Tutorial: Learn to build UML profiles

Ansgar Radermacher, Jérémie Tatibouët, Shuai Li, Patrick Tessier and François Terrier

Lab 1 – A profile to express relational databases

Objectives:

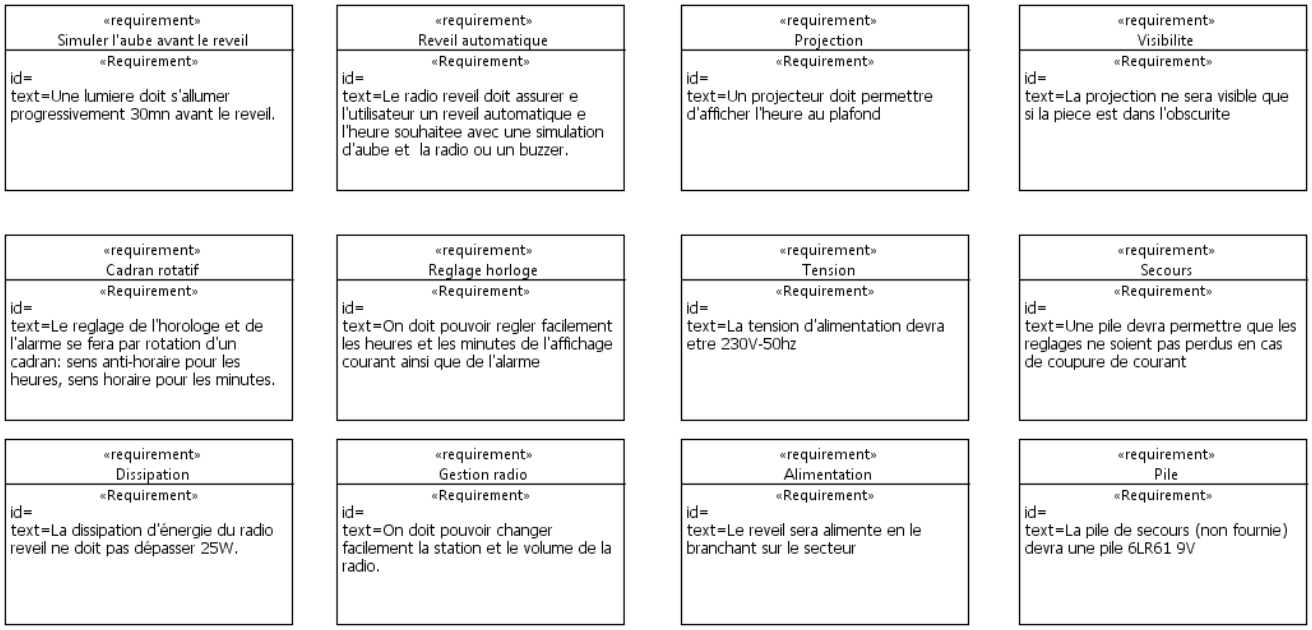
- Build a profile
- Apply it on a model

Exercise:

1. Create a profile to express relational databases
   A. Identify the concepts that you want to add to UML (Table, Primary key, etc.)
   B. Identify the meta-classes that will be extended by the concepts newly introduced
   C. Implement the profile within Papyrus
      - File -> New -> Papyrus Project -> Select "SW engineering – profile" as architectural context
2. Customize the notation associated to the stereotypes
   A. Associate each of your stereotype with a dedicated icon. This icon will be displayed when you will use your stereotypes on a model. Note: icons can be found at https://www.iconfinder.com/.
3. Use your profile on a database model
   A. Import the model that can be found in the subfolder **"Lab1-Profile-For-RelationalDatabases"**
   B. Apply your profile on the imported model
      - Click on the root element of your model
      - In the property view select the "profile" tab
      - Click on the "+" button and reference the profile that is in your workspace.
   C. Complete the database model (using your stereotypes) to satisfy the following requirements
      - A customer is identified by an "ID"
      - A customer can be client in zero or many banks
      - A bank handles zero or many account
      - A Bank is identified by an "ID"
      - A customer is the owner of a specific account
      - A account has only one owner
      - A customer can realize transactions.
      - A transaction takes place at a specific date.
      - A transaction is related to a specific account.
      - **Note: If you cannot fulfill all the requirements, this probably means you need to refine your profile.**

Lab 2 – Requirements modeling with SysML v1

REQUIREMENTS OF A RADIO ALARM CLOCK



Exercise (using Papyrus)

1. Create an AlarmClock SysML project
2. Add a package called “requirements”
3. Create a requirements diagram
4. Identify composite requirements and decompose them into elementary requirements. Think about composition links.
5. Add derivation relationships if needed.

Expected outcome

The report (as a PDF) and the model should be send to ansgar.radermacher@cea.fr. Please zip your report and your model in an archive **“FIRSTNAME-LASTNAME.zip”**