

Software Engineering Workshop

Workshop 1: Introduction to the UML

Objective

The objective of today's session is to get an introduction to the **Unified Modeling Language (UML)**, which is one of the most exciting and useful tools in the world of *system development*. Through the provided materials in the workshop and the exercises in this sheet, you will 1) Develop an appreciation for the importance of UML in the system development process, 2) Be aware of the origin of the UML (a bit of history here), 3) Learn about the different types of diagrams offered in the language.

*Please revise the slides and your notes before attempting any of the exercises below (5-10 minutes should be enough). Note that some of the questions might be beyond what is given to you in the slides and would rather need you to employ your own **thinking**.*

Exercise 1:

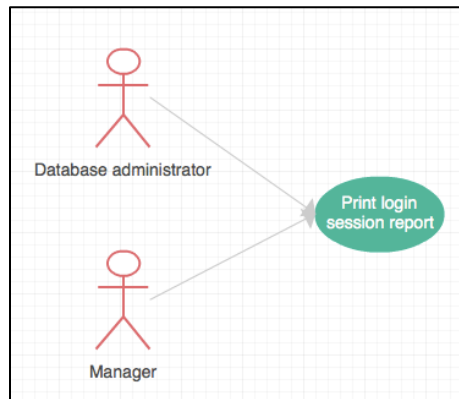
a) In the following requirement specification, identify the different possible interpretations of the written text (if any). What does that tell us? How could it be solved/avoided?

*“The system shall print a login session report to every Manager and Database Administrator”.*¹

Answer:

¹ More examples of ambiguity in natural languages can be found in the following thesis: https://cs.uwaterloo.ca/~dberry/FTP_SITE/students.theses/TjongThesis.pdf

With the same specification being visualized using UML notations shown below, would the same ambiguity remain?



Answer

b) The following is a requirement for creating a question for an online brain game:

“From any word related to software testing, create another word randomly by adding 3 letters. As a result, the following is one of the online questions created by the application:

Cross out 3 letters from “bcdecdeug” so that a familiar software testing vocabulary word will remain”.²

Is “debug” accepted? How about “bug”? Or both? What was ambiguous? Again, what does that tell us?

Answer

² <http://extremesoftwaretesting.com/Humor/AmbiguousRequirements.html>