**EXPERIMENT NO. 9**

**COHESION AND COUPLING**

**Aim**

Draw Architecture diagram and incorporate Cohesion and Coupling for each module of University Exam Scheduling

**Description**

The aim of performing this experiment is to implement data flow architecture in your project and show type of cohesion between operations and coupling between components in your project.

For Good project design, Cohesion should be high and coupling should be s low as possible.

**Coupling**

* The degree of interdependence between two modules”
* We aim to minimize coupling - to make modules as independent as possible

**Types of Coupling**

High Cohesion

Data coupling

Stamp coupling

Control coupling

Hybrid coupling

Common coupling

Content coupling

* **Data Coupling**

Modules communicate by parameters

* **Data coupling problems**

Too many parameters - makes the interface difficult to understand and possible error to occur  
A composite data is passed between modules

* **Control coupling**

A module controls the logic of another module through the parameter

* **Hybrid coupling**

A subset of data used as control

* **Common coupling**

Use of global data as communication between modules

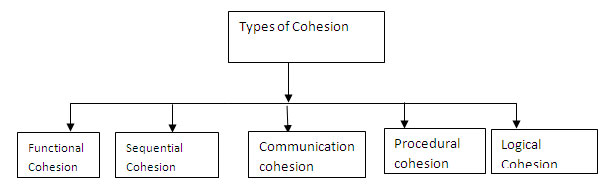
* **Content coupling**

A module refers to the inside of another module

**Cohesion**

* “The measure of the strength of functional relatedness of elements within a module”
* Elements: instructions, groups of instructions, data definition, call of another module
* Strong cohesion will reduce relations between modules - minimize coupling

**Types of Cohesion**

****

**Functional cohesion (Most Required)**

* All elements contribute to the execution of one and only one problem-related task

**Sequential cohesion**

* Elements are involved in activities such that output data from one activity becomes input data to the next

**Communicational Cohesion**

* Elements contribute to activities that use the same input or output data

**Procedural cohesion**

* Elements are related only by sequence, otherwise the activities are unrelated

**Temporal cohesion**

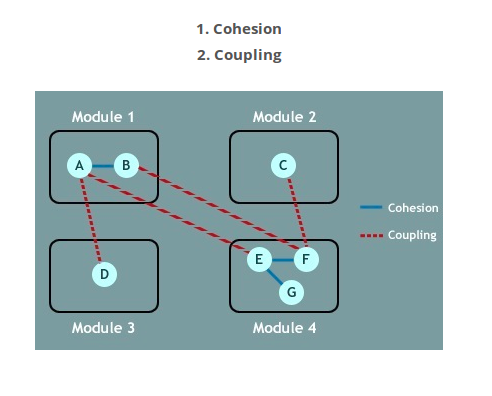
* Elements are involved in activities that are related in time

**Logical cohesion**

* Elements contribute to activities of the same general category

**Coincidental cohesion(Least Required)**

* Elements contribute to activities with no meaningful relationship to one another



**Conclusion**

Let us evaluate this architecture concerning some properties.

|  |  |
| --- | --- |
| **Criteria** | **Evaluation of the Proposed Architecture** |
| Change in some file  format | Good – change should impact only the filter  for that file. |
| Change in scheduling  algorithm | Good – only the scheduling component needs to be changed. |
| Adding new constraints for  scheduling | Good – only the scheduling component needs to be changed |
| Replacing files with GUI | Poor – switching to a GUI interface will require changing almost the complete implementation with this architecture. The  scheduling component can still be used. |

|  |  |
| --- | --- |
| Extension to web-based | Poor – This architecture does not support this change; will require a complete rewrite of the  system. |
| Provision of additional securities (passwords, etc) | Average – The architecture is not designed with security in mind. However, it is possible to add a security component for verification  in the start. |