Correctness by Construction

Term Project: ER Classification Room

The deadline to turn in the Event B model and related documents is:

Tuesday, May 19th 2026, 23:59

The project presentations will take place on:

Wednesday May 20th 2026, 15:00-18:00

and

Wednesday May 27th 2026, 15:00-18:00

Every presentation should be max. 20 minutes (questions included).

Please send me the data for each team by **Wednesday, May 6th**

Manuel Carro
manuel.carro@upm.es

April 20th, 2026

1 Goal

The objective of this project is to develop an Event B model for a simplified admittance control for the ER classification section of a hospital. Patients wait for a *classification room* to be available. They know whether a room is available or not because an external signal (for example, a red/green light) denotes so.

When a room is available, a patient can walk into it, activating an entrance sensor. Inside, a doctor performs a preliminary exam to determine how urgent is the case and to assign it to another doctor who will later perform a deeper examination and determine the treatment, if any. Then, the patient leaves the classification room through an exit door, is registered by an exit sensor, and enters a general waiting area, to wait to be called by a second doctor. At this point the patient leaves the second waiting area. Figure 1 shows a sketch of the areas previously described.

A patient is allowed to enter a classification room if and only if:

- There is no other patient inside the room.

- There is free space in the second waiting area for another patient to wait.

There is an unbound number of patients waiting to be classified. There is a constant, finite number of classification rooms. There is a constant, finite number of spots in the post-classification waiting area.

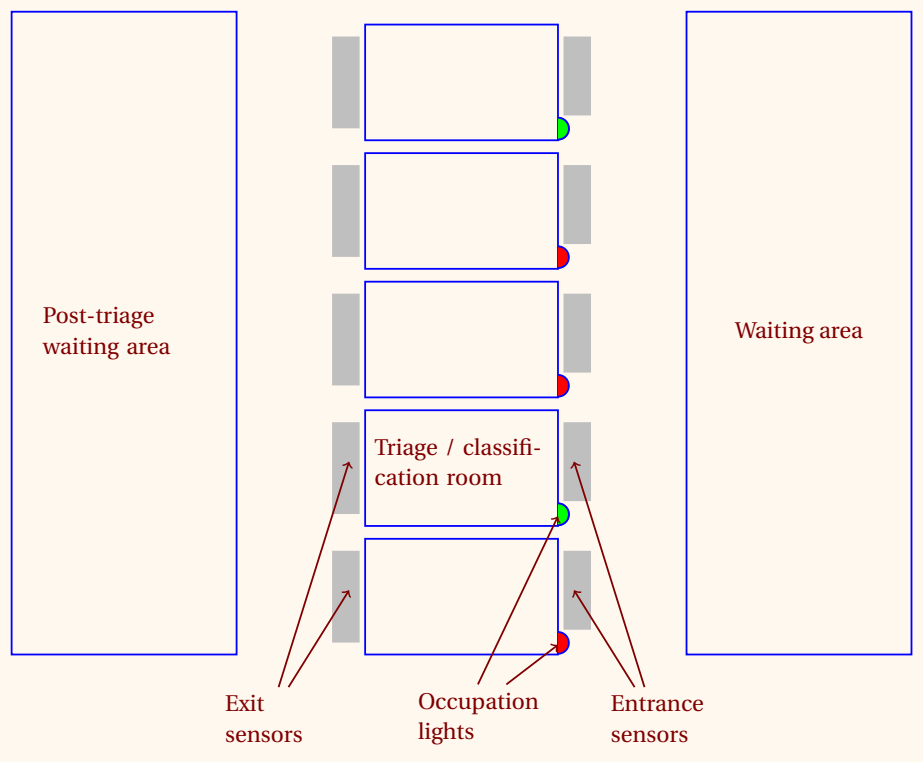

Figure 1: The hospital classification area. *Triage* is another term used to denote classification, especially in cases of emergency or great need.

2 Requirements

EQP 1	A hospital has *classification rooms*. There is a *busy / available* signal at the entrance of every classification room, as well as sensors at the entrance and exit of every classification room.

FUN 2	At most one patient can be in a classification room at any moment.

ENV 3	There is an unlimited number of patients waiting to enter the classification rooms.

FUN 4	One patient enters the classification room only when the signal marks it is *available*. The patient stays inside the classification room for an undetermined amount of time before exiting.

EQP 5	There is a sensor at the entrance and at the exit of every classification room.

FUN 6	A sensor is activated when it detects the presence of a person and deactivated when it does not detect the presence of a person.

EQP 7	The hardware and software are fast enough to register the entrance / exit of a patient without missing patients.

EQP 8	There is a post-classification waiting room with a maximum capacity.

EQP 9	Patients can be called to be seen by a doctor an undetermined amount of time after they enter the post-classification waiting room. Then, they leave the waiting room and they do not return directly to it.

FUN 10	A classification room is available only when there is no patient in that room and there is space for an additional patient in the post-classification waiting area.

FUN 11	The model must not deadlock.

FUN 12	The model must not have events that can be continuously enabled, thus running the risk of *starvation* or *livelock*.

FUN 13	The capacity of the post-classification waiting room must not be exceeded.

3 Tasks

Your task is to develop an Event B model to control the classification room lights, respecting the requirements presented in Section 2, according to the signals received from the sensors. Use invariants to capture these requirements when possible. You can decide whether to perform model refinement or not. All the proof obligations that Rodin generates should be proven or reviewed. Pay special attention to the separation between events that represent the actions of the environment and the events that represent code / actions that belong to a controller, and likewise for the variables. Variables that belong to the controller are not accessible from the outside and the other way around: the communication between both worlds must take place through sensors (environment ⇒ controller) and actuators (controller ⇒ environment). The only possible exception is implementing some specific property, such as "things are fast enough", that are intrinsic to the world and that are assumptions we must capture in order for some properties of the model to be provable.

4 Teams, Submission, and Presentation

The project is to be done (and turned in) by **teams of three students**. The work developed has to be presented in one of the presentation sessions. In order to schedule the presentations, please send me (manuel.carro@upm.es) an email with the names of the team members and a name for the team itself as soon as possible and, in any case, by **Wednesday, May 6, 23:59** so that we have time to schedule the presentations. Also, please get in touch with me ASAP if a team of three cannot be assembled.

The material to be submitted is:

1. A Rodin project with the model for the problem. The proofs necessary for Section 3 must be discharged. If some proof is not discharged, it must be reviewed (Ⓡ). Name it as Triage_TP_*Initials*, where *Initials* are the initials of your team's name.
2. The slides used for the presentation in PDF format.

Every team should make a presentation of at most 20 minutes, including questions (I suggest 15 min. presentation plus 5 min. questions) explaining the strategy to solve the problem and anything else that you think is worth mentioning (for example, difficulties found, etc.) Every team member should present part of the work, ideally dividing the time equally among team members. Your classmates will have the chance to make questions related to the presentation and its contents.

5 Additional Information

Using Additional Theorem Provers You will most likely need the *Atelier B* provers installed (which you should have, anyway). It should be possible to design a model for the problem for which Rodin can discharge all the proof obligations (almost) automatically — maybe clicking some buttons in the *Proving View*.

If, despite interacting with the theorem provers, you cannot discharge a PO that you are convinced is correct, you can try with the SMT solvers. Go to Help → Install new software, select Rodin plugins → Prover Extensions → SMT Solvers, and install them. In the Prover View a new button will appear with which you can select additional provers which use the hypotheses that appear in the Selected Hypothesis panel. The SMT solver can in many cases prove sequents that PP or ml cannot prove. Likewise, the Atelier B provers can sometimes discharge proofs that the SMT provers cannot.

If you cannot discharge some PO that you are convinced is correct, please mark it as *reviewed* (with the Ⓡ button) and follow the instructions in point 2 of Section 4.

Remember not to use the NewPP theorem prover. It is unsound: it is known to have bugs enough so that it is not reliable for normal use.

Sensors and Actuators Since we have already seen models to convert a physical sensor into a logical signal, you can assume a simple model for the sensors and actuators:

- We will assume that a sensor's state can be read by consulting the value of a variable associated to it. When the variable is on, the sensor has detected presence. It has to be

explicitly set to off again to register new patient's entering (resp., exiting) the classification room. The events that are associated to the environment and to the controller can read and change the values associated to the sensors.

- Likewise, setting the color of an occupancy light boils down to setting the value of a variable. Events representing what happens in the environment can "see" the light colors by reading their values.

Seeing the Whole Model A large model can sometimes be difficult to work with — the display may be cluttered with large formulas. Rodin can export models to LaTeX which can then be processed to generate a PDF that is displayed / printed. That is performed by a Rodin plugin — see http://wiki.event-b.org/index.php/B2Latex for an explanation of how to install the plugin and use the generated LaTeX file.

Theorems and Order of Formulas

- Although the different parts of the guards are in logical conjunction, sometimes changing the order of the guards helps the theorem provers to do their job (this is actually necessary in some cases — see the next bullet).

- The order of formulas is relevant to write "lemmas". Flagging a formula as a "theorem" in a context / invariant section / guard section makes Rodin to try to prove it from the *previous* formulas within its scope. If it is proven, it becomes available to further proofs. It would therefore work as a lemma that helps prove additional properties.