SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 OF THE UGC ACT, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

Unit-3
Component Level Design

Types of Coupling

Presented By Dr.R.Kavitha,SoC

SASTRA University

Key concepts

Coupling and Cohesion are two key concepts in software engineering that are used to measure the quality of a software system’s design.

Modules are another key concepts in software engineering. A big project will decomposed into different parts.

Modularization

Modularization is the process of dividing a software system into multiple independent modules where each module works independently.

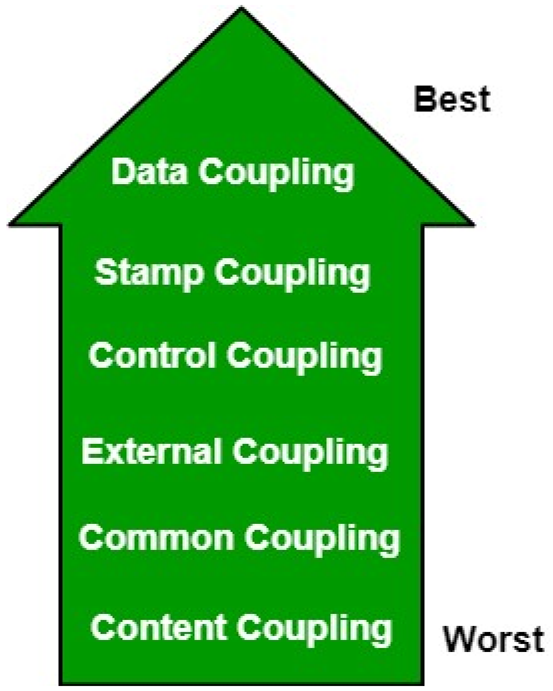
There are many advantages of Modularization in software engineering. Some of these are given below: Easy to understand the system.

- System maintenance is easy.
- A module can be used many times as their requirements. No need to write it again and again.

What is coupling in SwEngg?

- Coupling refers to the degree of interdependence between software modules. High coupling means that modules are closely connected and changes in one module may affect other modules(Ex. Window alignment).
- Low coupling means that modules are independent and changes in one module have little impact on other modules. (Ex: Menu design)

Hierarchical view of SW coupling types



Data Coupling

- If the dependency between the modules is based on the fact that they communicate by passing only data, then the modules are said to be data coupled. In data coupling, the components are independent of each other and communicate through data. Module communications don’t contain tramp data. Example-customer billing system.

Stamp Coupling

- In stamp coupling, the complete data structure is passed from one module to another module. Therefore, it involves tramp data. It may be necessary due to efficiency factors- this choice was made by the insightful designer, not a lazy programmer.

Control Coupling

- If the modules communicate by passing control information, then they are said to be control coupled. It can be bad if parameters indicate completely different behavior and good if parameters allow factoring and reuse of functionality. Example- sort function that takes comparison function as an argument

External Coupling

- In external coupling, the modules depend on other modules, external to the software being developed or to a particular type of hardware. Ex- protocol, external file, device format, etc.

Common Coupling

- The modules have shared data such as global data structures. The changes in global data mean tracing back to all modules which access that data to evaluate the effect of the change. So it has got disadvantages like difficulty in reusing modules, reduced ability to control data accesses, and reduced maintainability.

Content Coupling

- In a content coupling, one module can modify the data of another module, or control flow is passed from one module to the other module. This is the worst form of coupling and should be avoided.
- Ex: Payment gateway

Sequential Coupling

- Sequential coupling occurs when the output of one module is used as the input of another module, creating a chain or sequence of dependencies. This type of coupling can be difficult to maintain and modify.
- Ex. Payment & Delivery of POS

Temporal Coupling

- Temporal coupling occurs when two modules depend on the timing or order of events, such as one module needing to execute before another. This type of coupling can result in design issues and difficulties in testing and maintenance.
- Ex. OTP transfer

Communicational Coupling

- Communicational coupling occurs when two or more modules share a common communication mechanism, such as a shared message queue or database. This type of coupling can lead to performance issues and difficulty in debugging.
- Ex. Alert message to Bank and ACHolder

Functional Coupling

- Functional coupling occurs when two modules depend on each other’s functionality, such as one module calling a function from another module. This type of coupling can result in tightly-coupled code that is difficult to modify and maintain.

Ex. Line function calls theta function

Data-Structured Coupling:

- Data-structured coupling occurs when two or more modules share a common data structure, such as a database table or data file. This type of coupling can lead to difficulty in maintaining the integrity of the data structure and can result in performance issues.
- Ex. Block-chain structure

Interaction Coupling

- Interaction coupling occurs due to the methods of a class invoking methods of other classes. Like with functions, the worst form of coupling here is if methods directly access internal parts of other methods. Coupling is lowest if methods communicate directly through parameters
- Ex. Recursive calls,conditions

Component Coupling

- Component coupling refers to the interaction between two classes where a class has variables of the other class. Three clear situations exist as to how this can happen.
- Ex. Inheritance variables

THANK YOU