

Data and Process Modeling

Chapter 3

Data Models Start with User Views

Acme Industries INVOICE

Customer Number: 1454837
Customer: W. Coyote
General Delivery
Falling Rocks, AZ 84211
(999) 555-9345

Terms: Net 30
Ship Via: USPS
Order Date: 12/01/2008

Product No.	Description	Quantity	Unit Price	Extended Amount
SFR-2290	Super Strength Springs	2	24.00	\$ 48.00
STR-67	Foot Straps, leather	2	2.50	\$ 5.00
HLM-45	Deluxe Crash Helmet	1	47.88	\$ 47.88
SFR-1	Rocket, solid fuel	1	128,200.40	\$ 128,200.40
ELT-1	Emergency Location Transmitter	1	79.88	** FREE GIFT **
TOTAL ORDER AMOUNT:				\$ 128,321.28

Data Model Diagramming

- Entity Relationship Diagram (ERD) is most common
- Original by Peter Chen in 1976
- Common ERD Elements:
 - Entities in rectangles or boxes
 - Relationships shown with lines
 - Line end symbols show maximum cardinality
 - Symbols near line ends show minimum cardinality
 - Attributes are optional

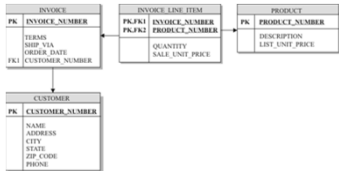
Chen's ERD Format



Chen's ERD Format

- Relationships have a diamond containing a descriptive word or phrase
- Many-to-many relationships have a rectangle around the diamond to represent the intersection entity
- Maximum cardinality shown with "1" or "M" near line end
- Attributes appear in ellipses (when included)

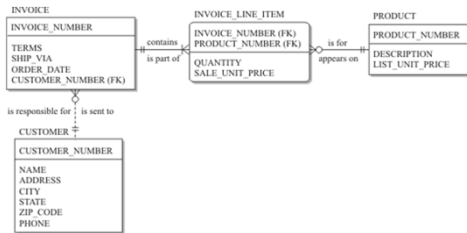
The Relational Format



The Relational Format

- Cardinality shown with single arrowhead for "one" and nothing at line end for "many"
 - Like object notation, arrow points to the entity class from which the subclass inherits
- Attributes listed inside the entity rectangles
- Unique identifier attribute(s) appear above horizontal line inside rectangle and/or **PK** next to attribute
- Foreign key attributes shown with **FK** next to attribute

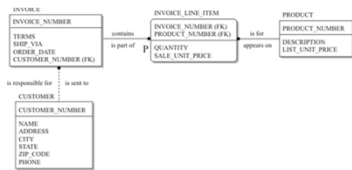
The IE Format



The IE Format

- Solid lines for *identifying* relationships (foreign key is part of the entity's primary key)
- Dotted lines for *non-identifying* relationships
- Maximum cardinality:
 - Short perpendicular line (or nothing) for "one"
 - "Crow's foot" for "many"
- Minimum cardinality
 - Small circle near line end for "zero"
 - Short perpendicular line for "one"

The IDEF1X Format

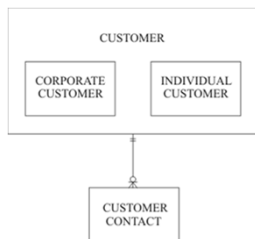


The IDEF1X Format

- Line format
 - Solid line for identifying relationships
 - Broken line for non-identifying relationships
- Solid circle means “zero, one or more”
 - Adding “P” means mandatory (“one or more”)
 - Adding “1” means “must be one and only one”
- No symbol as line end means “one and only one”
 - Adding a small diamond means “zero or one”

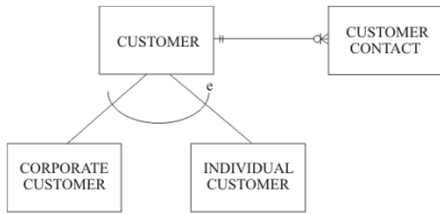
Representing Supertypes and Subtypes

Using Nested Entities



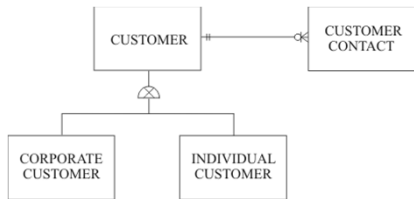
Representing Supertypes and Subtypes

Using Lines and an Arc



Representing Supertypes and Subtypes

Using Lines and a Special Symbol



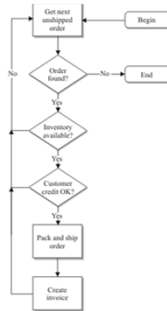
Process Models: Scenario

Find all unshipped orders in the database.

For each order, do the following:

- Check for available inventory. If sufficient inventory for the order is not available, skip to the next order.
- Check the customer's credit to make sure they are not over their credit limit and do not have some other credit problem, such as overdue payments. This would typically occur at the time the order is entered, but it needs to occur again here because a customer's credit status with Acme Industries can change at any time. If a credit problem is found, skip to the next order.
- Generate the documents required to pack and ship the order (packing slip, shipping labels, and so on) and route them to the shipping department.
- When the shipping department has finished with the order, create the invoice for the order and bill the customer accordingly.

The Flowchart



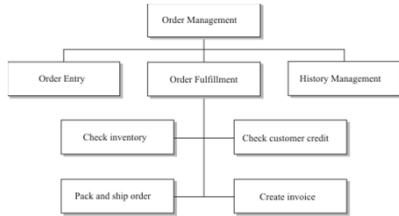
Flowcharts

- What are the strengths and weaknesses of flowcharts?

Flowcharts

- Strengths
 - Preferred by procedural language programmers
 - Applicable to non-programming contexts
 - Useful for spotting reusable components
 - Easy to modify
- Weaknesses
 - Not applicable to nonprocedural or object languages
 - Not always appropriate (e.g. recursive processes)

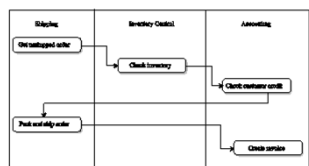
Function Hierarchy Diagram



Function Hierarchy Diagram

- Strengths
 - Quick and easy to learn and use
 - Quickly document bulk of function (80%)
 - Good overview at high and medium level of detail
- Weaknesses
 - Checking quality difficult and subjective
 - Cannot handle complex interactions among functions
 - Do not clearly show sequence or dependencies
 - Not effective for large hierarchies or high detail

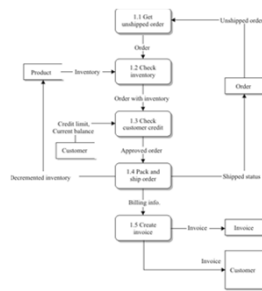
Swim Lane Diagram



Swim Lane Diagram

- Strengths
 - Unmatched ability to show who does what
 - Excellent for identifying inefficiencies and re-engineering opportunities
- Weaknesses
 - Less suitable for complex processes
 - Does not show exception or error handling

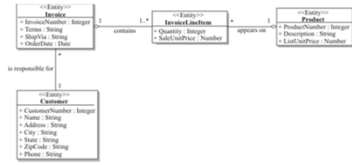
The Data Flow Diagram



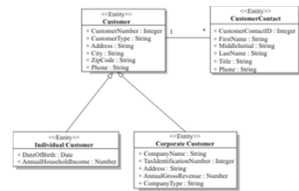
The Data Flow Diagram

- Strengths
 - Easily shows overall structure
 - Good for top-down design
 - Good for presentation of designs
- Weaknesses
 - Time-consuming and labor-intensive
 - Top-down design has proven ineffective
 - Poor at showing complex processing, but easily augmented

UML Class Diagrams



UML Class Diagram with Subclasses



CRUD Matrix

- One axis shows processes
- Other axis shows entities
- Cells show how they interact:
 - C – Create
 - R – Read
 - U – Update
 - D – Delete
- Good for identifying inconsistencies
