CIV1283
Civil Informatics

Tamer E. El-Diraby, PhD.
What do we mean by a Model?

Source: Prof. Ziga Turk
Modeling Tools

- IDEF0: Process
- UML: Software Systems
E-Construction Framework

Construction Environment & Challenges

Information Systems

IT

Traditional
Semantic

Process Models
Languages
Ontologies

Collaborative Projects
Virtual Processes
e-Legal
Knowledge Management
Information Culture

Modeling tools

Capturing User Intent
Requirement Analysis
Planning
Design
Procurement
Progress: form **Data to Knowledge** and from **Product to Actors**
Bo-Christer Bjork (2002). A formalised model of the information and materials handling activities in the construction process, *Construction Innovation 2002; 2: 133–149*
IDEF0 in Asset Management

Hassanain, Froese, Vanier, Implementation Of A Distributed, Model-based Integrated Asset Management System
IDEF0 for Land Development

- Acquisition Process
  - Purchase Agreement
- Planning Process
  - Preliminary schedule
  - Preliminary Budget
  - Approved draft plan
- Design Process
  - Proposed M-Plan
- Registration Process
  - Bid documents
  - Final design
- Procurement Process
  - All permits
  - Approved M-Plan
- Construction Process
  - Construction contracts

- Company conditions
- Market conditions
- Boundary Survey
- Topographic Survey
- Holdout Survey
- Market area land ownership map
- Sales department
- Permits department
- Architecture department
- Accounting department
- Contracts department
- Construction department
- Land development team
- Consultant team
- Lawyers

- City Standards
- City by-laws
- Company strategy

- City by-laws
- Built environment
- As-built survey
IDEF0 for OSP

- Investment strategy
- Project Planning
- OSP Design
- OSP Constr.
- Telecomm. design
- Procurement (Procur.)
- Design requirements
- Dwgs and Specs
- Contract
- As-Built Drawings
- Knowledge
- Built Facility
- Maintenance & Operations

- Market conditions
- Company strategy
- Financial Resources
- Codes & Regulations
- Software
- Organizational teams
- Design theories
- ISP design
- As-Built Drawings
- Knowledge
- Built Facility
- Maintenance & Operations

- Dwgs
- Specs
Design A Glass Wall

Diagram showing the process steps involved in designing a glass wall, including:

1. **Perform Preliminary Design**
   - Requirements
   - Project data
   - Original project data
   - Preliminary design package
   - Review comments (F)
   - Approved design concept

2. **Perform Detailing Design**
   - Detailing design package
   - Review of detailing design
   - Approved design drawing

3. **Perform Fabrication Design**
   - Fabrication drawing
   - Fabrication drawing & Prod. Info.
   - Contractor's design staff

Steps 1 to 3 are connected by arrows indicating the flow of work. The diagram also includes notes on cost efficiency, available resource feedback, quality control, constructability, schedule, budget, methodology, and professional standards.
Information Flow
Example: textile Industry Supply Chain

Chapman, Lathon, Petersen, *Demand Activated Manufacturing Architecture DAMA-I-1-01 Version 1.0*. Sandia National Laboratories
Example
LOGISTICS FOR CONSIGNEE, CONSOLIDATOR SHIPPER AND CARRIER
The Pizza Example
Let Us Model a Human Being to the Computer
Concepts of Object Orientation

- Entity
- Attributes
- Relationships
- Cardinality
- Generalization / inheritance
Concepts of OO

• Entity
  • Group of persons, places, objects, or concepts about which data is captured and stored
  • Entity instance is a single occurrence of the entity

• Attributes
  • Descriptive property or characteristic of an entity
  • Specific pieces of data stored about each instance of a given entity

• Relationship
  • An event that links entities
Cardinality

• Defines the maximum and minimum number of occurrence of one entity for a single occurrence of the related entity
• Defined in both directions for every relationship
• Three types
  – One-to-one
  – One-to-many
  – Many-to-many
Relationships

association

- binary
  - following_direction
  - opposite_to_direction
- unidirectional
- aggregation
  - whole_of_part
  - part_of_whole
- composition
  - c_whole_of_part
  - c_part_of_whole
Generalizations

• Commonalities between entities
• Attributes common to several types of an entity grouped into an entity called ‘supertype’
• ‘supertype’ has one or more one-to-one relationships with entity ‘subtypes’
• ‘is a’ relationship between ‘supertype’ and ‘subtype’
Object Orientation

- Encapsulation
- Inheritance
- Polymorphism

Object

Instant of (have values)

Class

Specify

Interface

Implements
Object Orientation

• OO: Encapsulation, Polymorphism, and Inheritance

• The older structured world concentrated on sequence, selection, and iteration

• A software packet that abstracts the salient behavior and characteristics of a real object into a software package that simulates the real object
  – Objects have characteristics and behavior encapsulated in it
  – Objects communicate by messages
  – Objects can inherit characteristics and behavior
## Basics of Objects

Encapsulation of objects:

<table>
<thead>
<tr>
<th>Example</th>
<th>Characteristic</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>5</td>
<td>add</td>
</tr>
<tr>
<td>List Box</td>
<td>location</td>
<td>drop down</td>
</tr>
<tr>
<td>Window</td>
<td>size</td>
<td>open</td>
</tr>
<tr>
<td>ATM</td>
<td>amount on hand</td>
<td>give cash</td>
</tr>
<tr>
<td>Customer</td>
<td>balance</td>
<td>pay bill</td>
</tr>
</tbody>
</table>
Encapsulation:

• The containment of the data behind a software membrane consisting of methods. The data can only be accessed through the encapsulated behavior.
Basics of Objects

Message:

• A signal from a client object requesting services from a server object
• The message may contain arguments
• The server object may return a response
Basics of Objects

Polymorphism:

• Messages mean different things to different objects:
  Print Word document
  Print Excel document

• This means different implementations can be hidden behind a common interface
Basics of Objects

Class:

• A Collection of like objects
• A template for defining new object instances
• The behaviors reside in the class
• Behavior is implemented by methods
Instance:

- A term used to refer to an software object
- It is a common synonym of object
Inheritance:

• A technique to allow classes to use a parent classes methods and data

• Inheritance can have many levels
The Ten Big Definitions

- **Object**: A software package
- **Method**: An object's procedure
- **Message**: A signal from one object to another
- **Class**: A template to create objects
- **Subclass**: A special case of a class
- **Instance**: An object's other name
- **Inheritance**: A mechanism to use another object's innards
- **Encapsulation**: Data & methods together
- **Abstraction**: Capturing behaviors & characteristics
- **Polymorphism**: Hiding alternative methods behind a common interface
What is Hierarchy?

Increasing abstraction:

- Asset
  - BankAccount
    - Savings
    - Checking
  - Security
    - Stock
  - RealEstate
    - Bond

Decreasing abstraction:

*Classes at the same level of the hierarchy should be at the same level of abstraction*
Basic Concepts of Object Orientation

- Object
- Class
- Attribute
- Operation
- Component
- Generalization
- Polymorphism
Registration Process

- Residential registration process
- Commercial Registration process
- Mixed-use Registration process

Survey certification process
- M-plan approval process
- Agreement approval process

Canadian National Railway approval process
- Heritage approval process
- Ministry of Transportation approval process
- Pipeline approval process
- School board approval process
- Subdivision approval process
- Ministry of the Environment approval process
Basic Structure of UML

- Activity Diagrams
- Use Case Diagrams
- Sequence Diagrams
- Collaboration Diagrams
- Deployment Diagram
- Class Diagrams
- Object Diagrams
- State Diagrams
- Component Diagrams
UML Architecture

Analysis

Define Entities
- Parameters
- Functions
- Messages
- Class
- Object
- Interface

Define Relationships
- Association
- Aggregation
- Dependency

Modeling

Presentation

Use Case
- Class Diagram
- State Diagram
- Activity Diagram
- Component Diag.
- Deployment Diag.
Design Cycle

- Manage Requirements
- Develop Iteratively
- Model Visually
- Verify Quality
- Use Component Architectures
- Control Changes
Design Sequence

Architect

Architectural Analysis

Architectural Design

Describe Concurrency

Describe Distribution

Review the Architecture

Architecture Reviewer

Use-Case Analysis

Subsystem Design

Use-Case Design

Class Design

Review the Design

Design Reviewer

Database Design

Database Designer

Designer
Use Cases for a Cash Machine

An actor is someone or something outside the system that interacts with the system.

A use case is a sequence of actions a system performs that yields an observable result of value to a particular actor.
Example: Use-Case Diagram

- **Student**
  - Login
  - View Report Card
  - Register for Courses
  - Select Courses to Teach

- **Professor**
  - Select Courses to Teach

- **Registrar**
  - Login
  - View Report Card
  - Maintain Student Information
  - Maintain Professor Information
  - Course Catalog

- **Billing System**
  - Close Registration

- **Submit Grades**
Use Case Example
Use Case

Assess Risk

Sensitivity Analysis

Financial Analysis

C/B Analysis

LCCA

Report To Owner

Project Manager
Flow of events for the Withdraw Money Use Case

1. The use case begins when the client inserts her ATM card. The system reads and validates information on the card.
2. The system prompts for the PIN. The system validates the PIN.
3. The system asks which operation the client wishes to perform. The client selects “Cash withdrawal.”
4. The system requests the amount. The client enters the amount.
5. The system requests the account type. The client selects checking or savings.
6. The system communicates with the ATM network . . .
Benefits of a Use-Case Driven Process

• Use cases are concise, simple, and understandable by a wide range of stakeholders
  – End users, developers and acquirers understand functional requirements of the system

• Use cases drive numerous activities in the process:
  – Creation and validation of the design model
  – Definition of test cases and procedures of the test model
  – Planning of iterations
  – Creation of user documentation
  – System deployment

• Use cases help synchronize the content of different models
Checkpoints: Use-Case Model

- Is the use-case model understandable?
- By studying the use-case model, can you form a clear idea of the system's functions and how they are related?
- Have all the actors been identified? Have all functional requirements been met?
- Does the use-case model contain any superfluous behavior?
- Is the division of the model into use-case packages appropriate?
Example: Activity Diagram

1. Select Course
2. Check Schedule
3. Check Pre-requisites
4. Assign to Course
5. Update Schedule

- [checks completed] [student added to course]
- [checks failed] Resolve Conflict
Example: Sequence Diagram
1. Submit Knowledge Item

2. Transfer Knowledge Item

3. Extract keywords

4. Submit a list of keywords.

5. Build list of concepts

6. Return the list of concepts matching the keywords

7. Build Knowledge representation

8. Transfer KR

9. Index KR

A statistical weight is calculated for each concept

An ontological weight is calculated for each concept
1. Log in

2. Check login/password

3. User logged

4. Click on "Search concepts with keyword" item

5. Enter a list of keyword

6. Submit query

7. Return the list of concepts holding the keywords

A weight is performed for each concept
Example: Collaboration Diagram
Sun, Aouad, Bakis, Swan INTEGRATED DESIGN SYSTEM TO SUPPORT PARTNERING PRACTICE IN THE WATER INDUSTRY
Modeling Domains
Material Process

Bo-Christer Bjork (2002). A formalised model of the information and materials handling activities in the construction process, *Construction Innovation 2002; 2: 133–149*
Bo-Christer Bjork (2002). A formalised model of the information and materials handling activities in the construction process, *Construction Innovation 2002; 2: 133–149*
Information Search & Retrieval

Owner planning

Soil conditions
Survey
Site layout
Community

Prelim Design

Detailed Design

Final cost estimate
Bidding

Risk assessment