

Tagging guidelines: conceptual models

This document explains the process for tagging alignment and discrepancies between the *static* conceptual models created by the students and those of the domain experts, starting from the requirements that the students had produced.

File structure. There is one folder per tagger, each containing a master tagging file called `tagging-NAME.xlsx`, with NAME being the tagger name. Within each folder, there is one sub-folder per requirements specification (and its conceptual model). For example, the folder `fabiano/g02-uc` contains the files related to the group member of group 02 who used use cases as specification language. Within each of these subfolders, there are two or three files:

- The conceptual model created by the domain experts, e.g., `g02-UC.uxf`
- The requirements and the conceptual model.
 - a. If both are available in assignment 2, a single file is shown, e.g., `g02-uc-req-cm.pdf`
 - b. If the requirements are in assignment 1 only, two files are shown, one for the conceptual model, one for the requirements, e.g., `g02-uc-req.pdf` and `g02-uc-cm.pdf`

Tagging process. The tagging is performed by working on multiple files:

- Highlighting classes in the conceptual model created by the domain expert, in UMLet.
- Highlighting classes, attributes, methods, and relationships in the conceptual model by the student, using a PDF viewer.
- Adding descriptive statistics to the Excel file in the parent folder.

Multiple situations can be encountered:

Situation	Description	Actions		
		CM-Stud	CM-Expert	Excel
Alignment	A concept is represented as a class in both models, either with the same name or using synonyms or clearly linkable names.	Green highlight class	Green highlight class	+1 Alignment
Wrong representation	A class in the domain expert model is incorrectly represented in the student model, either • via an attribute, method, or relationship rather than class, or • using a generic term (e.g., “user” instead of “urban planner”).	Yellow highlight attribute, method, or relationship	Yellow highlight class	+1 Wrong-Rep
System-orientation	A class in CM-Stud that denotes a technical implementation aspect, e.g., access control. Classes that represent legacy system or the system under design (portal, simulator) are legitimate, thus green.	Orange highlight class	-	+1 System-Or
Omitted	A class in CM-Expert that does not appear in any way in CM-Stud	-	Red highlight class	+1 Omitted
Missing	A class in CM-Stud that does not appear in any way in CM-Expert	-	-	+1 Missing Short explanation

Finally, an open field is available to add a few notes on the type of relationships that the student has employed, e.g., whether they are labelled, used correctly, and if they reflect the actions in the requirements.