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?

```plaintext
{protected = protected_action}
{role = (actor, role)}
{right = (role, protected_action)}
```
{protected = createWithInitialInformation}
{role = (Calex Karuuna, LeagueSecretary)}
{right = (LeagueSecretary, createWithInitialInformation)}

{protected = CreateTeam}
{role = ((Home, Away), TeamRepresentative)}
{right = (TeamRepresentative, CreateTeam)}

{protected = CreatePlayer}
{role = ((Home, Away), TeamRepresentative)}
{right = (TeamRepresentative, CreatePlayer)}

{protected = RegisterPlayerPerformance}
{role = (Billy-Goat the Beard, Umpire)}
{right = (Umpire, RegisterPlayerPerformance)}

{protected = FillResults}
{role = (Billy-Goat the Beard, Umpire)}
{right = (Umpire, FillResults)}

{protected = SignByTeamRep}
{role = ((Home, Away), TeamRepresentative)}
{right = (TeamRepresentative, SignByTeamRep)}

{protected = SignByUmpire}
{role = (Billy-Goat the Beard, Umpire)}
{right = (Umpire, SignByUmpire)}

{protected = Confirm}
{role = (Calex Karuuna, LeagueSecretary)}
{right = (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&lt;br&gt;➢ Role value of association tag {role}</td>
</tr>
<tr>
<td>Permission assignment</td>
<td>Association class stereotype «secuml.permission»</td>
<td>➢ Action&lt;br&gt;➢ 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&lt;br&gt;➢ Associated tag {protected}</td>
</tr>
<tr>
<td>Permission</td>
<td>Authorisation constraints</td>
<td>Not defined</td>
</tr>
</tbody>
</table>

© Springer International Publishing AG 2017