

Mini-Project (SCV)

Description

We propose to design, test and check formally an emergency care service. The injured people will be taken to emergency rooms for immediate treatment. Resources to be modeled include physicians, nurses, and examining rooms, as well as the resource consumers, the patients. When a patient first arrives at emergency room, he/she proceeds to check in. Then, the receptionist checks the resources and number of current patients to determine a waiting time. If the waiting time is bigger than a certain value, then the newly arrived patient would not be admitted into the emergency room. The responsible of the service can decide to ask for more resources from an external resource provider. Otherwise, the patient would check into the emergency room and be given paperwork to fill out. After the patient completes the paperwork, he/she would wait to be treated and an available nurse will start processing the paperwork. When the nurse completes the process and there is an examining room available for the patient, then the patient would enter the room. When a physician becomes available, he/she would start examining the patient in the examining room. After the completion of the treatment, the patient proceeds to check out and leave the emergency room. When there is no client waiting, the responsible of the service can decide to contact the resources provider, to offer a resource (a room or a doctor) in case other services need it. The resource provider, shares with the any service of the hospital the rooms and physicians resources, and communicates with it asynchronously via buffers (namely b_0 and b_1). A message in b_0 is an authorization from the emergency service to the provider to use its resources and to give them to an other service which needs it.

Within this project, you are asked to use the modeling, test, and verification concepts seen during the course in order to implement the above system. You will work in pairs (at most) and you have upload your work on <https://campus.groupe-efrei.fr> as a compressed folder containing the following elements :

- A structured pdf document containing the description of you method to design (UML models), to model (screen shot of the Petri net models using TINA toolbox), to test (screen shot of the Junit tests) and to verify (LTL formulae + screen shot of the TINA model checker) the system.
- the java source code
- the Petri net models(.ndr)

Formal modeling and verification

- Model the emergency care service with a Petri net
- Model the resource provider with a Petri net
- Combine the two previous models in order to model the interaction between the emergency care service and a resource provider through shared resources
- By varying the number of available resources initially, and using the TINA toolbox model checker, express and check the following properties :

- Deadlock freeness
- When a patient is admitted in the service, he/she will eventually be examined by a doctor
- Each time an examining room is reserved, it will eventually be released

UML-based modeling and test

Inspired by the formal model established in the first part,

- Identify the main use cases and draw the underlying use-case Diagram.
- Draw the sequence/communication diagram representing the admission of an injured person in the emergency care service.
- Draw the state-class diagram to model the emergency care service.
- Add at least 3 OCL invariants on class attributes, class methods, class association
- Design the hospital roster and write the body of a Java method prototype that draws it
- Write a class containing two unit tests written using JUnit to test compliance of the described the state machine of some classes (for instance, the emergency care service). Use the setUp method with the @Before annotation.
- Write a validation test inspired from the hospital roster. You can use the assertEquals class method of the Assert class with two arguments of type Object. 4. (affect a doctor to a patient.) requires interactions with other classes. We want to create at least 2 integration tests to verify these interactions.