Topic # 10

Structuring System Requirements: Conceptual Data Modeling
Objectives

1. Entities: Data objects
2. Entities: Class objects
3. Unary, Binary, and Ternary relationships
4. Cardinality
5. Entity-Relationship (E-R) diagrams
Conceptual Data Modeling

• **Conceptual Data Model** is a detailed model that
  1) captures the **overall structure of organizational data and data flow**, and
  2) is **independent of any database management system** or other implementation considerations.

• Some systems developers believe **that a data model is one of the most important parts of the statement of information system requirements** for three reasons:
  1) completely representing data requirements is crucial for the design of databases, programs, computer screens, and printed reports—critical elements of any information system;
  2) data rather than processes are the most complex aspects of many information systems, and hence must be modeled with clarity;
  3) data characteristics and natural structures (as opposed to processing requirements) are reasonably permanent, so designing information systems based on data yields more stable systems with longer lives (and less maintenance).

*Example: Online Course – Data; BlackBoard or Sakai LMS - Rules*
Entities: Data and Class Objects
A data object contains a set of attributes that act as an aspect, quality, characteristic, or descriptor of the object.

**Class Object Name:** Student

**Class Object Attributes:**
- first_name
- last_name
- year_of_admission
- major
- courses_taken
- credits_obtained
- home_address
- phone_number
- email_address
- etc.

**Class Object Functions/Methods:**
- register for a course
- pay for a course
- get individual course schedule
- etc.

The object encapsulates both data and the logical procedures required to manipulate the data.
Data Objects: examples

- external entities: (printer, user, sensor)
- things: (e.g., reports, displays, signals)
- occurrences or events: (e.g., interrupt, alarm)
- roles: (e.g., manager, engineer, salesperson)
- organizational units: (e.g., division, team)
- places: (e.g., manufacturing floor)
- data structures: (e.g., employee record, file, etc.)
Class Diagram

• Represent: 1) objects system manipulates 
2) operations applied to objects, and 
3) collaborations occurring between classes

• Elements of class model include:
  1) data objects 
  2) attributes 
  3) operations 
  4) collaboration diagrams, etc.

• Examine the problem statement and try to find nouns that fit the following 
categories and produce or consume information (i.e. grammatical parse) 
  – External entities (systems, devices, people) 
  – Things (e.g. reports, displays, letters, signals) 
  – Events occurring during system operation 
  – Roles (e.g. manager, engineer, salesperson) 
  – Organizational units (e.g. division, group, team) 
  – Places 
  – Structures (e.g. sensors, vehicles, computers)
Class Diagram

Class Objects

Attributes
Operations

Associations: enrolled, on waiting list, etc.

Source: http://www.agilemodeling.com/artifacts/classDiagram.htm
Class Diagrams:
examples on Bradley University campus
Relations.
ERD Diagrams.
Entity-Relationship (E-R) Modeling

• Entity-Relationship (E-R) Diagram
  – A detailed, logical representation of the entities, associations and data elements for an organization or business

• Notation uses 3 main constructs (see corresponding graphic symbols below):
  – Data entities
  – Relationships
  – Attributes
Degree of Relationship

- Degree: number of entity types that participate in a relationship
- Three cases
  - Unary: between two instances of one entity type
  - Binary: between the instances of two entity types
  - Ternary: among the instances of three entity types
In-classroom practice:

**Bradley University campus: examples of 1) 1-to-1, 1-to-many, and many-to-many relationships**
Cardinality

- **Cardinality**: the number of instances of entity B that can or must be associated with each instance of entity A

- **Minimum Cardinality**
  - The minimum number of instances of entity B that may be associated with each instance of entity A

- **Maximum Cardinality**
  - The maximum number of instances of entity B that may be associated with each instance of entity A

- **Mandatory vs. Optional Cardinalities**
  - Specifies whether an instance must exist or can be absent in the relationship
An Example of Conceptual Data Model Diagram
Examples of Unary Relationship (Bradley University campus)
Example 1:
Mandatory Cardinalities

Example 2:
One optional, One Mandatory Cardinalities
ERD (E-R Diagram): university-related examples
Topic # 10

Structuring System Requirements: Conceptual Data Modeling

Additional information: UML
Object Modeling Using Class Diagrams

UML Language

• Object-oriented approach

• Features
  – Objects and classes
  – Encapsulation of attributes and operations
  – Polymorphism
  – Inheritance
UML associations are analogous to E-R relationships.

UML multiplicities are analogous to E-R cardinalities.

Examples of association relationships of different degrees
An association with its own attributes, operations, or relationships

UML association classes are analogous to E-R associative entities.
Example of generalization, inheritance and constraints

Generalization and inheritance implemented via superclass/subclasses in UML, supertypes/subtypes in E-R
Final E-R Diagram for Hoosier Burger's Inventory Control System