# Assignment 1 – Domain Modeling

**Due Date**: (See Slate\ Assignments)
**Date**: Fall, 2018
**Type**: **Group work of 2 students ONLY.**
**Weight:** **10%**

**Summary**

Create a domain model using Visual Paradigm that describes the Coffee Maker system from a domain viewpoint. The Coffee Maker case study is described in Chapter 20, Heuristics and Coffee in (Martin & Martin, 2006). In this assignment you will start with the Visual Paradigm software model that contains a requirements model with a single-use case, Make Coffee. The requirements for the Coffee Maker Simulator system are defined in the Use-Case description (Info Tab).

**Submission checklist**

- ✓ Software Model created using Visual Paradigm containing the following models and diagrams
    a. Domain Model
        i. Overview Domain Diagram
        ii. Minimum two detailed domain model diagrams focused on the main domain concepts, their attributes and their relationships
    b. Glossary: Defining all important terms and their aliases.
    c. Design Model (preliminary): Shows the main classes of the system and their relationships.

- ✓ Assignment report using Word and PDF with the domain model rationale. Ensure your work is presented professionally

**Details**

**Part II (50%) Domain Model**. Create the domain model for the Coffee Maker Simulator system. The model shall contain the elements determined by analyzing the *Make Coffee* use-case of the system and therefore it only needs to contain design elements related to it. See Appendix 2 for the use-case details.

1. Analyze the overview requirements of the *Make Coffee* use-case and identify the nouns representing domain objects and verbs identifying relationships
2. Create an overview class diagram that visualizes all domain concepts and their relationships
    a. Create a *Domain Overview* diagram that identifies all the domain classes and their relationships. Do not show any attributes in the main domain diagram.

b. Ensure all relationships are named using domain terminology
c. Create detailed diagrams for each of the main domain concepts showing the domain concept and its attributes, all its relationships with the rest of the domain elements and their attributes. Identify at least two such detailed domain diagrams.

**Part II (10%) Project Glossary**. Using Visual Paradigm, define the *Glossary* of the system.

1. Define all the domain terms relevant to the use-case analyzed.
2. Define aliases for domain terms that are used in different forms (e.g. app and application)
3. For each term provide a short description of what the term means
4. Define all software terms relevant to the problem being modeled, terms that should be clearly understood by the domain experts.

**Part III (20%) Design Model**. Create the preliminary design model for the Coffee Maker Simulator system based on the domain model defined above. The design model shall only contain design elements related to the *Make coffee* use-case.

1. Assess each domain class to determine if it should be a class in the design model (and the code) or not.
2. Identify classes and abstractions as you believe at this stage to need in the implementation of the system.
3. Create a "Design Model" in the software model and describe the classes and their relationships in the design model using class diagrams. Classes do not need to contain any details at this stage.

**Part IV (20%) Domain Model Rationale**. Using an assignment report, provide evidence and reasoning for your domain modeling choices supporting the claim that the domain model is a "good and complete model".

1. Identify which were the main domain concepts you have highlighted and why
2. Identify specific domain modeling decisions you have taken such as identifying cardinality, direction of relationships
3. Describe the relationship between the domain model concepts and the design model classes you have identified.

**Notes:**

1. The **professionalism of your submission**, clarity of written communication is extremely important. The ability to communicate your knowledge is as important as the knowledge itself. Up to 40% of the mark for any written work can be deducted due to poor presentation / communication: document organization (10%), layout (10%) spelling (10%), title page (10%)

2. **All assignment shall be submitted by the deadline.** Late submissions will be penalized with 10% per day for up to 3 calendar days after which the assignment cannot be submitted anymore. **An**

**email must be sent** should you choose to submit a late assignment. If no such emails are received the solution will be posted. **Assignments are not accepted after the solutions have been posted.**

3. This assignment shall be **a group work**. See the Academic Honesty at Sheridan.

4. Submission is done in electronic format **using SLATE DropBox. DO NOT email your submission.**

# Appendix I: Coffee Maker Use-Case Context Diagram

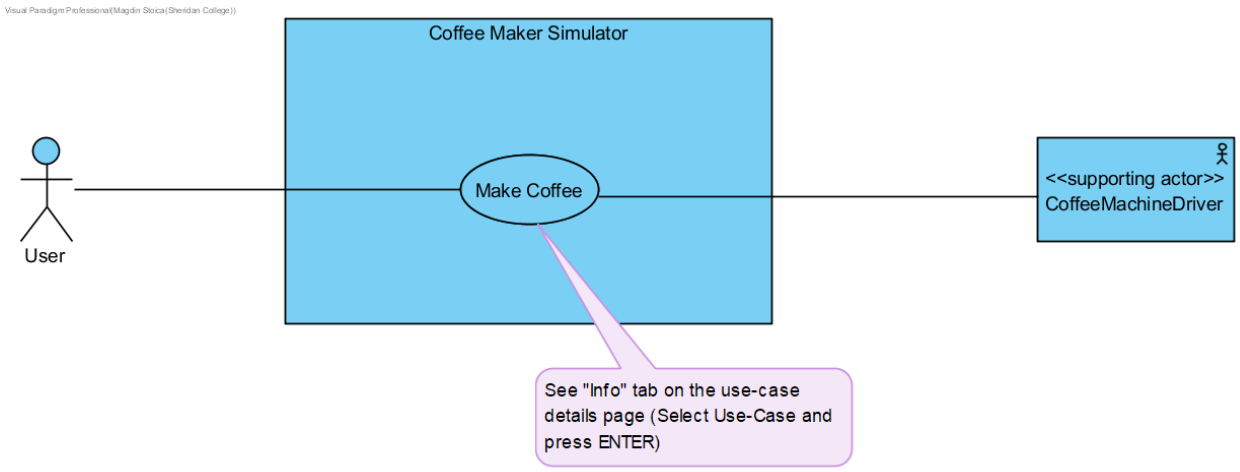


Figure 1: Context Diagram for Coffee Maker Simulator System

Chapter 20: Heuristics and Coffee (Martin & Martin, 2006)

## Appendix 2: Software Requirements for the Coffee Maker Simulator

Below are the software requirements for the Coffee Maker Simulator as defined in Chapter 20:

"The Mark IV Special makes up to 12 cups of coffee at a time. The user places a filter in the filter holder, fills the filter with coffee grounds, and slides the filter holder into its receptacle. The user then pours up to 12 cups of water into the water strainer and presses the Brew button. The water is heated until boiling. The pressure of the evolving steam forces the water to be sprayed over the coffee grounds, and coffee drips through the filter into the pot. The pot is kept warm for extended periods by a warmer plate, which turns on only if coffee is in the pot. If the pot is removed from the warmer plate while water is being sprayed over the grounds, the flow of water is stopped so that brewed coffee does not spill on the warmer plate. The following hardware needs to be monitored or controlled:

- The heating element for the boiler. It can be turned on or off.
- The heating element for the warmer plate. It can be turned on or off.
- The sensor for the warmer plate. It has three states: warmerEmpty, potEmpty, potNotEmpty.
- A sensor for the boiler, which determines whether water is present. It has two states: boilerEmpty or boilerNotEmpty.
- The Brew button. This momentary button starts the brewing cycle. It has an indicator that lights up when the brewing cycle is over and the coffee is ready.
- A pressure-relief valve that opens to reduce the pressure in the boiler. The drop in pressure stops the flow of water to the filter. The value can be opened or closed.

The hardware for the Mark IV has been designed and is currently under development. The hardware engineers have even provided a low-level API for us to use, so we don't have to write any bit-twiddling I/O driver code. The code for these interface functions is shown in Listing 20-1. If this code looks strange to you, keep in mind that it was written by hardware engineers."

(Martin & Martin) Martin, Robert C. and Martin Micah: *Agile Principles, Patterns, and Practices in C#*. Prentice Hall PTG, 20060720. VitalBook file.