Architectural Thinking for Intelligent Systems

Assignment for Lecture A9

Team Members:
1. Ankit Agrawal, 2581532
2. Anika Fuchs, 2580781

1) Architectural principles:

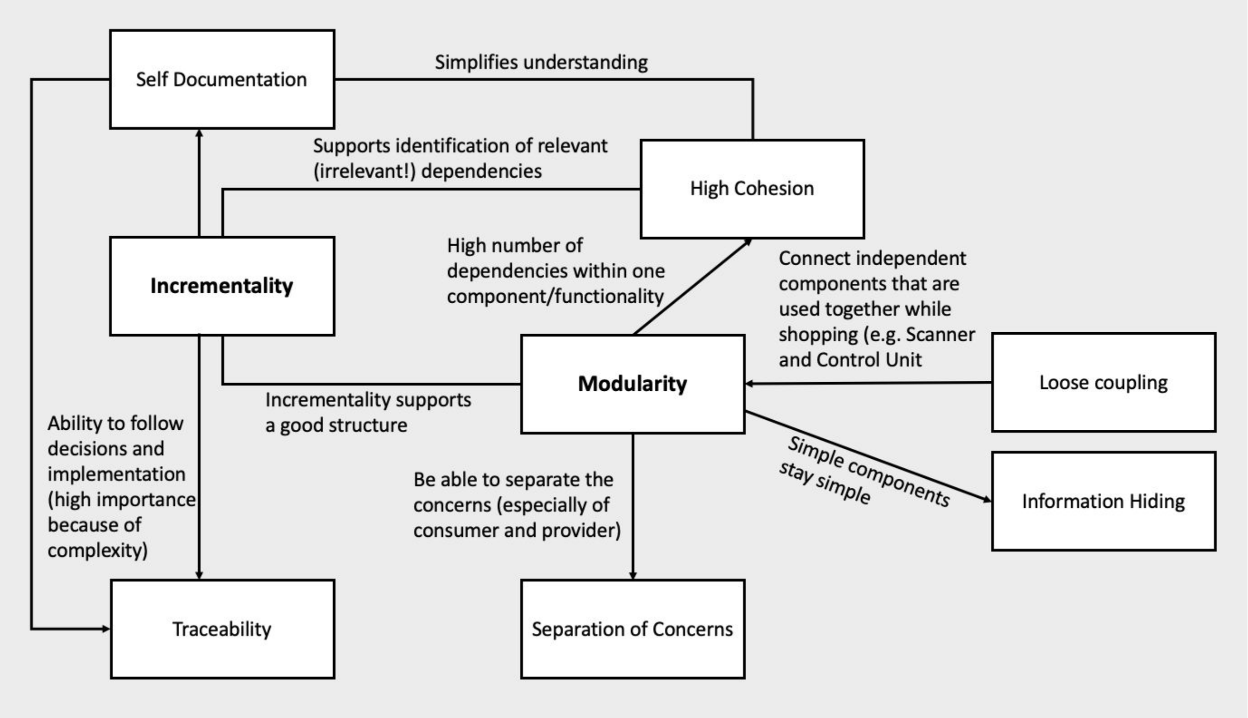
The two architectural principles that are most important for our system are:
Modularity
Incrementality

System qualities that we selected for Assignment 6 are:
Performance
Reliability

Modularity: As our system is multi-faceted and utilizes techniques from various domains such as Control unit, Scanner, Locomotion, Data Engineering, and Software Engineering. We could modularize each functionality based on the technology used and then hire different specialists from the above-mentioned fields and employ them to design & develop their respective components in parallel. Further, this modular approach would fasten up the development process, reduce complexity, and increase the understandability of the system and also paves a way for evolutionary improvement of the modules.
With the help of inducing modularity, we build the individual components separately and make them perfect for their main usage, for example, the locomotion(wheels and legs) are entirely separate, self-contained, and independent modules, so we need to separate them, built them parallelly and by individual domain experts. With this method, the individual components are better managed, trained, and personalized in special environments and constraints and hence makes the system more reliable and increases the overall performance of the system.

Incrementality: Development of a system in one iteration or using something like the waterfall model would be very challenging and won't be much reliable. As the system is very complex and there are many different components included, it is always a better idea to follow agile methodologies. The development should always be done in phases/sprints, with small milestones and most importantly, there should be continuous feedback from the stakeholders. Following this methodology, helps us to do early prototyping and as our system is focused more on user's statistics and usage, it is very important to get feedback early and regularly. Hence, this architecture principle helps to achieve better performance and makes the system more reliable.

2) Diagram variant of slide 20:



3) Tactics:

Quality 1: **Performance**:
Tactic: **Performance tactic**:

The goal of performance tactics is to generate a response to an event arriving at the system within some time-based constraint. Performance tactics control the time within which a response is generated. This may include caching or making multiple copies of the product and user data. The frequently used product data can be cached for faster retrieval and a shorter wait time. Further, the accuracy of the system (major quality attribute) can be increased using various techniques such as Cross-folding, Normalization of data, Increased Compute Memory resources, etc. Improving the algorithms used in critical areas such as payment, billing, etc to decrease latency.

Quality 2: **Reliability**:
Tactic: **Availability tactic**:

The availability tactic builds on top of reliability by adding the notion of recovery and repair. To make a system reliable, it should be able to detect and recover from any failure, there should be less downtime, the system should be able to adapt in any environment. Similarly, all approaches to maintaining availability involve some type of redundancy, some type of health monitoring to detect a failure, and some type of recovery when a failure is detected making a system more reliable. In our tool, we can induce these features in critical areas like payment, database connection, user profiles, locomotion, scanner, etc.