ISM 50 - Business Information Systems

Lectures 14

Instructor: John Musacchio
UC Santa Cruz
November 10, 2009
Class announcements

- For Next Class
  - Read: MySQL Database Case

- For next Tuesday
  - Assignment 4

- Database tutorials
Student Presentations
**Implementation**

Module B

Compute Mean and Variance

![Diagram](image)

**Implementation 1:**

- **SUM** = \( \sum_{i=1}^{N} x_i \)
- **MEAN** = \( \frac{\sum_{i=1}^{N} x_i}{N} \)
- **VARIANCE** = \( \sum_{i=1}^{N} (x_i - \text{MEAN})^2 \)

- **Should he use it?**
  - **NO!!! Why??**

- Either A should compute “SUM” himself, or sit down with B and redesign the interface
Encapsulation

The designer of B might take measures to hide “SUM” from A so that A is not able to violate the agreed interface.

- Example: B does not declare “SUM” as a global variable.

Making a modules implementation details inaccessible to other modules is called **encapsulation**.
This simple interface example allows for only one action of module B.
- Action is “Compute mean and variance.”

Other examples are possible.
Possible software interface

Menu of actions

Example:
Action 1: Compute mean
Action 2: Compute variance
Action 3: Compute mode
Etc..
Protocol

In addition to atomic actions, an interface may define protocols

- Protocol $\equiv$ finite sequence of actions required to achieve a higher level function
- One action can be shared by multiple protocols
- Multiple modules may participate in a protocol
Hello: I'm the HHC of Airplane#1234

Hello: I'm the gate 32 server

These were the unruly passengers on last flight

"Passengers noted"

Tell me about the passengers of my next flight

Return Passenger Data

(Might be passed as an array of a compound data type "passenger," which in turn is composed of standard types like integer, and string)