**cardinality 1**	**cardinality 2**	**resource type 2**
room	1	0..room.maxWorkplace	workplace
workplace	1	1..*	computer
workplace	1	1..*	table
computer	0..swLicense.maxUsages	*	software license
employee	*	1..*	workplace

There are different roles with different responsibilities:

Table 3. Roles at Exempler Inc.

role	responsibilities
chairman	manages employees and may also see and edit their hourly rate
server administrator	administrates the IT infrastructure in the server room and manages software licenses (e.g. buys new ones); is marked as MRMS administrator role and thus has administration access to the MRMS
client administrator	administrates and manages all computers that are not in the server room; assigns computers to workplaces and installs software licenses on the computers but may not manage the software licenses by themselves and has *no* administration access to the MRMS
facility manager	manages rooms, workplaces and tables: knows how many workplaces can be and how many are in a specific room and knows if a specific workplace is unused; assigns employees to workplaces but may not see the employees' hourly rate

The roles have different access rights that are assigned by an MRMS administrator, i.e. an owner of a role that is marked as MRMS administrator role (in our example this is the role “server administrator”). There are the following types of accesses (the points where the access rights are defined are in brackets):

- create a resource (resource type)
- delete a resource (resource type)
- read an attribute of a resource (attribute of the resource type)
- edit an attribute of a resource (attribute of the resource type)
- link two resources (link rule for the two resource types)
- unlink two resources (link rule for the two resource types)

The following table shows the access rights the roles in our example have. In the rows for the resource types “full” indicates that the role may create and delete resources of this type and may read and edit all attributes of resources of this type, “read” indicates that the role may read all attributes of a resource of this type. In the rows for the link types “full” indicates that the role may link and unlink the resources, “none” indicates that the role may neither link nor unlink the resources. Note, that reading/following a link is always possible for a role if at least one attribute of the link end can be read by that role.

Table 4. Access rights for the roles at Exempler Inc.

resource type	**chairman**	**server administrator**	**client administrator**	**facility manager**
room	full	read	read	full
workplace	full	read	read	full
computer	full	read	full	read
software license	full	full	read	read
table	full	read	read	full
employee	full	read without “hourly rate”	read without “hourly rate”	read without “hourly rate”
link type	**chairman**	**server administrator**	**client administrator**	**facility manager**
room - workplace	full	none	none	full

link type	chairman	server administrator	client administrator	facility manager
workplace - computer	full	none	none	full
workplace - table	full	none	none	full
computer - swLicense	full	none	full	none
employee - workplace	full	none	none	full

The following employees work at Exempler Inc. and take one or more of the roles above:

Table 5. Employees at Exempler Inc.

employee	role(s)
Bernadette Bossa	chairman
Helene Hacker	server administrator
Chap Compscrew	server administrator and client administrator
Chris Clientow	client administrator
Florian Facilitan	facility manager

Installation

1. Helene Hacker installs the system on a server computer.
2. She logs in with the default administrator username and password; see Section 2.1.1, “User logs in” [7].
3. She changes the administrator password; see Section 2.1.3, “User changes password” [8].
4. She defines the types of the resources that should be managed (see in the table above). That includes specifying which attributes should be saved for a specific resource type; see Section 2.2.1, “Define resource type” [12].
5. She enters the link rules from the table above; see Section 2.2.2, “Define link rule” [13].
6. She creates the roles of Exempler Inc. from the table above. This includes defining the rights that these roles have (see the table above); see Section 2.1.6, “Create role” [10] (includes Section 2.1.7, “Edit rights for role” [11]).
7. She creates user accounts for all employees that take one of the roles; see Section 2.1.4, “Create user account” [9] (includes Section 2.1.5, “Edit role assignment of user account” [10]).
8. After these initial configuration steps she logs out; see Section 2.1.2, “User logs out” [8].
9. Then she informs the future users of the system about their usernames.

Initial data input

Now the system is set up and configured and the users have their usernames. They can begin to enter the resource data into the system; see Section 2.3.1, “Create resource” [15].

- Chris Clientow creates all computers he manages.
- Chap Compscrew creates all software licenses he manages.
- Florian Facilitan creates all rooms, workplaces and tables he manages.

(All of them log in for the first time and must set their passwords; see Section 2.1.4, “Create user account” [9].)

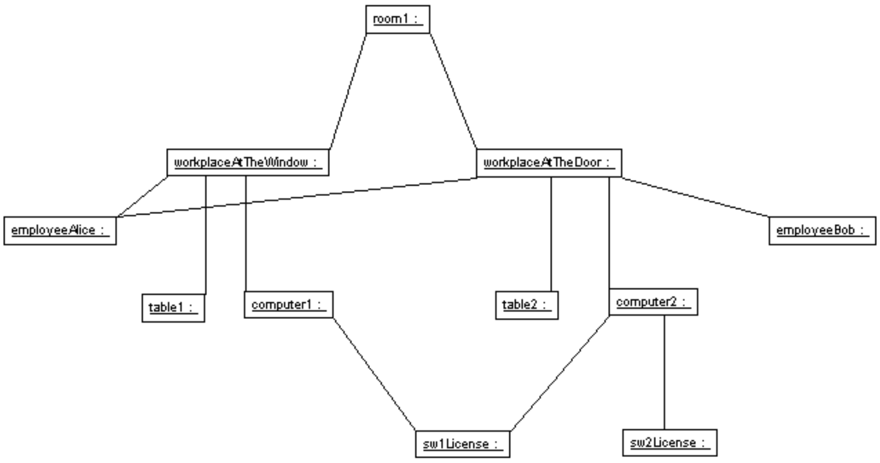
Initial link input

Now that all resources have been created in the system, the users create links between the resources to generate an image of the real resources' interrelationsships; see Section 2.3.4, “Add resource link” [17].

- Chris Clientow links the software licenses to the computers they are used on.
- Florian Facilitan links the workplaces to the rooms they are in and to the computers and tables they consist of.

The following diagram shows a small excerpt from the data the users have entered. We will use it for the subsequent examples. You can see that there is just one room with two workplaces. Every workplace has its own table and computer. Both computers use the license for software 1 and computer2 (that belongs to the workplace at the door) also uses the license for software 2. Two employees work in this room: Alice and Bob. Bob uses the workplace at the door and Alice uses the workplace at the window. Sometimes, when she needs software 2, Alice works at the other workplace as well. (Note, that the “sometimes” is not saved in the system).

Figure 1. Some resources at Exempler Inc.



Working with resources

Here we present some possible events that necessitate to work with the MRMS in order to correct the MRMS's image of the real resources or to find free resources:

Table 1 is old and broken and needs to be replaced:

1. Table 1 gets discarded.
2. Florian Facilitan deletes table 1 from the system; see Section 2.3.2, “Delete resource” [16].

3. If he would know check the consistency, he would be reported that the workplace at the window does not have any table which is required by the link rules; see Section 2.3.7, “Create filtered collection of resource entries” [19].
4. Another table, table 3, is bought and brought to the workplace.
5. Florian Facilitan creates table 3 in the system; see Section 2.3.1, “Create resource” [15].

Table 3 is not solid enough and needs to be replaced:

1. Florian Facilitan removes the link between the workplace at the window and table 3; Section 2.3.5, “Delete resource link” [18].
2. He does not create a new link with table 3, i.e. the table 3 now has not got any links to any other resources: it is a free resource. For this reason, Florian Facilitan transports the table to the business' storage.
3. Another table is bought, ...

Alice needs more memory for her computer 1:

1. Chris Clientow installs more memory into computer 1.
2. He edits the data of the resource “computer 1” in the system; Section 2.3.3, “Edit resource” [17].

The license for software 1 should be used on Alice's computer 1:

1. Chap Compscrew removes the license from computer 2 and installs it on computer 1.
2. He reproduces this in the system by relinking the license for software 1 from computer 2 to computer 1; see Section 2.3.6, “Relink resource” [18].

Bernadette Bossa needs an additional table:

1. Florian Facilitan searches for free tables in the system; see Section 2.3.7, “Create filtered collection of resource entries” [19].
2. He finds table 3 and transports it to Bernadette.
3. He reproduces this in the system by linking the table to Bernadette's workingplace; see Section 2.3.4, “Add resource link” [17].

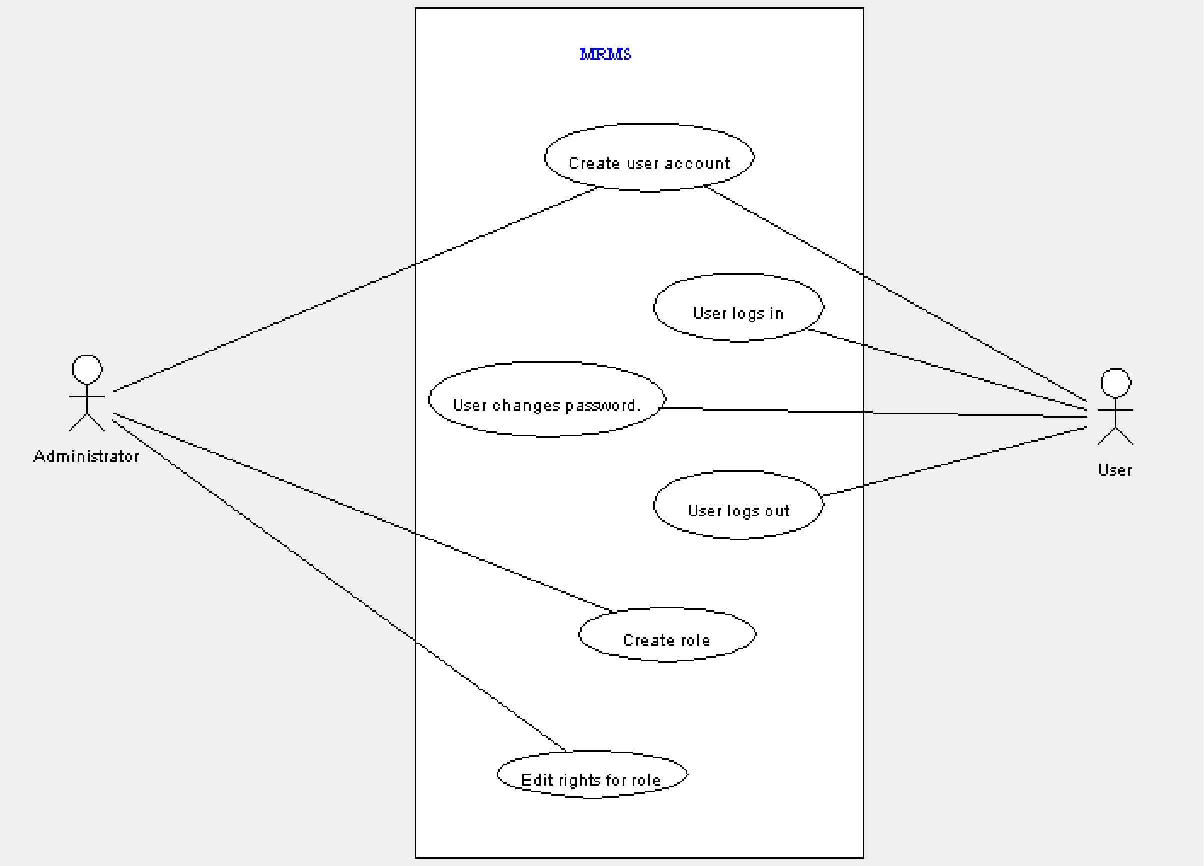
2. Use Cases

These are the use cases derived from the mandatory functionality described in the requirements specification. We have grouped them to their purpose: user- , meta data- and data management.

2.1. User Management

These are the use cases needed for user management. Please note, that the administrator in the overview diagram is also a user, but one with special access rights. So, actually, the administrator must have been connected to all use cases the user is connected to (for example “user logs in”), but in order to present a more concise overview we decided to connect him only to those use cases where he acts a special part.

Figure 2. Overview User Management UCs



2.1.1. User logs in

Goal	The user starts a new working session.
Category	Primary
External Actors	User
Precondition	The user has an account and is not yet logged in.
Triggering Event	The user starts/accesses the software.
Postcondition Success	The user is logged into the system and is able to work with it.
Postcondition Failure	The user must not be able to work with the system.
Description	

1. The system requests an identification and a password from the user.
2. The user enters his identification and password and submits them.

3. The system checks both and logs the user in.

Extensions

- **3a.** If identification and/or password are not valid, the system informs the user about this and requests him to correct it.
- **3b.** If the password is marked 'expired' , the use case Section 2.1.3, “User changes password” [8] is triggered.

Alternatives	---
Additional Requirements	---
Annotation	---

2.1.2. User logs out

Goal	The user ends his working session.
Category	Primary
External Actors	User
Precondition	The user is logged into the system.
Triggering Event	The user requests the system to log out.
Postcondition Success	The user is logged out and must not be able to work with the system without logging in again.
Postcondition Failure	The user remains logged in and is able to work with the system.

Description

1. The system requests the user to confirm logout.
2. The user confirms the logout action.
3. The system logs the user out.

Extensions	---
Alternatives	---
Additional Requirements	---
Annotation	The user is logged out automatically after a specific time of inactivity.

2.1.3. User changes password

Goal	The password for a user of the system changes.
Category	Primary
External Actors	User

Precondition The user is logged into the system.

Triggering Event The user (1) requests the system to set a new password for his account or (2) he must set it because his password is marked 'expired'.

Postcondition Success The new password has been set according to the user's input.

Postcondition Failure The user's password has not been changed.

Description

1. The system requests the user to change the password for his account.
2. The user enters his current password once and his new one twice and submits his input.
3. The system changes sets the users new password.

Extensions

- **3a.** If the current password is incorrect, the user is prompted to correct it.
- **3b.** If the two input values for the new password are not equal, the user is prompted to correct it.
- **3c.** If the new user enters a password that is too weak (less than six characters), he is prompted to enter a stronger password.

Alternatives ---

Additional Requirements ---

Annotation ---

2.1.4. Create user account

Goal A new user account is created for an employee.

Category Primary

External Actors Administrator

Precondition A user is logged in who has administration access, i.e. who is an administrator.

Triggering Event The administrator requests the system to create a new user account.

Postcondition Success A new user account been created according to the administrators input.

Postcondition Failure No new user account has been created that corresponds to the administrators input.

Description

1. The system requests the administrator to enter the users identification along with other information for the new account.
2. The administrator enters the information for the new user account and submits his input.
3. The system creates a new user accout with the given information and marks

the user accounts password 'expired'.

Extensions	**3a.** If the administrator enters a user name, that is already in use, the system requests him to choose another name and goes back to step 2.
Alternatives	---
Additional Requirements	---
Annotation	---

2.1.5. Edit role assignment of user account

Goal	(1) Because of a change in responsibility, a users role assigment changes. (2) Roles are initially assigned to a new user account.
Category	Primary
External Actors	Administrator
Precondition	A user is logged in who has administration access, i.e. who is an administrator.
Triggering Event	The administrator requests the system to edit role assignment of a specific user account.
Postcondition Success	The assigned roles of a user have been changed according to the administrator's input.
Postcondition Failure	The assigned roles of a user have not been changed corresponding the the administrators input.
Description	1. The system loads current role assignment and requests the administrator to change it. 2. The administrator selects or deselects entries from a collection of all possible roles and submits his input. 3. The system updates the users role assignment.
Extensions	---
Alternatives	---
Additional Requirements	---
Annotation	It is possible that a user account has not assigned any roles. In this case the user has no access rights.

2.1.6. Create role

Goal	A new role is identified and created, because (1) the existing access rights should be differentiated or (2) there is a new resource type which needs to be managed with independent access rights from already created roles.

Category	Primary
External Actors	Administrator
Precondition	A user is logged in who has administration access, i.e. who is an administrator.
Triggering Event	The administator requests the system to create a new role.
Postcondition Success	A new role has been created according to the administrator's input.
Postcondition Failure	No new role has been created.
Description	

1. The system requests the administrator to enter a name for the new role.
2. He enters the role's name and submits his input.
3. The system creates the role with all access rights unset.
4. The administrator sets the access rights for this role; see Section 2.1.7, “Edit rights for role” [11].

Extensions	---
Alternatives	---
Additional Requirements	---
Annotation	---

2.1.7. Edit rights for role

Goal	(1) The rights for a specific role are changed because of changes in the business' set-up or (2) an initial setting for a newly created role is made.
Category	Primary
External Actors	Administrator
Precondition	A user is logged in who has administration access, i.e. who is an administrator.
Triggering Event	The administrator requests the system to set rights for a specific role.
Postcondition Success	The rights for the role have been set according to the administrator's input.
Postcondition Failure	The rights for the role have not been changed.
Description	

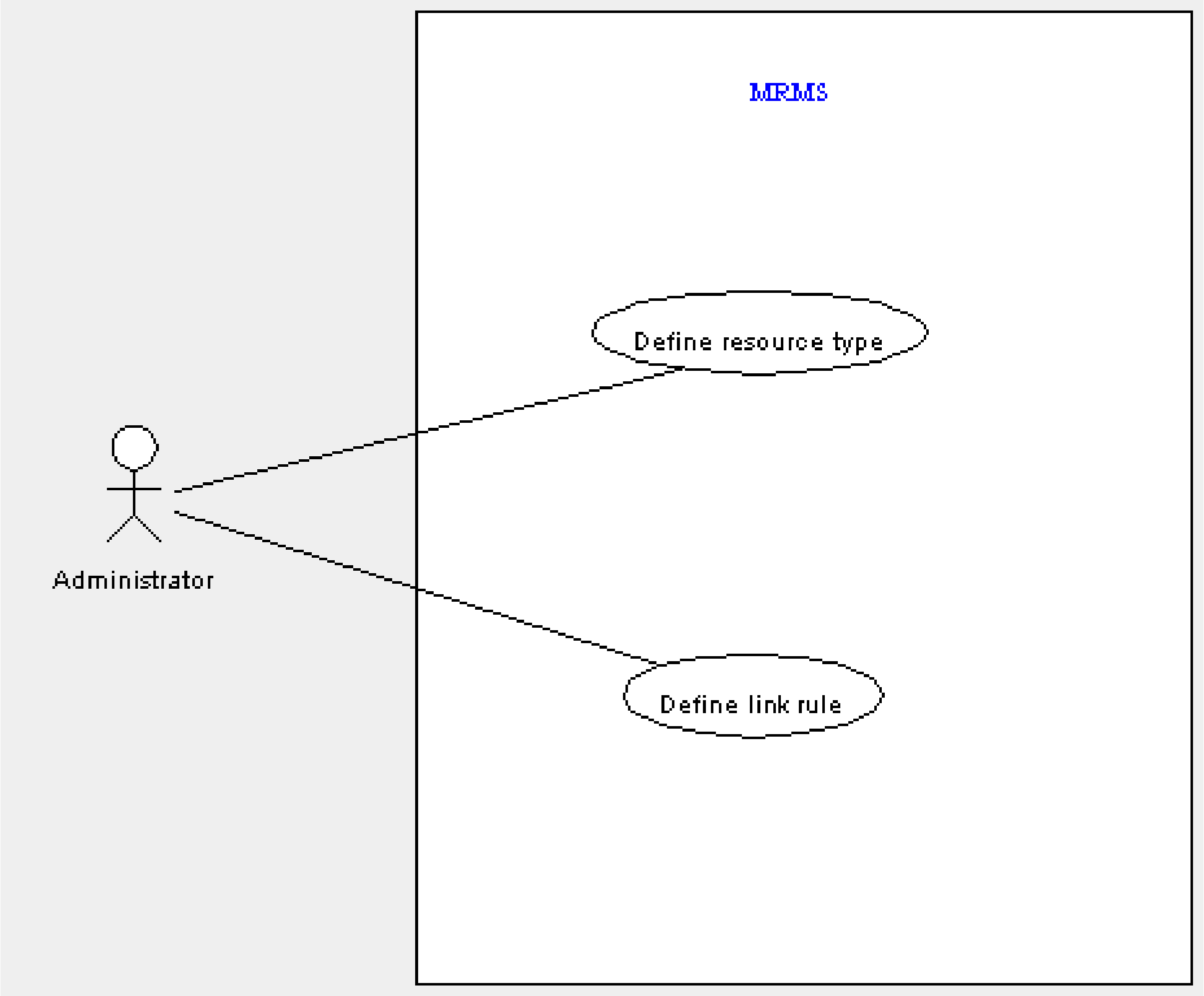
1. The system requests the administrator to edit the access rights.
2. For every resource type he selects if creation and/or deletion or none of them is allowed.
3. For every attribute of every resource type he selects if reading and/or writing or none of them is allowed.
4. For every link rule he selects if linking and/or unlinking or none of them is

	allowed. 5. He submits his input. 6. The system updates the roles rights assignment.
Extensions	---
Alternatives	---
Additional Requirements	---
Annotation	---

2.2. Meta Data Management

These use cases have been identified for managing resource meta data.

Figure 3. Overview Meta Data Management UCs



2.2.1. Define resource type

Goal	A new type of resources that should be managed by the system has been identified and is created.
Category	Primary
External Actors	Administrator
Precondition	A user is logged in who has administration access, i.e. who is an administrator.
Triggering Event	The administrator requests the system to define a new resource type.
Postcondition Success	A new resource type has been created according to the administrator's input.
Postcondition Failure	No new resource type has been created.
Description	

1. The system requests the administrator to determine a name for the resource type.
2. The administrator enters the name for the resource type.
3. The system checks the name and requests the administrator to define a set of attributes for the resource type.
4. For every attribute to define, he determines a name, typtheme and if it's mandatory or optional; after that, he submits the complete set.
5. A new type of resources is created by the system using the information entered by the administrator.

Extensions	**3a.** If the administrator enters a resource type name, that is already in use, the system requests him to choose another name and goes back to step 2.
Alternatives	---
Additional Requirements	--
Annotation	At least one attribute must be added to a resource type.

2.2.2. Define link rule

Goal	Determine that two resource types can be linked together and specify the corresponding cardinalities.
Category	Primary
External Actors	Administrator
Precondition	A user is logged in who has administration access, i.e. who is an administrator.
Triggering Event	The user requests the system to define a new link rule for a specific resource type.
Postcondition Success	The link rule has been defined according to the administrator's input.
Postcondition Failure	No new link rule has been created corresponding to the administrators input.
Description	

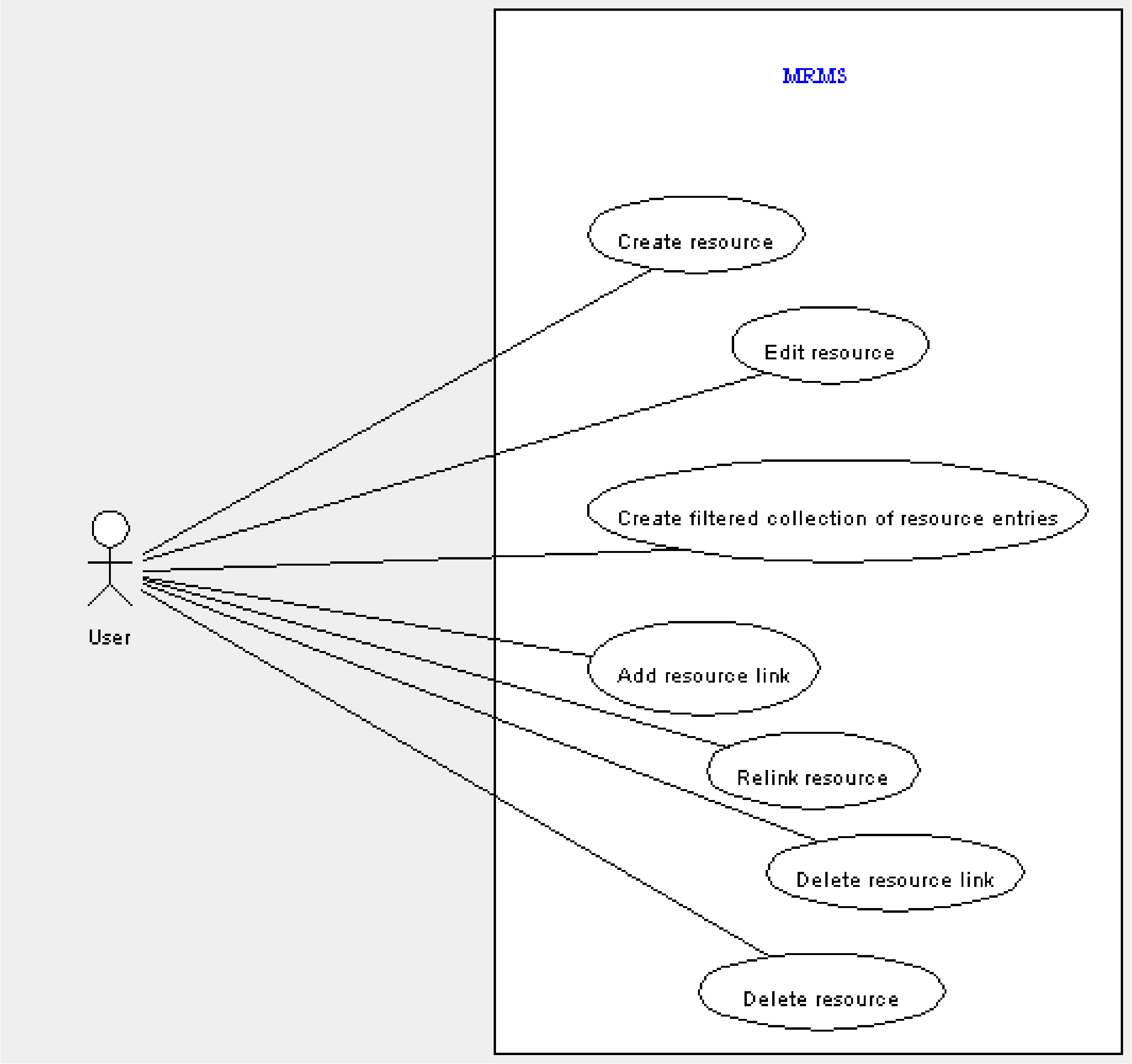
1. The system requests the user to configure the new link rule.

	2. The user chooses two resource types that may be linked, assigns both a cardinality and submits his input. 3. With the information given, a new link rule for the resource type is created by the system.
Extensions	**3a.** If the configuration entered is identical with that of an already existing link rule, the system requests the user to change his configuration and goes back to step 2.
Alternatives	---
Additional Requirements	---
Annotation	---

2.3. Data Management

The following use cases are needed for resource data management.

Figure 4. Data Management UCs



2.3.1. Create resource

Goal	A new instance of a resource type is created, that should be managed by the system.
Category	Primary
External Actors	User
Precondition	A user is logged in who has proper access rights to create the new resource.
Triggering Event	The user requests the system to create a new resource.
Postcondition Success	A new resource has been created according to the users input.
Postcondition Failure	No new resource has been created.
Description	

1. The system requests the user to choose type and unique name for the resource to create.

2. The user determines type and unique name and submits his input.
3. The system loads the attribute set definition for the given resource type and requests the user to enter values.
4. He enters values for all attributes the resource type defines and submits them.
5. The system creates a new resource of the given type and with the attribute values given by the user.

Extensions	• **5a.** If not all mandatory attributes have been given a value, the user is prompted to correct that in step 4.
Alternatives	---
Additional Requirements	---
Annotation	---

2.3.2. Delete resource

Goal	The resource does not exist in the business any longer and is deleted from the system.
Category	Primary
External Actors	User
Precondition	A user is logged in who has proper access rights to delete the resource.
Triggering Event	The user requests the system to delete a specific resource.
Postcondition Success	The resource has been deleted from the system.
Postcondition Failure	The resource still exists in the system and no changings have been made to its attributes.
Description	1. The system requests the user to confirm deletion. 2. The user confirms deletion of the resource. 3. The system deletes the resource.
Extensions	---
Alternatives	• **1a.** If there are resources that are linked to the resource that should be deleted, the system asks the user if they should be deleted as well or if just the link to them should be deleted. If he decides for deletion and these resources are linked as well he will be asked the same question again, and so forth.
Additional Requirements	---

Annotation	---

2.3.3. Edit resource

Goal	Real world information about a resource has changed and is updated in the system.
Category	Primary
External Actors	User
Precondition	A user is logged in who has proper access rights for editing the resource.
Triggering Event	The user requests the system to edit a specific resource.
Postcondition Success	The resource has been modified according to the user's input.
Postcondition Failure	The resource has not been modified.
Description	1. The system requests the user to edit attributes of the resource. 2. The user performs desired changes on attribute values and submits his input. 3. The system updates the resource.
Extensions	• **3a.** If the input mask does not contain values for all mandatory attributes, the user is prompted to correct it in step 2.
Alternatives	---
Additional Requirements	---
Annotation	---

2.3.4. Add resource link

Goal	A resource's real world linkage has been extended and is updated in the system.
Category	Primary
External Actors	User
Precondition	A user is logged in who has proper access rights for adding the link.
Triggering Event	The user has selected one resource as “link source” and requests the system to add a link.
Postcondition Success	A new link has been created between the specified resources.
Postcondition Failure	No resource link has been created between the specified resources.
Description	1. The system requests the user to select a resource type for the “link target” out

of a collection of possible options.

2. The user selects a resource type for the “link target”.
3. According to the selection, the system offers a collection of resources that would be valid “link targets”.
4. The user selects one “link target” and submits his input.
5. The system creates the resource link between the specified resources.

Extensions	---
Alternatives	---
Additional Requirements	---
Annotation	The list of possible resource types for the “link target” in step 1 is determined according to the link rules: a resource type t1 will only be displayed if it appears together with the resource type of the “link source” in one of the link rules and if the “link source” has less links to t1-resources than the maximum cardinality allows.

2.3.5. Delete resource link

Goal	A resource's real world linkage is no longer existent and is removed from the system.
Category	Primary
External Actors	User
Precondition	A user is logged in who has access rights allowing him to delete the resource link.
Triggering Event	The user requests the system to delete a specific resource link.
Postcondition Success	The link has been deleted.
Postcondition Failure	The link has not been deleted.
Description	

1. The system requests the user to confirm the deletion.
2. The user confirms the deletion.
3. The system deletes the resource link.

Extensions	---
Alternatives	---
Additional Requirements	---
Annotation	---

2.3.6. Relink resource

Goal	A link end r2 of a resource r1 should be substituted with another resource r3 (with same resource type as r2).
Category	Secondary
External Actors	User
Precondition	A user is logged in who has proper access rights for unlinking and linking the participating resource types; r1 must be linked with at least one other resource.
Triggering Event	The user requests the system to substitute a link end of a resource r1.
Postcondition Success	R1 and r2 are not linked, r1 and r3 are linked.
Postcondition Failure	R1 and r2 are linked, r1 and r3 and not linked.

Description

1. The system requests the user to choose a link end r2 to be substituted.
2. The user selects the “link source” r1 and the “link target” r2 he wants to substitute .
3. The system offers a collection of possible alternative “link targets” of the same type as r2 and requests the user to choose one.
4. The user selects the resource r3 and submits his input.
5. The system updates the linkage.

Extensions	---
Alternatives	---
Additional Requirements	---
Annotation	---

2.3.7. Create filtered collection of resource entries

Goal	Collect a set of resources meeting a specific criterion and offer it to the user for further processing.
Category	Secondary
External Actors	User
Precondition	A user is logged in.
Triggering Event	The user requests the system to create a filtered collection of resources.
Postcondition Success	The user is shown a collection of resources that passed the filter he created.
Postcondition Failure	---

Description

1. The system requests the user to configure a filter listed resources will have to pass.
2. The user chooses resource types and resource states that may pass the filter

(consistent, linkable, free).

3. The system collects all resources passing the specified filter and offers them to the user for further processing.

Extensions

- **2a.** If he has selected only one resource type he may additionally enter filter rules regarding the attributes' values.

Alternatives ---

Additional Requirements ---

Annotation ---

Glossary

Free resource

A *resource* is called “free” if it has not got any links to any other resource; also called unlinked resource. Attend the difference to a *linkable resource*.

Link

The MRMS allows resources to be linked together according to the defined *link rules*. Every link is undirected and affects two resources.

Link end

If two resources are *linked*, they are called link ends.

Link rule

A link rule consists of two *resource types* whose instances - the *resources* - can be *linked* together. The link rule also specifies cardinalities which may either be fix numbers or attribute-values of the linked resources. Take the “room-workplace” link rule from Section 1, “Usage scenario” [1] as an example. Resource type 1 is “room”, cardinality 1 is “1”, resource type 2 is “workplace”, cardinality 2 is “0..room.maxWorkplace”. You can read this as follows: One workplace must be linked to exactly one room (cardinality 1); one room is linked to zero up to room.maxWorkplace (the value of the room's attribute maxWorkplace) workplaces (cardinality 2).

Link source

Although the *links* in the MRMS are undirected we use the terms link source and link target. When the user navigates along the links through the system, we call the resource he is currently using/editing/viewing the link source and all resources that are linked with this link source are the link targets.

Link target

See *link source*.

Linkable resource

A *resource* r1 of *resource type* t1 is called “linkable” in terms of another resource r2 of resource type t2 if r1has less *links* to t2-resources than the cardinality of the link rule for t1-t2 allows. Attend the difference to a *free resource*.

Resource

The term resource can have two meanings: (1) a real object that is needed in a business to produce the business' products or services, e.g. computers, white boards, rooms, cars, etc. and (2) the virtual representation of a resource (in the first meaning) in the MRMS.

A resource in the latter meaning has attributes that are defined by its *resource type*. In this sense a resource is an instance of a resource type. Resources can be *linked* together according to the *link rules* defined in the MRMS.

Unlinked resource

See *free resource*.