



University
of Windsor

School of Computer Science
<https://cs.uwindsor.ca>

Master of Applied Computing

COMP-8117

Advanced Software Engineering Topics

Dr. Aznam YACOUB – <https://www.y-az.one>

519-253-3000 (ext. 3781) - aznam@uwindsor.ca

Office Hours : Thursday & Friday (9.00am – 12.00pm EST)

Lab 1 : Specifications (1%)

Goal

The goal of this lab is to give you a methodology and techniques to describe and define specifications of a software project. The system you will study in this lab is composed of 2 parts : the real system and a simulator. Your work will consist on describe these two parts using both informal and formal specifications. We propose the following working path to succeed in this lab :

- Week 1 : Training (Non-Evaluated)
- Week 2 : Mandatory Part

Scenario

Toto Inc., a company directed by Mrs Sonya S., wants to develop and commercialize a new kind of **automated bedroom** because they identified problems with the existing ones. First, most of the citizens in western countries, especially in Canada, complain about a **rude awakening** in the morning. Most of them live in a basement apartment, meaning they don't have any windows in their room. Consequently, **they don't see the sun**, which impacts their mood, and they don't really have a good feeling of time when they wake-up. Moreover, some customers would like to have **advices to choose their clothes according to the news of the day, the weather, and the planned events**. 58% declared that they would be happy **that a system plays for them a music** to improve their mood. 70% of people would like also that the bedroom proposes to them a **personalized program of fitness in the morning**, and a **personalized program to rest well in the night**.

Mrs Sonya S. is not a software engineer, nor a system engineer, and she decided to hire you as an expert to advice her development team. She requires you to create a proof of concept (i.e. a simulator), and the final program. As a first step, you decide to work on the specifications of the both software.

Training (Week 1 – Not evaluated)

This part of this work will help you to understand what a specification is. For that, you have to understand the difference between a need, a requirement and a specification.

As a reminder, a need is **an informal expectation** expressed by a customer or a user.

Example : « I would like to move quickly from the point A to B. »

Generally, this need is described using scenarios and use cases.

A requirement is a statement of one functionality/feature/quality fulfilled by the product (software or system) to achieve the need.

Example : « My system must have 4 wheels and allows people to move from A to B. »

A specification is a **written** detailed analysis of the set of requirements and the associated constraints. There is different kind of specifications at different levels (from abstract to concrete level). Specifications systematically organize requirements. There is a lot of classification and specifications may be both formal (mathematical) or informal (natural). Whatever the specification, its goal is to answer one unique question : « What and Why ? »

« What is the software ? What the software does ? What is the system ? What are the components of the system ? What are the functionalities of the system ? Why this software should to that ? »

Generally, we use a bottom-up or top-down approach to go through specifications, but different other strategies exist. Moreover, the level of details depends on the project and the quality depends on the engineer. Keep in mind that a good specification is a clear specification : « when I read a specification, I should be able to imagine exactly the quality/functionality/feature/product », including its external and internal work (from a functional point of view). A specification is therefore a complete description of an object and its functionalities.

1. As a training, we ask you to go outside and pick any object you find in your environment. Try to describe it as accurate as possible. For each object :
 - Describe a need (Why this object is useful) – Use case scenario may help you to understand the need(s)
 - Describe the different aspects of the object which helps to fulfill the needs (What functionalities/qualities of the object are useful for the need)
 - Describe the internal work and the constraints
 - Repeat it for each component of this object
 - When do you decide to stop your description ? Why ?
2. Repeat the question 1 with a piece of software of your choice.
3. Send your specifications to your mentor to get feedbacks.

Mandatory Part (Week 2)

You must propose functional specifications of the system developed by Toto Inc. We remind you that the system is composed by four parts :

- The **operative system** which is the set of elements and components interacting each others to achieve a goal
- The controller which is the **software** which pilots the system (the automated bedroom)
- The user interface which is the interface between a user and the bedroom
- The simulator which simulates this bedroom (don't forget this element has components).

1. Describe the specifications of this system and software as accurate as possible in an informal way.

Tips : Remember that there are **functional requirements and non-functional (technical) requirements and constraints**. Try to describe both of them.

For this lab, there is no specific format to follow. You can use schemes if needed. Your report should be between 10 and 20 pages. Grading will be based on the quality of writing, clarity and completeness (« Are the specifications enough clear to imagine the objects, software and fonctionnalités by reading them ? »). **Remember that you are SOFTWARE ENGINEER, so we have to focus on the software part in your specifications (find the good balance when you specify the other aspects of the system).**

Optional Part (1% extra credit)

2. Formalize your specifications using Use Case Diagrams, Sequence Diagrams, Class Diagrams, and Activity Diagrams. Formalize the algorithms and the mathematical structures.
Remember : You must not describe the implementation.

Methodology

For this work, we propose to use a top-down approach. Start by describing and specify the most obvious elements and decompose them into smaller pieces. For each of these elements, describe the data (what they are, the attributes, ...) and the fonctionnalités they provide, the interactions with others components. You can stop the decomposition when you consider the specifications don't concern anymore your part as software engineer.