



Model Based Programming

ECSE 223 (Section 1)

Question 2

April 21st, 2020 07:00 -- April 23rd, 2020 07:00

EXAMINER: **Dániel Varró**
<daniel.varro@mcgill.ca>

ASSOC. EXAMINER: **Shane McIntosh**
<shane.mcintosh@mcgill.ca>

STUDENT NAME:		McGILL ID:											
---------------	--	------------	--	--	--	--	--	--	--	--	--	--	--

EXAM DELIVERY and INSTRUCTIONS:	<p>The all-electronic, and open book final exam of the Winter 2020 edition of ECSE 223 takes place from April 21 07:00 (EDT) to April 23 07:00 (EDT). The final exam will consist of four parts:</p> <ol style="list-style-type: none">1. Concepts and Semantics (25 marks): This part takes the form of a MyCourses Quiz, and it will be available for completion between April 21 07:00 and 14:00 (EDT). However, you will have 90 minutes to complete this part from the time you first opened the quiz,2. Domain modeling (30 marks): This part will be a MyCourses assignment where you need to submit a single Umpel file representing your domain model (with appropriate comments to highlight your design decisions). The submission deadline is April 23 07:00 (EDT).3. Statechart modeling (30 marks): This part will be a MyCourses assignment where you need to submit a single Umpel (.ump) or Yakindu (.sct) file representing your statechart model. The submission deadline is April 23 07:00 (EDT).4. Practice (15 marks): The specification will be made available to you as a MyCourses assignment, but you need to complete this task dominantly on a private GitHub repository where an initial project content will be provided to you. You are required to make a Github commit for your work in your private repository. Finally, you will need to submit the URL link of your final commit (aka commit link) on MyCourses. The submission deadline of the final commit link is April 23 07:00 (EDT). Note that for this part, submitting your source code directly to MyCourses does not count as a submission and it will not be evaluated. <p>You are required to follow the specific instructions to each question!</p>
PREAMBLE:	<p><i>By submitting this exam, I certify that it represents solely my own efforts. I confirm that I understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures, and am aware of my responsibilities under the Student Assessment Policy.</i></p>

Question 2 (30 marks): Domain Modeling

Description: A smart home automation system (SHAS) offers various users to automatically manage smart home automation tasks. A smart home (located at a physical address) consists of several rooms, each of which may contain sensor devices and actuator (controller) devices of different types (e.g. temperature sensor, movement sensor, light controller, lock controller). Each sensor and actuator have a unique device identifier. Once a new sensor or actuator is activated or deactivated, SHAS will recognize the change and update its infrastructure map.

When SHAS is operational, a sensor device periodically provides sensor readings (recording the measured value and the timestamp). Similarly, a predefined set of control commands (e.g. *lockDoor*, *turnOnHeating*) can be sent to the actuator devices with the timestamp and the status of the command (e.g. requested, completed, failed, etc.). All sensor readings and control commands for a smart home are recorded by SHAS in an activity log.

Relevant alerts in a smart home can be set up and managed by its owner by setting up automation rules. An automation rule has a *precondition* and an *action*. The precondition is a Boolean expression constructed from *relational terms* connected by basic *Boolean operators* (AND, OR, NOT). Atomic relational terms may refer to rooms, sensors, actuators, sensor readings and control commands. The action is a sequence of control commands. For example, a sample rule could specify:

when actualTemperature by Device #1244 in Living Room < 18 **and** window is closed
then turnOnHeating in Living Room

Automation rules can be created, edited, activated and deactivated by owners. Only deactivated rules can be edited. Rules can also depend on or conflict with other rules, thus a complex rule hierarchy can be designed. SHAS records whenever an active rule was triggered using a timestamp.

Specific instructions: Specify the *domain model* of the SHIS as UML Class diagrams in Umple and store it in a file named as **YourMcGillID_02.ump**. You must indicate all relevant classes, enumerations, attributes and associations covered in the domain description. For associations, you need to include role names, directionality, multiplicity, and compositionality. The types of attributes also need to be shown. You do not need to show any (a) operations or (b) constraints.