ISM 50 - Business Information Systems
Lecture 13
Instructor: John Musacchio
UC Santa Cruz
November 5, 2009

Announcements
- 2nd Folio Article due today
  - Read
    - Messerschmitt Ch 11.1 - 11.2 (325-335)
    - Messerschmitt Ch 15.1 - 15.2 (415-425)
- Guest Speaker Tuesday
  - Mike Quinn, Senior Vice President Cisco

Student Talks
Amy Lin (Business paper)
Meredith Yaxley (Business paper)

Architecture Example

Architecture
- Conceptualization
  - What is it you are trying to do?
- Example Concept:
  - Small HHC for flight attendants.
  - HHC tells flight attendants which passengers are higher priority.
    - Who paid the highest fares
    - Who has been a more valuable customer in past
  - Flight attendant discriminates based on this
    - Free drinks, meals, and pillows to valuable customers
    - Ignore less valuable customers

Example Concept:
Architecture

- How do you begin to architect a solution for a problem like this?
- Break it into modules!

HHC Architecture

- When a module is composed of sub-modules, the architecture is hierarchical.

Granularity tradeoff.

- How big should we make the modules
  - Many simple small ones
  - Or a few complicated big ones...
- This aspect of modularity is called granularity.
- Which is better?
Our architecture makes use of the existing interface of the airline database, so we don't need to redesign it!

Interface specifications are often made precise by using data types.

- Example type: float
  - A number with a decimal place
  - Has a certain allowable range, and precision.

More on Data types

- Data passing an interface is often specified in terms of a limited number of standard data types
- Data type = range of values and allowable manipulation
- Data type does not presume a specific representation, to allow heterogeneous platforms
  - Representation must be known when data passes a specific module interface

Example data types

- Integer
  - "natural number between -32,767 and +32,768"
  - Could be represented (in many ways) by 16 bits
    - since $2^{16} = 65,536$

- Float
  - "number of the form $m \times 10^n / 32768$, where $m$ is in the range -32.767 to +32.768 and $n$ is in the range -255 to +256"
  - Could be represented by $16 + 8 = 24$ bits
  - Could be represented by 16+8 = 24 bits

More data types

- Character
  - "values assuming a-z and A-Z plus space and punctuation marks"
    - could be represented by 7 or 8 bits

- Character string
  - "collection of $n$ characters, where $n$ is customizable"
    - could be represented by $7 \times n$ bits
**Compound data types**

**Programmer-defined composition of basic data types**

**Example:**

```java
Employee {
    String name;
    String address;
    Integer year_of_birth;
    etc.
}
```

**Interfaces**

- **PARAMETERS:**
  - N numbers of Float type

- **REturns:**
  - 2 Numbers of float type that signify: Mean, Variance

- **Computations of key statistics**

- **MODULE A**
  - **Compute Mean and Variance**
    - **MEAN** = \( \frac{\sum x_i}{N} \)
    - **VARIANCE** = \( \frac{\sum (x_i - \text{MEAN})^2}{N} \)

- **HIDDEN From Module A!!**

- **MODULE B**
  - **Compute Mean and Variance**
    - **SUM** = \( \sum x_i \)
    - **MEAN** = \( \frac{\text{SUM}}{N} \)
    - **VARIANCE** = \( \frac{\sum (x_i - \text{MEAN})^2}{N} \)

**Implementation**

- **Module A**
  - **Computation of key statistics**

- **Module B**
  - **Computation of key statistics**

- **Module A**
  - \( x_i, i = 1..N \)
  - **MEAN**, **VARIANCE**

**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**.

- **One module should not be concerned with other module’s implementation**
  - "Separation of concerns."

- **One module should see the other only through its interface – implementation details hidden.**
  - "Abstraction"

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

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