Lecture 2

Wednesday, 6 September 2023 15:00

Recognize viewpoints, their addressed concerns, and typically used models

System architecture and software architecture

quality of an architecture

architectures are useful to

- understand a complicated system
- analyze design alternatives
- system properties
- communicate between the authors/customers
- document for later reference
- decompose & synthesize parts

purpose of the architecture

1. communication / understanding
2. analysis
3. synthesis (construction)

relocation / migration → different books use different definitions, or do not differentiate at all

non is valid

ordinarily is the process of conceiving, defining, expressing, documenting, communicating, certifying proper implementation of, maintaining and improving an architecture throughout a system's life cycle

architecture consist of the fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution

architecture description are work products used to express an architecture

architecture framework

environment of a system: context determining the setting and circumstances of all influences upon a system

stakeholders of a system: individual, team, organization (or classes thereof) holding concerns with respect to a system

system concern: interest in a system relevant to one or more of its stakeholders

architecture view: work product expressing the architecture of a system from the perspective of specific system concerns

architecture viewpoint: work product establishing the conventions for the construction, interpretation and use of architecture views to frame specific system concerns

model kind: convention for a type of modeling

stakeholders + concerns = environment

↓

responsibilities

1. communicate concerns to the architect
2. review the architectural description
3. decide on the acceptance of the architecture and claims about the system

systems do not need to have an architecture

↓

small ones in particular

model kinds are used to describe architecture viewpoints

a view is the description of the whole system architecture from the perspective of a set of concerns

a viewpoint defines the rule for constructing, interpreting and using a view

Google File System (GFS): chunk size = 64 MB

example used below

for following roles:

- GFS client
- GFS master → single point of failure; use shadow master as backup
- GFS chunkserver

replicates to 3 different locations (non chunk)

client —file name, chunk index→ master

client handle, chunk location

Kruchten viewpoints:

	stakeholders
1. logical view	user
2. development view	programmer
3. process view	system integrator
4. deployment / physical view	system engineer
+ 1: scenarios (use case diagram)	

RM viewpoint catalog: (pick your favorite viewpoints for specific system, it's common to mix them)

- context viewpoint: describes relationships, dependencies and interactions between system and its environment (people, systems, external entities)
- functional viewpoint: describes the system's runtime functional elements and their responsibilities, interfaces and primary interactions
  - e.g. client requests chunk handle location from master; client holds responsibility to store chunk from chunk servers, the files are stored in different location, which can be accessible by multiple client (multiple users)
- information viewpoint: describes the way the architecture stores, manipulates and distributes information
  - e.g. database; client requests chunk handle from master, the master checksum
- concurrency viewpoint: describes the concurrency units of the system, their functionality, and the required coordination
  - e.g. schedule access / checksum; the master waits until the file data of the master
- development viewpoint: describes the development of the system
  - e.g. how code is divided into modules
- deployment viewpoint: describes the deployment environment of the system
  - e.g. hardware, cloud/local/datacenter, ...
- operational viewpoint: describes operational aspects of the system in its production environment
  - e.g. install, upgrade, configure

problem addressed by (google) data center: organize servers such that physical hardware can easily replaced safely and securely for maintenance purposes, even though racks and power are handled efficiently

servers, hard drives are replaced (e.g. because they fail)

↳ with hot drive replaced, and data is retrieved from a different location for the process

Extra-functional requirements: mentioned through the modules

non-functional requirements

architecture	design
active entity (component/module)	components & subsystems of entity
different and ≥ different views	different design may alternative version
class of acceptable solution	single acceptable solution
broader concerns of stakeholders	conform to an architecture
modeled under constraints	make correct decision
technology-independent	

design process

1. (formal) analysis
2. apply strategies
3. synthesize
4. refinement