17-423/723 Homework 2: Design for Reuse

Spring 2024

Released: Wed, March 13, 2024
Due: Wed, March 20, 2024 11:59 pm (on Gradescope)
Estimated Time Effort: 2-3 hours

Learning Objectives:

- Recognise and apply principles for design for reuse
- Evaluate design alternatives and their trade-offs between design for reuse and other design objectives
- Generate viable design solutions that improve the reusability of given design solution
- Apply appropriate abstractions & modeling techniques to communicate the interface of a module

Pre-required Knowledge:

- Basic understanding of publish-subscribe (See lecture on Quality Attributes & Trade-offs & the required reading for this lecture)
- Knowledge of design principles for design for reuse & design with reuse (See lectures on Design for Reuse & Design with Reuse)
- Basic understanding of other quality attributes. (See lectures on Quality Attributes & Trade-offs, Design for Change, Design for Interoperability, and Design for Testability)

Case Study Package

The case study package for this assignments is the PyPubSub package implementing a publish-subscribe connector for Python: https://pypubsub.readthedocs.io/en/v4.0.3/

The basic usage of the package is described here:
https://pypubsub.readthedocs.io/en/v4.0.3/usage/usage_basic.html

The interfaces of the core classes are documented here:
https://pypubsub.readthedocs.io/en/v4.0.3/usage/core_classes.html

Please don't try to understand all classes and methods. Focus on:

- **`Publisher.sendMessage`** (`pub.sendMessage` forwards to the `Publisher` class), which publishes a message to a topic (similar to a combination of the `advertise` call directly followed by the `publish` call in ROS from the reading)
- **`Publisher.subscribe`** (`pub.subscribe` forwards to the `Publisher` class), which lets a callback function subscribe to a topic (equivalent to the `subscribe` call in ROS from the reading)

If it takes you longer than ~30min to understand what the package does and how it does it, please message us and we'll help you.

Tasks

Q1. EVALUATE EXISTING DESIGN (40 points)

i. Describe at least two **assumptions** that this package makes about its reuse context.
ii. Describe two reuse scenarios in which reusing the package is **appropriate** and why reusing the package in these scenarios is appropriate. Remember that reuse scenarios include a reuse context (i.e., a concrete software system that is reusing the package or a specific application domain) and measurement.
iii. Describe two scenarios in which reusing the package is **not appropriate** and why reusing the package in these scenarios is not appropriate. **Hint:** You will need to design an improved version for one of the scenarios, so please pick at least one for which you can imagine how to satisfy this reuse scenario.
iv. Describe which **design principles** (at least two) most significantly contribute to making this package as reusable as it is, how this package applied the design principles, and how this is contributing to its reusability.

Q2. GENERATE & COMMUNICATE IMPROVED DESIGN (60 points)

For one of the scenarios you described in Q1.iii, generate an improved design of the package that supports this reuse scenario (while possibly scarifying other quality attributes)

i. Specify the part of the **interface** (syntax and semantics) that is relevant to the reuse scenario to communicate the design of your improved package.
ii. Describe what **changes** to the implementation of the PyPubSub package are needed to support the changed interface (max. 1 paragraph).
iii. Describe how your redesign **improves the reusability of the package**. You might refer to reusability design principle(s) that you have applied or base your argument on the evaluation of the reuse scenario.

Tips

This homework is intentionally designed open-endedly to allow you to pick reuse scenarios from a domain that you are familiar with or a domain that you would like to practice in this homework. If the open-ended nature of this homework makes it harder for you to think of many different potential reuse contexts, consider, for example:

- Internet-of-Things systems for which senders and receivers are deployed on different devices and communicate with each other via a network.
- Robotics systems that have strict real-time requirements for high-priority tasks and do not have real-time requirements for low-priority tasks. **Hint**: Think of priority queues!

If you pick an example different from the two hinted above, you will receive **5 bonus points** for accepting the learning opportunity to think more creatively as a reward for a solution that is above the expectations! :)

Submission & Grading

Compile your answers into a single PDF file and upload it through Gradescope by the indicated deadline. **Please correctly map the pages in the PDF to the corresponding questions on Gradescope.**

This assignment is out of **100** points. For full points, we expect:

- **Q1**: (i) Correctly identified assumptions made by the case study package about its reuse context (**6 pts**), (ii) measurable and concrete reuse scenarios and a clearly argued reason why the case study package supports it (**10 pts**), (iii) measurable and concrete reuse scenarios and a clearly argued reason why the case study package does not support it (**10 pts**), (iv) clearly argued connection between design principles mentioned in the lecture and the case study module (**14 pts**)
- **Q2**: (i) Clearly documented syntax and semantics of the improved interface (**30 pts**), clearly described changes of the underlying implementations (ii) (**10 pts**), (iii) a clear chain of arguments connecting the improved reusability; arguments should be clearly based on design elements described in Q2 (**20 pts**)
  5 Bonus points for solutions different from the hints.