

Software Architecture

Background

- Any complex system is composed of sub-systems that interact
- While designing systems, an approach is to identify sub-systems and how they interact with each other
- Software Arch tries to do this for software

Architecture

- Arch is a design of the software that gives a very high level view of parts and they relate to form the whole
 - Partitions the sys in parts such that each part can be comprehended independently
 - And describes relationship between parts
- A complex system can be partitioned in many diff ways, each providing a useful view
 - Same holds true of software also
 - There is no unique structure; many possible

Key Uses of Arch Descriptions

- Understanding and communication
 - By showing a system at a high level and hiding complexity of parts, architectural description facilitates communication
 - To get a common understanding between the different stakeholders (users, clients, architect, designer,...)

Uses...

- Reuse
 - A method of reuse is to compose systems from parts and reuse existing parts
 - This model is facilitated by reusing components at a high level providing complete services
 - To reuse existing components, arch must be chosen such that these components fit together with other components
 - Hence, decision about using existing components is made at arch design time

Uses..

- Construction and evolution
 - Some structures in arch description will be used to guide system development
 - Partitioning at arch level can also be used for work allocation to teams as parts are relatively independent
 - During software evolution, arch helps decide what needs to be changed to incorporate the new changes/features
 - Arch can help decide what is the impact of changes to existing components on others

Uses...

- Analysis
 - If properties like performance, reliability can be determined from design, alternatives can be considered during design to reach the desired performance levels
 - E.g. reliability and performance of a system can be predicted from its arch, if estimates for parameters like load etc is provided

CSIS0521

Software Architecture

7

Architectural Views

- There is no unique arch of a system
- They are different *views* of a software system
- A view consists of *elements* and *relationships* between them, and describes a *structure*
- The elements of a view depends on what the view wants to highlight
- Diff views expose different properties
- A view focusing on some aspects reduces its complexity

CSIS0521

Software Architecture

8

Views...

- Many types of views have been proposed
- Most belong to one of these three types
 - Module
 - Component and Connector
 - Allocation
- The different views are not unrelated – they all represent the same system

CSIS0521

Software Architecture

9

Views...

- Module view
 - A system is a collection of code units
 - i.e. elements are modules, eg. Class, package, function, procedure,...
 - Relationship between them is code based, e.g. part of, depends on, calls, generalization-specialization,...

CSIS0521

Software Architecture

10

Views...

- Component and Connector (C&C)
 - Elements are run time entities called components
 - i.e. a component is a unit that has identity in the executing system, e.g. objects, processes, .exe, .dll
 - Connectors provide means of interaction between components, e.g. pipes, shared memory, sockets

CSIS0521

Software Architecture

11

Views...

- Allocation view
 - Focuses on how software units are allocated to resources like hardware, file system, people
 - i.e. specifies relationship between software elements and execution units in the environment
 - Expose structural properties like which process runs on which processor, which file resides where, ...

CSIS0521

Software Architecture

12

Views...

- An arch description consists of views of diff types, each showing a diff structure
 - Different system need different types of views depending on the needs
 - E.g. for performance analysis, allocation view is necessary; for planning, module view helps
- The C&C view is almost always done, and has become the primary view
 - We focus primarily on the C&C view

CSIS0521

Software Architecture

13

Component and Connector View

- Two main elements – components and connectors
- Components: Computational elements or data stores
- Connectors: Means of interaction between components
- A C&C view defines the components, and how they are connected through connectors
- The C&C view describes a runtime structure of the system – what components exist at runtime and how they interact during execution
- Is a graph; often shown as a box-and-line drawing
- Most commonly used structure

CSIS0521

Software Architecture

14

Components

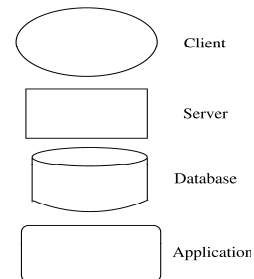
- Units of computations or data stores
- Has a name, which represents its role, and provides its identity
- A component may have a type; different types represented by different symbols in C&C view
- Components use ports (or interfaces) to communicate with others
- An arch can use any symbols to rep components; some common ones are shown

CSIS0521

Software Architecture

15

Some Component examples...



CSIS0521

Software Architecture

16

Connectors

- Interaction between components happen through connectors
- A connector may be provided by the runtime environment, e.g. procedure call
- But there may be complex mechanisms for interaction, e.g http, tcp/ip, ports,...; a lot of software needed to support them

CSIS0521

Software Architecture

17

Connectors...

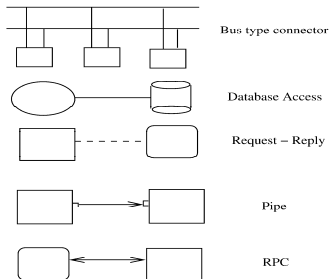
- Connectors need not be binary, e.g. a broadcast bus
- Connector has a name (and a type)
- Often connectors represented as protocol – i.e. components need to follow some conventions when using the connector
- Best to use different notation for different types of connectors; all connectors should not be shown by simple lines

CSIS0521

Software Architecture

18

Connector examples



CSIS0521

Software Architecture

19

An Example

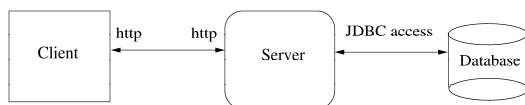
- Design a system for taking online survey of students on campus
 - Multiple choice questions, students submit online
 - When a student submits, current result of the survey is shown
- Is best built using web; a 3-tier architecture is proposed
 - Has a client, server, and a database components (each of a different type)
 - Connector between them are also of different types

CSIS0521

Software Architecture

20

Example...



CSIS0521

Software Architecture

21

Example...

- At arch level, details are not needed
- The connectors are explicitly stated, which implies that the infrastructure should provide http, browser, etc.
- The choice of connectors imposes constraints on how the components are finally designed and built

CSIS0521

Software Architecture

22

Extension 1

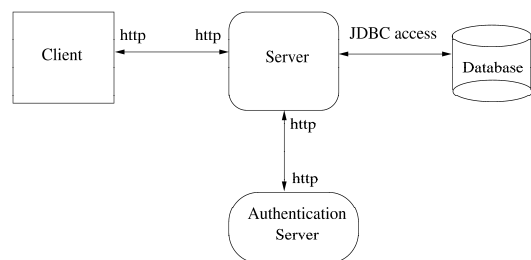
- This arch has no security – anyone can take the survey
- We want that only registered students can take the survey (at most once)
 - To identify students and check for one-only submission, need a authentication server
 - Need to use cookies, and server has to be built accordingly (the connector between server and auth server is http with cookies)

CSIS0521

Software Architecture

23

Extension 1...



CSIS0521

Software Architecture

24

Extension 2

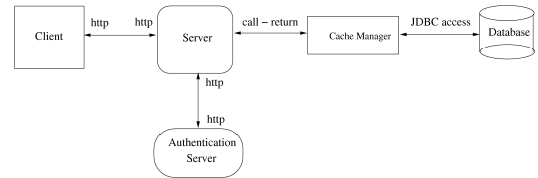
- It was found that DB is frequently down
- For improving reliability, want that if DB is down, student is given an older survey result and survey data stored
- The survey data given can be outdated by at most 5 survey data points
- For this, will add a cache component, which will store data as well as results

CSIS0521

Software Architecture

25

Extension 2...



CSIS0521

Software Architecture

26

Example...

- One change increased security, the other increased performance and reliability
- i.e. Arch level choices have a big impact on system properties
- That is why, choosing a suitable arch can help build a good system

CSIS0521

Software Architecture

27

Architectural Styles for C&C View

- Different systems have different C&C structure
- Some structures are general and are useful for a class of problems – architectural styles
- An arch style defines a family of archs that satisfy the constraint of that style
- Styles can provide ideas for creating arch for a system; they can be combined also
- We discuss a few common styles

CSIS0521

Software Architecture

28

Pipe and filter

- Well suited for systems that mainly do data transformations
- A system using this style uses a network of transforms to achieve the desired result
- Has one component type – filter
- Has one connector type – pipe
- A filter does some transformation and passes data to other filters through pipes

CSIS0521

Software Architecture

29

Example

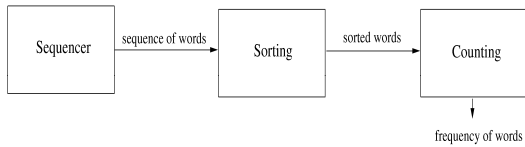
- A system needed to count the frequency of different words in a file
- One approach: first split the file into a sequence of words, sort them, then count the #of occurrences
- The arch of this system can naturally use the pipe and filter style

CSIS0521

Software Architecture

30

Example..



CSIS0521

Software Architecture

31

Shared-data style

- Two component types – data repository and data accessor
- Data repository – provides reliable permanent storage
- Data accessors – access data in repositories, perform computations, and may put the results back also
- Communication between data accessors is only through the repository

CSIS0521

Software Architecture

32

Shared-data style...

- Two variations possible
 - Black board style: if data is posted in a repository, all accessors are informed; i.e. shared data source is an active agent
 - Repository style: passive repository

CSIS0521

Software Architecture

33

Example

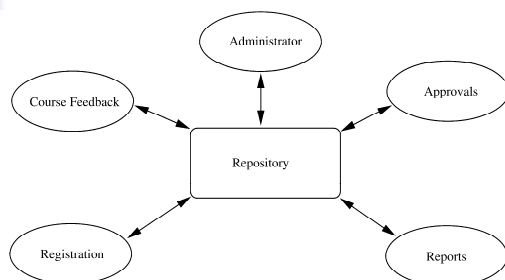
- A student registration system of a university
- Repository contains all the data about students, courses, schedules,...
- Accessors like admin, approvals, registration, reports which perform operations on the data

CSIS0521

Software Architecture

34

Example..



CSIS0521

Software Architecture

35

Example..

- Components do not directly communicate with each other
- Easy to extend – if a scheduler is needed, it can be added as a new accessor
 - No existing component needs to be changed
- Only one connector in this – read/write

CSIS0521

Software Architecture

36

Client-Server Style

- Two component types – clients and servers
- Clients can only communicate with the server, but not with other clients
- Communication is initiated by a client which sends request and server responds
- One connector type – request/reply
- Often the client and the servers reside on different machines

CSIS0521

Software Architecture

37

Client-server style...

- A general form of this style is the n-tier structure
- A 3-tier structure is commonly used by many application and web systems
 - Client-tier contains the clients
 - Middle-tier contains the business rules
 - Database tier has the information

CSIS0521

Software Architecture

38

Summary

- Arch of a software system is its structures comprising of elements, their external properties, and relationships
- Arch is a high level design
- Three main view types – module, component and connector, and allocation
- Component and connector (C&C) view is most commonly used

CSIS0521

Software Architecture

39

Summary...

- There are some C&C styles that are commonly used, e.g. pipe-and-filter, shared data, client server,....
- An arch description should document the different views and their relationship – views can be combined

CSIS0521

Software Architecture

40