FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

MESTRADO EM ENGENHARIA INFORMÁTICA E COMPUTAÇÃO

SOFTWARE SYSTEMS ARCHITECTURE

29 April 2024 • Estimated lenght: 3-4 pages • Duration: 75min + 15min tolerance

Before starting your individual test, please carefully read the description of the software system and answer the questions **always justifying them succinctly and clearly**. When useful, you may explicit all the assumptions you made to answer the questions.

Design a system to control a **nuclear power plant**. Here are some details.
(Caveat: we don't REALLY know how a nuclear power plant runs, so these are wild guesses.)

i. The basic idea is that a reactor generates heat, turning water to steam. Then you use the steam to turn turbines that produce electricity. You cool the steam to water and send it back into the reactor core to turn to steam again.
ii. You control the intensity of the nuclear reaction with uranium fuel rods and graphite control rods. Inserting the fuel rods more increases the reaction. Inserting control rods slows the reaction because the graphite absorbs the gamma rays. So you push them in and pull them out. (I believe that the pushing and pulling is controlled by the computer.)
iii. If the reaction is too intense, the core gets too hot, melts, and all sorts of bad things happen. But if the reaction is too slow, you don't produce any power, which is also bad.
iv. Security: to make the system as secure as possible, it is not connected to the internet. It communicates with various terminals, but all within the secure facility.
v. There are different terminals, and they display different data in different formats, depending on the needs. Terminals include:
    a. Safety: shows reactor temperature, control rod positions, radiation levels, etc.
    b. Control: shows some of the safety things but tracks and allows limited control of the reactor. Remember, we want to minimize human error.
    c. Power production.
vi. Of course, safety is everything. In the worst case, the system shuts the reactor down. Of course, you want to avoid it if at all possible, because it is VERY expensive to restart the reactor.

The architecture itself is simple, but this system is all about quality attributes. Do the following:

Q1. Design and document the architecture above, which must include the following:
- Diagrams: combine the logical and physical view but be clear what is hardware and what is software.
- Name and descriptions of at least two architecture patterns you use; also make them clear in the diagram.
- A textual description of the architecture, including descriptions of the components, how things work, etc.

Q2. Design and document two quality attributes, as suggested below: (1) reliability AND (2) one other quality attribute of your choice from availability, performance, usability, or security:
- Describe what the QA means in this context, including several aspects of it.
- Describe consequences of failure. Be specific ("the reactor blows up" is not sufficient)
- Describe at least two tactics you will use.

NOTE: No sequence diagrams, use cases or use case maps are required.

The End.

Ademar Aguiar, Neil Harrison