Jan Rehwaldt, 2012-04-25, University of Tartu

Improvement of Exercise 4, Principles of Secure Software Design

**SecureUML model**

<<secuml.user>>

**Max**

<<secuml.roles>>

**Bank Employee**

<<secuml.user>>

**Moritz**

<<secuml.user>>

**Erni**

<<secuml.roles>>

**User**

<<secuml.resource>>

**Account**

- addMoney()

- removeMoney()

- checkBalance()

- changeMoneyOwner()

- money

- administrator

1..\*

- owner

1

- userAccount

- account

1

<<assignment>>

<<assignment>>

<<assignment>>

<<secuml.permission>>

**EmployeePermission**

- changeMoneyOwner : Change

- checkBalance : View

<<secuml.permission>>

**UserPermission**

- checkBalance : View

- transferMoneyToAccount : Add

- transferMoneyFromAccount : Remove

**UMLsec transformation**

**User**

Bank Employee

Account

transferMoneyToAccount

addMoney

checkBalance

transferMoneyFromAccount

removeMoney

checkBalance

changeMoneyOwner

changeMoneyOwner

authorized change requested

no change requested

<<rbac>>

The connections between different tasks had to be modeled manually. Additionally is the mapping of Permission to Action on Resource not automatically transformable in this model if their naming is not equal.

{**protected** = (addMoney)}

{**right** = (User, addMoney)}

{**role** = (Moritz, User)}

{**protected** = (checkBalance)}

{**right** = (User, checkBalance)}

{**role** = (Moritz, User)}

{**protected** = (removeMoney)}

{**right** = (User, removeMoney)}

{**role** = (Moritz, User)}

{**protected** = (addMoney)}

{**right** = (User, addMoney)}

{**role** = (Erni, User)}

{**protected** = (checkBalance)}

{**right** = (User, checkBalance)}

{**role** = (Erni, User)}

{**protected** = (removeMoney)}

{**right** = (User, removeMoney)}

{**role** = (Erni, User)}

{**protected** = (checkBalance)}

{**right** = (Bank Employee, checkBalance)}

{**role** = (Max, Bank Employee)}

{**protected** = (changeMoneyOwner)}

{**right** = (Bank Employee, changeMoneyOwner)}

{**role** = (Max, Bank Employee)}