Function-Oriented Design

Greg Butler

Computer Science and Software Engineering
Concordia University, Montreal, Canada

Email: gregb@cs.concordia.ca

Function-Oriented Design

Design with functional units which transform inputs to outputs

... consists of module definitions
with each module supporting a functional abstraction

... views a system as a set of modules
with clearly defined behaviour
— the services —
that interact with each other in a clearly defined manner
—the interface

Function-Oriented Requirements

Requirements = WHAT

Data processing applications focus on

- data input flows
- data output flows
- data stores
- *functional processes* transforming inputs to outputs

Modeling

In old days ... as data flow diagrams
Today ... using UML activity diagrams

Problem Partitioning and Hierarchy

Divide and Conquer

Decompose system into smaller and smaller pieces

Idealy, each piece can be designed separately

Ideally, each piece can be modified independent of other pieces

Each piece must communicate with other pieces

Modularity

A system is **modular** if it consists of discrete components
so that each component can be implemented separately
and a change to one component has minimal impact on other
components.

Each component needs to support a well-defined abstraction
... and have a specific interface
... that other modules use to interact with it

Jalote, 1991, An Integrated Approach to Software Engineering
"Modularity is where abstraction and partitioning come together."

Stepwise Refinement

Stepwise Refinement

… a top-down approach that starts with the system as a whole,
and decomposes it into subcomponents
that exist at lower levels of abstraction.

Module

A module is a logically separable part of a program.

A module is a program unit that is discrete and identifiable with respect to compiling and loading.

A module can be a function, a procedure, a process, or a package.

Cohesion

Cohesion is a concept that tries to capture intra-module bonds.

Cohesion shows how closely the elements of a module are related to each other.

A highly cohesive module has attributes and behaviour that relate only to one task or concern

compare to the concept of *responsibility*

Coupling

Coupling is *"how strongly"* different modules are dependent.

Definition

"Coupling between modules is the strength of interconnections between modules
or
a measure of interdependence among modules."

... not easy to quantify
We talk about *"loose coupling"*

Information Hiding

Information Hiding, David Parnas 1972

Hide things that are likely to change behind interfaces
as internal secrets of a module

Encapsulation

... the **only way**
.... to access information in a module is using the interface