Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Academic Year : 2024-25

Experiment No. 9
Develop architectural model for the selected case study
Name: Krisha Chikka
Std/Div: TE/1 Roll no. : 30
Date of Performance: 30/09/2024
Date of Submission: 07/10/2024

Aim: To develop architectural model for the selected case study.

Objective: To analyse the case study for Currency Detector for the Visually Impaired and design architectural model for the same.

Theory:

An architecture model is a partial abstraction of a system. It is an approximation, and it captures the different properties of the system. It is a scaled-down version and is built with all the essential details of the system. Architecture modeling involves identifying the characteristics of the system and expressing it as models so that the system can be understood. Architecture models allow visualization of information about the system represented by the model. The modeling process can be bottom up/inside out, by which details of the system are built utilizing knowledge about components and interconnections and how they compose together to realize the characteristics of the system. Alternatively, it can be top-down/outside in, by which details of the components and interconnections are extracted from knowledge of the whole.

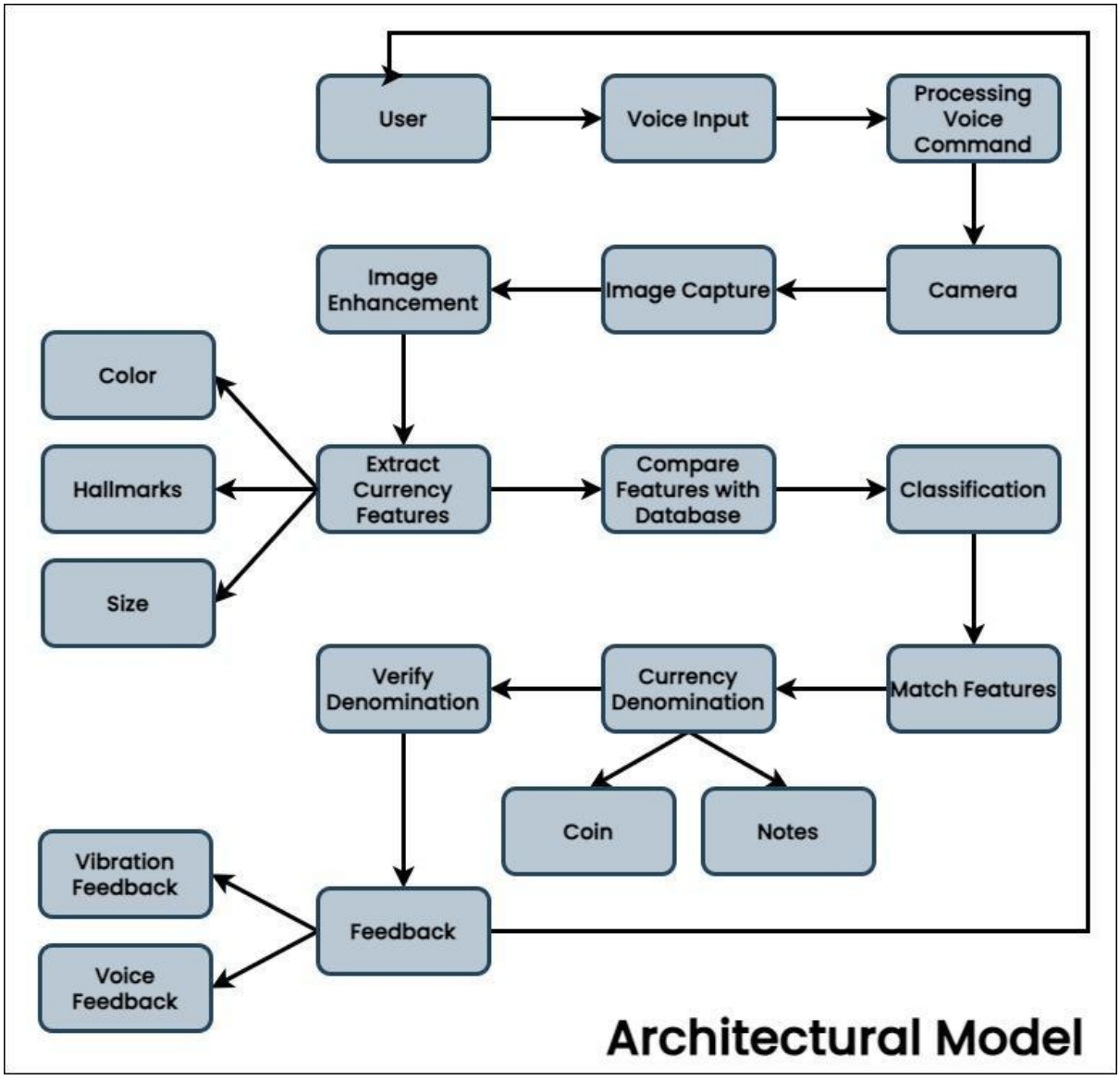
An **architectural model** (in software) is a rich and rigorous diagram, created using available standards, in which the primary concern is to illustrate a specific set of trade-offs inherent in the structure and design of a system or ecosystem. Software architects use architectural models to communicate with others and seek peer feedback. An architectural model is an expression of a viewpoint in software architecture.

Some key elements in software architectural model are:

- **rich**: for the viewpoint in question, there should be sufficient information to describe the area in detail. The information should not be lacking or vague. The goal is to minimize misunderstandings, not perpetuate them. See notes below on 'primary concern.'
- **rigorous**: the architect has applied a specific methodology to create this particular model, and the resulting model 'looks' a particular way. Here's the test of rigorousness: If two architects, in different cities, were describing the same thing, the resulting diagrams would be nearly identical (with the possible exception of visual layout, to a point).
- **diagram**: in general, a model may refer to *any* abstraction that simplifies something for the sake of addressing a particular viewpoint. This definition specifically subclasses 'architectural models' to the subset of model descriptions that are represented as diagrams.
- **standards**: standards work when everyone knows them and everyone uses them. This allows a level of communication that cannot be achieved when each diagram is substantially different from another. UML is the most often quoted standard.

- **primary concern**: it is easy to be too detailed by including many different needs in a single diagram. This should be avoided. It is better to draw multiple diagrams, one for each viewpoint, than to draw a 'mega diagram' that is so rich in content that it requires a two-year course of study to understand it. Remember this: when building houses, the architect delivers many different diagrams. Each is used differently. Frequently the final package of plans will include diagrams with the floor plan many times: framing plan, electrical plan, heating plan, plumbing, etc. They don't just say: it's a floor plan so 100% of the information that CAN go on a floor plan should be put there. The plumbing subcontractor doesn't need the details that the electrician cares about.
- **illustrate**: the idea behind creating a model is to communicate and seek valuable feedback. The goal of the diagram should be to answer a specific question and to share that answer with others to (a) see if they agree, and (b) guide their work. Rule of thumb: know what it is you want to say, and whose work you intend to influence with it.
- **specific set of trade-offs**: the architecture trade-offs analysis method (ATAM) methodology describes a process whereby software architecture can be peer-reviewed for appropriateness. ATAM does this by starting with a basic notion: there is no such thing as a 'one-size-fits-all' design. We can create a generic design, but then we need to alter it to specific situations based on the business requirements. In effect, we make trade-offs. The diagram should make those specific trade-offs visible. Therefore, before an architect creates a diagram, he or she should be prepared to describe, in words, which trade-offs they are attempting to illustrate in this model.
- **trade-offs inherent in the structure and design**: a component is not a trade-off. Trade-offs rarely translate into an image on the diagram. trade-offs are the first principles that produced the design models. When an architect wishes to describe or defend a particular trade-off, the diagram can be used to defend the position.
- **system or ecosystem**: modelling in general can be done at different levels of abstraction. It is useful to model the architecture of a specific application, complete with components and interactions. It is also reasonable to model the systems of applications needed to deliver a complete business process (like order-to-cash). It is not commonly useful, however, to view the model of a single component and its classes as software architecture. At that level, the model, while valuable in its own right, illustrates design much more so than architecture.

Solution:



Conclusion: In this experiment, we successfully developed an architectural model for the Currency Detector for the Visually Impaired. The model provided a visual representation of the system, highlighting key components and their interactions. By following a structured approach, we captured the essential characteristics of the system and the trade-offs involved in its design. This architectural model helps in understanding the overall structure, making it easier to communicate, evaluate, and refine the system based on the specific requirements of the case study.