What is Model & Modeling? [1]

Definition of MODEL

- a three-dimensional representation of a person or thing, typically on a smaller scale
- (in sculpture) a figure made in clay or wax which is then reproduced in a more durable material
- something used as an example
- a simplified mathematical description of a system or process; used to assist calculations and predictions
- an excellent example of a quality
- a person employed to display clothes by wearing them
- a person employed to pose for an artist
- a particular design or version of a product

Oxford Dictionary
What is Model & Modeling? [2]

Definition of MODEL
- A simplified representation of a limited part of reality with related elements
  [FAO(1996), Agro-ecological zoning; guidelines. FAO Solis Bulletin, 73]

Definition of Modeling
- The construction of physical, conceptual or mathematical simulations of the real world. Models help to show relationships between processes (physical, economic or social) and may be used to predict the effects of changes in land use
  [FAO(1993), Guidelines for land-use planning. FAO Development Series, 1]

Visit the UMTP(UML Modeling Technology Promotion) International Association Homepage
http://www.tiu.ac.jp/org/forum-01/index.html

Model & Modeling

The modeling technology became more important in the societies where they can be easily connected each other through the internet, to convey semantics of information or software systems to others in different domains or countries, penetrating the uniqueness of culture, language, etc.
Normative modeling facilities can enhance sharability and reusability of models in IT industries, as follows;
- Common languages for describing models (modeling language)
- Common constructs to be used in the modeling (modeling constructs)
- Common view points for describing models (modeling views)
Model

- A representation of a business or organization
  - Structured technique
  - Examine various aspects of a business
  - Graphical representation
  - Text description supports Graphics
- A framework for structured analysis
- A standard format for communicating the results of analysis

IDEF


- Build a model of real system
- Analyze the built model
- Design a TO-BE model
- Transition it to the real system
A hammer is designed to pound nails. It can also be used to put in a screw or make a hole in something. But is it right tool for the job (screwing or drilling)?

Purpose to use IDEF

- To document
- To communicate (Consensus)
- To educate
- To improve understanding
- To capture and analyze business system and processes
- To design new business system before coding or programming
IDEF family

- IDEF0 – Function Modeling (Analysis)
- IDEF1 – Information Modeling (Analysis)
- IDEF1x – Data Modeling (Design)
- IDEF3 – Process Modeling (Analysis & Design)
- IDEF4 – Object-oriented design (Design)
- IDEF5 – Ontology Description Capture (Analysis)
- IDEF6 – Design Rationale Capture
- IDEF7 – Information System Audit Method
- IDEF8 – Human-System Interaction Modeling
- IDEF9 – Business-Constraint Discovery Method (Analysis)
- IDEF10 – Implementation Architecture Modeling
- IDEF11 – Information Artifact Modeling
- IDEF12 – Organizational Design Method
- IDEF13 – 3-Schema Architecture Design Method
- IDEF14 – Network/Distribution Design Method

IDEF Modeling

- **Purpose (Objective)**
  - The reason a model is created

- **Viewpoint (Bias)**
  - Establishes the goal of the communication intended by the model
  - The perspective from which a subject is analyzed
  - Specifies how the model will be used

- **Context (Subject)**
  - Establishes the subject of a model
  - The boundaries of the subject matter
**IDEFO – Function Modeling**

- **Motivation**: manufacturing system function modeling
- **Origin**: Structured Analysis and Design Technique (SADT) of SofTech
- **June 1981, ICAM Function Modeling Manual (IDEF0)**

Representation of “What Do I Do ?”
Boxes = activity, action, process, operation
Arrows = relationship between activities

**IDEF1 – Information Modeling**

- **Motivation**: Capturing information, object of Manufacturing system’s function
- **Developed by**: Hughes Aircraft and D. Appleton Company
- **June 1981, ICAM Information Modeling Manual (IDEF1)**

Representation of “What Must I Know to Do What I Do ?”
Boxes = objects
Arrows = relationship between objects
IDEF1x – Data Modeling

- For database design
- Under ICAM Program, Integrated Information Support System (IISS) Project
- Origin: General Electric’s Logical Database Design Technique (LDDT)
- December 1985, IISS Information Modeling Manual (IDEF1x)

Representation of Data Relationships and Business Rules

- Boxes = entities
- Arrows = relationship between entities

IDEF1 vs. IDEF1x

<table>
<thead>
<tr>
<th>IDEF1</th>
<th>IDEF1x</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis method</td>
<td>Design method</td>
</tr>
<tr>
<td>Used by system analysts</td>
<td>Used by database designers</td>
</tr>
<tr>
<td>Defines information Requirements</td>
<td>Defines design criteria</td>
</tr>
<tr>
<td>Technology independent</td>
<td>Tied to relational technology</td>
</tr>
<tr>
<td>Results in a single requirements model</td>
<td>Given requirements, multiple design models are possible</td>
</tr>
</tbody>
</table>
IDEF3 – Process Modeling

- Motivation: Describe some activity step-by-step or by cause-effect
- Under Information Integration for Concurrent Engineering (IICE) Program
- Developed by Knowledge Based Systems, Inc. (KBSI)
- May 1992, IICE IDEF3 Process Description Capture Method (IDEF3)

Representation of “How things are done”
Boxes = functions/activities
Arrows = relationship between activities

IDEF0, IDEF1/1x, IDEF3

- **IDEF0**
  - What do you do
  - Functional dependencies
  - No time
  - Used to “target” activities that need improvement

- **IDEF1/1x**
  - What do you need to know
  - Information Management or Database Design
  - Information or Data Requirements

- **IDEF3**
  - How do you do it
  - Precedence (time) or Cause-Effect
  - Reduce Cycle Time
  - Improve a specific process
IDEFO & IDEF3

IDEFO Modeling before IDEF3 modeling
- When explicit sequence, procedure, process are unknown
- 면접자가 무엇을 하는지는 아는데 어떻게 하는지 모를 때
- 발견된 activities 사이에 명확한 구분이 없을 때
- 업무절차보다 정책을 입안할 때

IDEF3 Modeling before IDEF0 modeling
- 문제상황이 시간에 상관관계를 가질 때
- 전후관계가 명확할 때
- 면접자가 자기 업무를 명확하게 파악할 때

Benefits from using IDEF

시스템 분석 및 설계를 위한 IDEF 적용은 초기에 더 많은 Resource를 필요로 하나 조사분석 및 예비설계 단계에서의 상호 Communication 증대를 통하여 시스템 구현 단계에서 Resource를 줄일 수 있음
IDEF is a language

Function Modeling with IDEF0
Function Model

A representation of the activities/functions and relationships between activities/functions in an existing or planned system
- Identify What System Does
- Provide functional foundation for systems development
- Provide foundation for modeling Data
- Reduce the Time and Cost of Analysis

Definition of Input, Control, Mechanism, Output

ICOM concept

- **Controls** (required) - Trigger to the Activity
  Conditions that govern the function
  Every activity must have at least one control
  (When a component is confused whether it is Input or Control, regard it as a Control)

- **Inputs** - Item(s) changed by the function
  Each activity may not have an input

- **Function, Action, Activity, Process**

- **Outputs** (required) - Item(s) produced by the function

- **Mechanisms** - People, machines, systems that perform the function
Activity

- An action, function, or operation. Represented by a box and labeled as a verb phrase.

Activity (Verb Phrase)

Input

- Any real object or data needed to perform an activity. Transformed through the completion of the activity.

Input (Verb Phrase)
**Output**

- Results from the completion of the activity.

**Control**

- Directs, guides, or initiates the activity. May also combine in some way with input(s) to result in an output.
Mechanism

- Indicates how the activity is accomplished.

Activity (Verb Phrase)

Control

Input → Output

Mechanism

Minimal Requirements

Control

Activity (Verb Phrase) → Output
Components of IDEF0 Model

- **Diagrams**
  - Boxes and arrows used to graphically describe all or part of the business
- **Text**
  - Written description of important aspects of the business; usually includes a narrative description of each diagram
- **Glossary**
  - Definitions of the information represented in the diagrams: includes a definition of each activities and arrow
- **Node Tree**
  - A structured list of diagrams in the model

Beginning of Modeling

- **Define Scope/Context**  
  (Business Boundaries)

- **Define Purpose**
  - What is the goal or interest of this model that is of critical importance to its success?

- **Define Viewpoint**
  - Who is the audience or person that is viewing this model for correctness?
  - From which viewpoint do we wish to model this business?
The context defines the boundaries of your model or what you will include in the model.

Employee/Position Data comes from outside the model.

Scopes the model and defines the boundaries
- If the scope is too big, the model becomes too complex and resource-intensive
- If the scope is too small, the model becomes trivial

Determining the context is the most critical step in Activity/Function Modeling.
Purpose:
The purpose defines the reason to develop this particular activity model.

To document the activities associated with managing Personnel Actions and identify non-value added activities that might be eliminated.

Viewpoint:
The viewpoint describes the perspective of the person or group developing the model.

Viewpoint: Personnel Officer
Beginning of Modeling

Example Top-level Diagram

Decomposition

Used to further define an activity by dividing it into its sub-activities

“Parent” Activities는 “Children”보다 더 추상화된 것

참고 : 한 Diagram당 3~6개의 Activity를 가도록 할 것
Numbering Diagrams

Topics are divided into pieces in a structured manner
How the pieces fit together can be determined from node numbers

Node number of detail diagram = Diagram node number + Box number
e.g.) A42 = A4+2

NOTE: Node numbers shown here indicate that the box has been detailed. The C-number or pign number of the child diagrams could have been used instead of the node number.
**Branching/Joining Arrows**

<table>
<thead>
<tr>
<th>Graphic</th>
<th>Interpretation</th>
<th>Graphic</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Branching Arrow" /></td>
<td>means A</td>
<td><img src="image2" alt="Joining Arrow" /></td>
<td>means A and B</td>
</tr>
<tr>
<td><img src="image3" alt="Branching Arrow" /></td>
<td>means A</td>
<td><img src="image4" alt="Joining Arrow" /></td>
<td>means A and B</td>
</tr>
<tr>
<td><img src="image5" alt="Branching Arrow" /></td>
<td>means A</td>
<td><img src="image6" alt="Joining Arrow" /></td>
<td>means A and B</td>
</tr>
</tbody>
</table>

**Connection between Boxes**

- **Tax Requirements**
- **Ordered Product**
- **Account Clerk**

<table>
<thead>
<tr>
<th>Block</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEEP RECORDS 1</td>
<td>This fork means that “Files” contains “Customer Records” (needed by Function 2) and “Price &amp; Tax Tables” (needed by Function 3).</td>
</tr>
<tr>
<td>DELIVER PRODUCTS 2</td>
<td>This join means “Account Entries” are created by “Deliver Products” and “Do Billings”.</td>
</tr>
<tr>
<td>DO BILLING 3</td>
<td>Price &amp; Tax Tables</td>
</tr>
<tr>
<td></td>
<td>Account Entries</td>
</tr>
</tbody>
</table>
Bundling / Unbundling

Parent의 Concept이 Child diagram에서 분해/취합되는 현상

Tunneling

Tunneled Arrows at Connected Ends
(Concept Does Not Appear on the Next Lower Level.)

Tunneled Arrows at Unconnected ends
(Concept Does Not Appear on the Next Higher Level.)
Basic Rules

- Excluding the A-0 diagram which has only one Activity Box, all other diagrams should have no less than 3 and no more than 6 Activity boxes.
- Each activity box must have at least one control and one output, but no more than 6 of each type of ICOM.
- An activity shall have zero or more input arrows.
- Every diagram in a model must adhere to the model’s overall viewpoint, purpose, and context.
- Constraints do not indicate “how” or “when”
Application of IDEF0

- Enterprise Analysis (both “AS-IS” and “TO-BE”)
- Software Design
- Knowledge Acquisition and Engineering
- Project Planning
- Information Resource Management
- Concurrent Engineering Planning
- Critical Success Factor Definition
- Simulation Requirements Definition

Process Modeling with IDEF3
Process Modeling

- Process = Activity + Time
- Communicates how the enterprise/system functions
- To capture and describe Processes and Objects
- Simulate Processes for Dynamic Analysis
- Two Perspectives
  - Process-centered views
  - Object-centered views
- Developed by Knowledge Based Systems Inc. (1992.5)

Element of IDEF3 Language

<table>
<thead>
<tr>
<th>Unit of Behavior</th>
<th>UOB</th>
<th>IDEF Ref #</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UOB #</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Link</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Precedence Link</td>
<td></td>
</tr>
<tr>
<td>Relational Link</td>
<td></td>
</tr>
<tr>
<td>Object Flow Link</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Junction Type</th>
<th>Asynchronous</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&amp; – AND</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Synchronous</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Referent           |       |

- Precedence Link
- Relational Link
- Object Flow Link
- Asynchronous
- Synchronous
Unit of Behavior (UOB)

Function  | Action  | Process
Activity   | Act     | Operation
Event      | Scenario| Decision
Procedure

UOBs always have a unique IDEF3 reference number and may have an associated IDEF0 reference number.

IDEF3 Links

- **Purpose**
  - Describe Temporal, Logical, Conventional or Natural Constraints between UOBs

- **Types of Links**
  - **Precedence**: Process 1 must end before Process 2 starts
    
    ![Precedence Diagram]

  - **Relational**: Process 1 should not start before Process 1 starts
    
    ![Relational Diagram]

  - **Object Flow**: object flows from Process 1 to Process 2
    
    ![Object Flow Diagram]
Junctions

Provide a mechanism to specify the logic of process branching
– Junctions simplify the capture of timing and sequencing relationships between multiple process paths

Junction

Convergence/ Divergence

Logic

Timing

XOR(X)

AND(&)

OR (O)

Synchronous

Asynchronous

Fan-Out

Fan-In

Junctions

Fan-Out

Fan-In
**Fan-Out Junctions**

<table>
<thead>
<tr>
<th>Junction Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp; Asynchronous “AND”</td>
<td>All succeeding process paths will eventually start and all UOBs on each path will eventually happen</td>
</tr>
<tr>
<td>&amp; Synchronous “AND”</td>
<td>All succeeding process paths will start together and all UOBs on each path will eventually happen</td>
</tr>
<tr>
<td>O Asynchronous “OR”</td>
<td>One or more the following process paths will eventually start and all of the UOBs on these paths will happen</td>
</tr>
<tr>
<td>O Synchronous “OR”</td>
<td>There will be a synchronized initiation of one or more process paths</td>
</tr>
<tr>
<td>X XOR</td>
<td>Exactly one of the following process paths will be initiated and only the UOBs on that path will happen</td>
</tr>
</tbody>
</table>

**Fan-In Junctions**

<table>
<thead>
<tr>
<th>Junction Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp; Asynchronous “AND”</td>
<td>All preceding processes must complete</td>
</tr>
<tr>
<td>&amp; Synchronous “AND”</td>
<td>All preceding processes will complete simultaneously</td>
</tr>
<tr>
<td>O Asynchronous “OR”</td>
<td>One or more the preceding processes will complete</td>
</tr>
<tr>
<td>O Synchronous “OR”</td>
<td>One or more the preceding processes will complete simultaneously</td>
</tr>
<tr>
<td>X XOR</td>
<td>Exactly one of the preceding processes will complete</td>
</tr>
</tbody>
</table>
**Exemplary Junctions and Decomposition**

**Referents**

- Used to point to an IDEF3 elements
- Draws the reader’s attention to an important point or note
- Provides Feed-back mechanism for recursive process flows
- Provides mechanism for re-use of predefined UOB’s or scenarios

**Call and Continue Referent**

The referenced elements needs only to initiate before the focus IDEF3 elements can progress to completion

**Call and Wait Referent**

The referenced elements needs to both initiate and complete before the focus IDEF3 elements can progress to completion

<table>
<thead>
<tr>
<th>Referent Type</th>
<th>Referenced Element Label</th>
<th>Locator</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOB</td>
<td>UOB Label</td>
<td>UOB #</td>
</tr>
<tr>
<td>SCENARIO</td>
<td>Scenario Label</td>
<td>Scenario #</td>
</tr>
<tr>
<td>TS</td>
<td>Transition Schematic Label</td>
<td>Transition Schematic #</td>
</tr>
<tr>
<td>GO-TO</td>
<td>UOB Label</td>
<td>Scenario Label</td>
</tr>
</tbody>
</table>
Exemplary Referents

Object Schematics

- Describe how to express detailed object-centered process information
- Object State Transition Network (OSTN)

<table>
<thead>
<tr>
<th>Process View</th>
<th>Object State View</th>
</tr>
</thead>
<tbody>
<tr>
<td>Represent actions applied to objects</td>
<td>Represent condition or status of objects applying actions</td>
</tr>
<tr>
<td>Process-centered</td>
<td>Object-centered</td>
</tr>
<tr>
<td>Sequence</td>
<td>State of objects, entry &amp; exit conditions</td>
</tr>
</tbody>
</table>

- If Process View exists, develop Object View to verify each action/activity is necessary in the process
- If having trouble with sequence, develop Object View to identify actions needed to transition between states
OSTN Elements

Referent elements:
- scenario
- UOB
- action/activity
- other OSTN

Exemplary OSTN

Object: Pizza

- UOB/Add Sauce and Cheese
- UOB/Add Topping
- UOB/Bake Pizza
- Well Cooked Pizza
- Burned Pizza
Representing Stochastic Processes

- Transition schematic illustrating possible complex state transition logic
Process Model → Simulation

- Develop Process Model → Allocate captured Objects to each Process
- Select Object Type (entity, resource, location, queue, transport)
- Input Object Information (arrival time, duration time, etc.)
- Decide Junction Logics → Automatically generate WITNESS™ simulation program → Analyze Simulation results