Information Systems Security Engineering

Exercise Feedback

Prof. Dr. Raimundas Matulevičius

University of Tartu
email: rma@ut.ee
To guide your solution, consider

- What are objects and their concerned attributes?
- What operations do change the values of the attributes?
- What are roles?
- What are the security actions?
- What are the permissions of roles towards the object?
- Who are the users?
To guide your solution, consider:

- What are objects and their concerned attributes?
- What operations do change the values of the attributes?
- What are roles?
- What are the security actions?
- What are the permissions of roles towards the object?
- Who are the users?
To guide your solution, consider:

- What are objects and their concerned attributes?
- What operations do change the values of the attributes?
- What are roles?
- What are the security actions?
- What are the permissions of roles towards the object?
- Who are the users?
To guide your solution, consider
• What are objects and their concerned attributes?
• What operations do change the values of the attributes?
• What are roles?
• What are the security actions?
• What are the permissions of roles towards the object?
• Who are the users?
To guide your solution, consider:

- What are objects and their concerned attributes?
- What operations do change the values of the attributes?
- What are roles?
- What are the security actions?
- What are the permissions of roles towards the object?
- Who are the users?
AC#1: Attribute `makeReport` is associated to the `GameReport` operation `createWithInitialInformation()`, which **Insert** new `GameReport` (new attributes `leagueName`, `secretaryContactData`, `region`, `division`, `gameNumber1`, `gameNumber2`, `teamHomeName1`, `teamHomeName2`).

AC#2: Attribute `confirmReport` is associated to the `GameReport` operation `confirm()`, which **Update** `GameReport` attribute `confirmation`.

AC#3: Attribute `signAndComment` is associated to the `GameReport` operation `signByTeamRep()`, which **Update** `GameReport` attributes `signature1` and `comments1` if TeamRepresentative.home is true. If TeamRepresentative.away is true then, `GameReport` attributes `signature1` and `comments2` are **Update**.

AC#4: Attribute `makeTeam` is associated to the `Team` operation `createTeam()`, which **Insert** new `Team` (new attributes `name`).

AC#5: Attribute `fillScores` is associated to the `Team` operation `fillResults()`, which **Update** `Team` attributes `finalScore`, `fullScore`, `extraTimeScore`, `penalties`.

AC#6: Attribute `makePlayer` is associated to the `Player` operation `createPlayer()`, which **Insert** new `Player` (new attributes `name`, `registrationNumber`).

AC#7: Attribute `signAndComment` is associated to the `GameReport` operation `signByUmpire()`, which **Update** `GameReport` attributes `umpireSignature` and `umpireComments`.

AC#8: Attribute `giveYellowCard` is associated to the `Player` operation `giveYellowCards()`, which **Update** `Player` attributes `yellowCards`.

AC#9: Attribute `giveRedCard` is associated to the `Player` operation `giveRedCards()`, which **Update** `Player` attributes `redCard`.

AC#10: Attribute `giveGoal` is associated to the `Player` operation `giveGoal()`, which **Update** `Player` attributes `goals`.

AC#11: Execute `createTeam()`, if t<2h (where t is game starting time).

- **What authorisation constraints should be defined in your model?**
  - They can be defined in any **formal**, **semiformal** or **natural** language (in case of natural language, use English 😊)
To guide your solution, consider
• What are the objects
• What are their operations?
• What are the roles?
• What are the role’s rights?
• What are the associated tags?
• Who are the users?
To guide your solution, consider:

- What are the objects?
- What are their operations?
- What are the roles?
- What are the role’s rights?
- What are the associated tags?
- Who are the users?
To guide your solution, consider:

- What are the objects?
- What are their operations?
- What are the roles?
- What are the role’s rights?
- What are the associated tags?
- Who are the users?
To guide your solution, consider
• What are the objects
• What are their operations?
• What are the roles?
• What are the role’s rights?
• What are the associated tags?
• Who are the users?

{\textbf{protected} = protected\_action}

{\textbf{role} = (actor, role)}

{\textbf{right} = (role, protected\_action)}
<table>
<thead>
<tr>
<th>Role</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calex Karuuna, LeagueSecretary</td>
<td>createWithInitialInformation</td>
</tr>
<tr>
<td>LeagueSecretary</td>
<td>MakeReport</td>
</tr>
<tr>
<td>(Home, Away), TeamRepresentative</td>
<td>CreateTeam</td>
</tr>
<tr>
<td>TeamRepresentative</td>
<td>CreatePlayer</td>
</tr>
<tr>
<td>Team</td>
<td>FillResults</td>
</tr>
<tr>
<td>TeamRepresentative</td>
<td>SignByTeamRep</td>
</tr>
<tr>
<td>TeamRepresentative</td>
<td>SignByUmpire</td>
</tr>
<tr>
<td>Umpire</td>
<td>FillScores</td>
</tr>
<tr>
<td>Umpire</td>
<td>RegisterPlayerPerformance</td>
</tr>
</tbody>
</table>

**Roles and Rights:***

- **CreateWithInitialInformation**: Created by Calex Karuuna, LeagueSecretary.
- **CreateTeam**: Rights include TeamRepresentative, CreateTeam.
- **CreatePlayer**: Rights include TeamRepresentative, CreatePlayer.
- **RegisterPlayerPerformance**: Rights include Umpire, RegisterPlayerPerformance.
- **FillResults**: Rights include Umpire, FillResults.
- **SignByTeamRep**: Rights include TeamRepresentative, SignByTeamRep.
- **SignByUmpire**: Rights include Umpire, SignByUmpire.
- **Confirm**: Rights include LeagueSecretary, Confirm.
Task 3

Explain what RBAC policy concerns are captured in your...

... **SecureUML** model and *not* in the **UMLsec** model

... **UMLsec** model and *not* in the **SecureUML** model
## Language comparison

### Construct Semantics

<table>
<thead>
<tr>
<th>RBAC concepts</th>
<th>SecureUML</th>
<th>UMLsec</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Class stereotype «secuml.user»</td>
<td>Actor value of association tag {role}</td>
</tr>
<tr>
<td>User assignment</td>
<td>Dependency stereotype «assignment»</td>
<td>Associated tag {role}</td>
</tr>
<tr>
<td>Roles</td>
<td>Class stereotype «secuml.role»</td>
<td>Activity partition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Role value of association tag {role}</td>
</tr>
<tr>
<td>Permission assignment</td>
<td>Association class stereotype «secuml.permission»</td>
<td>Action</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Associated tag {right}</td>
</tr>
<tr>
<td>Object</td>
<td>Class stereotype «secuml.resource»</td>
<td>Activity partition</td>
</tr>
<tr>
<td>Operation</td>
<td>Operation of «secuml.resource» class</td>
<td>Action</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Associated tag {protected}</td>
</tr>
<tr>
<td>Permission</td>
<td>Authorisation constraints</td>
<td>Not defined</td>
</tr>
</tbody>
</table>