

1

Definition of SW Arch

the set of structures needed to reason about the system, which comprises of both software elements, relations among them, and properties of both

AKA how sets of structures relate

2

3 categories of arch structures

modules

component and connector

allocation

3

Basic definition of module structures

structures that allow for the static structuring of code

AKA containers for written source code

4

2 characteristics of modules

each module has a computational responsibility

each module is the basis of work assignments for programming teams

5

4 important questions relating to modules

what is the primary functional responsibility assigned to each module?

what other SW elements is a module allowed to use?

what other SW does it actually use and depend on?

what modules are related to other modules by generalization or specialization (inheritance) relationships?

6

The basic definition of component-and-connector structures?

component - a structure that represents a runtime executable

connector - represents the SW and HW at runtime that dictates how components communicate with each other

7

1 important characteristic of component structures

they do not have a 1-to-1 correspondence with module structures. Typically a many-to-many correspondence

8

6 important questions relating to component-and-connector structures?

what are the major executing components and how do they interact at runtime?

what are the major shared data stores?

what parts of the system are replicated?

how does the data progress through the system?

what parts of the system can run in parallel?

can the system's structure change as it executes and how?

9

The basic definition of allocation structures

they describe the mapping from software structures to the system's environment

10

4 types of environments for a software system

organizational (company structure)

developmental (how dev teams are organized)

installation (what need to be installed on a machine that the system runs on)

execution (how are components executed on certain CPUs)

11

3 characteristics of an arch structure abstraction

-specifically omits info about elements that is not useful for reasoning (thinking about) the system as a whole

-omits info that has no ramifications outside of a single element

-suppresses certain info like private details of elements that solely pertain to internal implementation

12

Why do we have abstractions for architecture?

to tame the complexity of the system architecture

13

Definition of a view

a set of architectural structures and their relationships that a pertinent to a certain category of stakeholders

14

3 important questions in regards to allocation structures

-what processor does each software element execute on

-in what directories or files is each element stored during development, testing and system building

-what is the assignment of each software element to dev teams

15

5 types of module structures

decomposition structure

uses structure

layer structure

class (or generalization) structure

data model

16

5 characteristics of decomposition structures

-the units are modules that are related by the is-a-submodule of relation

-it shows how modules are decomposed into smaller modules recursively until the modules are small enough to be easily understood

-it determines the system's modifiability, by assuring that likely changes are localized

-used as the basis for the project's organization such as the documentation, and the integration and test plans

-units in this structure have names that are organization specific like "segment" or "subsystem"

17

3 characteristics of uses structures

- the units are modules (also possibly classes)

- the units are related by the uses relation

- is used to engineer systems that can be extended to add functionality, or from which useful functional subsets can be extracted

18

What does it mean for a unit of software to "use" another (e.g the two nits have a uses relation)

if the correctness of the first unit of SW requires the presence of a correctly functioning version of the second

19

3 characteristics of layer structures

- the module units in this structure are called layers

- a layer is an abstract "virtual machine" that provides a cohesive set of services through a managed interface

- layers are allowed to use other layers in a strictly managed fashion

20

1 important benefit of using layered structures

layer structures give the system portability (the ability to change the underlying computing platform)

21

4 characteristics of class (generalization) structures

- module units in this structure are called classes

- the relation is inherits-from or is-an-instance-of

- this view supports reasoning about collections of similar behavior or capability

-allows one to reason about reuse and the incremental addition of functionality

22

1 characteristic of the data model

describes the static information structure in terms of data entities and their relationships

23

2 types of component-and-connector structures

service structure

concurrency structure

24

2 characteristics of service structures

-the units are services that interoperate with each other by service coordination mechanisms such as SOAP (Simple Object Access Protocol)

-it helps to engineer a system composed of components that may have been developed anonymously and independently of each other

25

4 characteristics of concurrency structures

-helps to determine opportunities for parallelism and the locations where resource contention may occur

-the units are components

-the connectors are their communication mechanisms

-the components are arranged into logical threads

26

3 types of allocation structures

deployment structures

implementation structures

work assignment structures

27

5 characteristics of deployment structures

- shows how the software is assigned to hardware processing and communication elements

- the elements are software elements, hardware entities, and communication pathways

- employs allocated-to and migrates-to relations

- used to reason about performance, data integrity, security and availability

- of interest in distributed and parallel systems

28

What are allocated-to and migrates-to relations

allocates-to relation shows which physical units software elements reside

migrates-to is the same thing but when the allocation is dynamic

29

2 characteristics of implementation structures

- shows hows software elements (usually modules) are mapped to file structures in a system's development, integration, or configuration control environments

- the mapping is critical for the management of deployment activities and build processes

30

2 characteristics of a work assignment structure

- assigns responsibility for implementing and integrating the modules to the teams who will carry it out

- determines the major communication pathways among teams

31

2 benefits of including work assignment structures in the architecture

Study These Flashcards

- it demonstrates that the decision about who does what work has architectural and management implications

- allows the architect to know the expertise required for each team

32

How do structures map to each other (in terms of relationships between each other)

Study These Flashcards

a given structure can be a single element within another structure, it can be part of several elements within a structure or even be a part of an element within a structure

AKA mapping between structures is generally many-to-many

33

what is an architectural pattern?

Study These Flashcards

elements composed in ways that solve particular known problems

34

3 characteristics of an architectural pattern

Study These Flashcards

the particular compositions of elements

- have been found useful over time, and over many different domains

- they have been documented and disseminated

-provide packaged strategies for solving some of the problems facing a system

35

1 common module type pattern

Study These Flashcards

layered pattern

36

what is the layer pattern?

Study These Flashcards

when the uses relation among software elements is strictly unidirectional, a system of layers emerges

(a layer being a set of related functionality)

37

2 common component and connector type patterns

Study These Flashcards

shared data (or repository) pattern

client-server pattern

38

3 characteristics of the shared-data pattern

Study These Flashcards

-comprised of components and connectors that create, store, and access persistent data

-repository usually is commercial DB

-connectors are the protocols for managing the data, such as SQL

39

2 characteristics of the client-server pattern

Study These Flashcards

-the components are the client and the servers

-the connectors are the protocols and the messages they share with each other to carry out the system's work

40

2 common allocation type patterns

Study These Flashcards

multi-tier pattern

competence center pattern and platform pattern

41

2 characteristics of the multi-tier pattern

Study These Flashcards

-describes how to distribute and allocate the component of a system in distinct subsets of hardware and software connected by some communication medium

-this pattern specializes the generic deployment structure

42

1 key characteristic for judging an architecture

Study These Flashcards

how easy is it to add a new feature after the system has been initially deployed

43

5 characteristics of processes which typically generate good architectures

Study These Flashcards

-the architecture should be the product of a single architect or a small group of architects with a clear technical leader

-the architecture should be based on a prioritized list of well-specified quality attribute requirements

- the architecture should be specified using views that address the most important stakeholders

- the architecture should be evaluated for its ability to deliver the system's important quality attributes

- the architecture should lend itself to incremental implementation

44

4 structural characteristics that typically lead to good architectures

Study These Flashcards

- the arch should feature well-defined modules whose functional responsibilities are assigned based on the principles of information hiding and separation of concerns

- unless the requirements are totally unique, the quality attributes should be achieved using well known architectural patterns and tactics specific to each attribute

- the architecture should never depend on a particular version of a commercial product or tool

- modules that produce data should be separate from modules that consume data

45

4 more structural characteristics that typically lead to good architectures

Study These Flashcards

- don't expect a one-to-one correspondence between modules and components

- every process should be written so that its assignment to a specific processor can be easily changed

- there should only be a small number of ways for components to interact

- should contain a specific and small set of resource contention ar