Data Flow Diagrams

A structured analysis technique that employs a set of visual representations of the data that moves through the organization, the paths through which the data moves, and the processes that produce, use, and transform data.
Why Data Flow Diagrams?

• Can diagram the organization or the system
• Can diagram the current or proposed situation
• Can facilitate analysis or design
• Provides a good bridge from analysis to design
• Facilitates communication with the user at all stages
Types of DFDs

- **Current** - how data flows now
- **Proposed** - how we’d like it to flow
- **Logical** - the “essence” of a process
- **Physical** - the implementation of a process
- **Partitioned physical** - system architecture or high-level design
Levels of Detail

• **Context level diagram** - shows just the inputs and outputs of the system
• **Level 0 diagram** - decomposes the process into the major subprocesses and identifies what data flows between them
• **Child diagrams** - increasing levels of detail
• **Primitive diagrams** - lowest level of decomposition
Recommended Progression

• **Current logical diagrams**
  – start with context level
  – decompose as needed for understanding

• **Proposed logical diagrams**
  – start at level where change takes place
  – decompose as far as possible

• **Current physical diagrams**
  – at level of change

• **Proposed physical diagrams**
  – same levels as proposed logical
  – lower levels become design
Four Basic Symbols

- Source/Sink
- Data Flow
- Process
- Data Store
Context Level Diagram

• Just one process
• All sources and sinks that provide data to or receive data from the process
• Major data flows between the process and all sources/sinks
• No data stores
Running Example

Course Registration: Context level Diagram
Level 0 Diagram

- Process is “exploded”
- Sources, sinks, and data flows repeated from context diagram
- Process broken down into subprocesses, numbered sequentially
- Lower-level data flows and data stores added
Running Example

Course Registration: Current Logical Level 0 Diagram

1.0 Register Student for Course
   - Student Class Records
   - Student and Course Data

2.0 Collect Student Fee Payment
   - Payment Information
   - D2 Student Payments

3.0 Produce Student Schedule
   - Student Schedule

4.0 Produce Class Roster
   - Class Roster

5.0 Produce Enrollment Report
   - Enrollment Report

Registrar

Professor

Student

Student

Student Class Record

Student Class Record

Student Class Record
Child Diagrams

• “Explode” one process in level 0 diagram
• Break down into lower-level processes, using numbering scheme
• Must include all data flow into and out of “parent” process in level 0 diagram
• Don’t include sources and sinks
• May add lower-level data flows and data stores
Running Example

Course Registration: Current Logical Child Diagram

1.1 Check Prerequisites Met

1.2 Check for Availability

1.3 Enroll Student in Class

D3 Semester Schedule

Available Seats

Available Seats

Valid Class Request

Feasible Class Request

Error

Student and Course Data

D1 Student Class Records

D4 Student Transcripts

D5 Course Catalogue

Course Record

Student Record

Class Request

Error
Physical DFDs

- Model the implementation of the system
- Start with a set of child diagrams or with level 0 diagram
- Add implementation details
  - indicate manual vs. automated processes
  - describe form of data stores and data flows
  - extra processes for maintaining data
Running Example

Course Registration: Current Physical Child Diagram

1.1 Check Prerequisites Met (manual)

1.2 Check for Availability (myUMBC)

1.3 Enroll Student in Class (STARS)

D1 Semester Enrollment DB

D3 Semester Schedule DB

Available Seats

Available Seats

D4 Department Student File

D5 Course Catalogue (text)

Class Request

Advisement Authorization

Feasible Class Request

Unavailability Message

Student Notified (verbally)

Course Description

Student and Course Data

Student File
Running Example

Course Registration: Proposed Physical Child Diagram

1.1 Check Prerequisites Met (automated)

1.2 Check for Availability (automated)

1.3 Enroll Student in Class (automated)

D1 Semester Enrollment DB
D3 Semester Schedule DB
D5 Course Catalogue DB
D4 Registrar’s Student DB

Class Request

Student Requested

Authorized Class Request

Valid Class Request

Available Seats

Available Seats

Available Seats

Student Notified (email)

Student Emailed

Student Emailed

Student Emailed

Course Record

Course Record

Course Record

Student and Course Data

Student and Course Data

Student and Course Data
Partitioning a physical DFD

• Part of system design
• System architecture
  – high-level design
  – overall shape of system
  – some standard architectures
• Decide what processes should be grouped together in the system components
Running Example

**Course Registration: Physical diagram (partitioned)**

1.1 **Check Prerequisites Met (automated)**

1.2 **Check for Availability (automated)**

1.3 **Enroll Student in Class (automated)**

D1 Semester Enrollment DB

D3 Semester Schedule DB

D4 Registrar’s Student DB

D5 Course Catalogue DB

Class Request

Authorized Class Request

Available Seats

Valid Class Request

Student Emailed

Student Notified (email)

Student and Course Data

Student Emailed

Course Record

Available Seats

Available Seats
Another Example

Perfect Pizza: Context Level Diagram

Customer
- Phone Number
- Customer Order
- Customer Info

Cook
- Weekly Report
- Cook Order

Delivery Person
- Delivery Information

Management
Another Example

Perfect Pizza: Current Logical Level 0 Diagram

1.0 Find Customer Record

2.0 Take Customer Order

3.0 Print Delivery Order

Delivery Person

4.0 Sales Records

5.0 Add Customer Record

6.0 Send Order to Cook

7.0 Print Weekly Totals
Another Example

Perfect Pizza: Current Logical Child Diagram

1. Order Information
   - Customer History
     D2 Customer History
   - Order Information
   - Discount Amount
   - Discount Information
     D3 Sales Records

2. Determine Customer Discount
   - Customer Information
   - Discount Amount
   - Record Discount
     3.2 Record Discount

3. Print Delivery Instructions
   - Delivery Information
   - Print Delivery Instructions
Another Example

Perfect Pizza: Current Logical Child Diagram

5.1 Record Customer Information

Customer Information

5.2 Store Customer Record

Raw Customer Information

Customer Record

D1 Customer Master
Another Example

Perfect Pizza: Physical Child Diagram

5.1 Clerk Types Customer Information

5.2 System Validates Customer Information

5.3 Clerk Visually Confirms Cust. Info.

5.4 Format Customer Record

D1 Customer DB

Phoned Customer Information

Recorded Customer Information

Syntax Errors

Valid Customer Information

Cancelled Transaction

New Customer Information

Customer Record

Customer DB

Phone Number
Another Example

Perfect Pizza: Current Physical Level 0 Diagram

1.0 Clerk Finds Customer Row

2.0 Clerk Takes Customer Order (by phone)

3.0 System Prints Delivery Order

4.0 Customer History Record

5.0 Clerk Adds Customer Row

6.0 Clerk Sends Order to Cook (paper)

7.0 Mgr Prints Weekly Totals (batch)

8.0 Mgr Updates Customer History (nightly)

Customer

Delivery Person

Phone Number

Customer Information

Copy of Order Slip

Copy of Order Slips & Del. Printouts

Weekly Report

Management

Cook

Customer Spreadsheet

Phoned Customer Order

Customer Record

Customer History DB

Sales Records File

Copy of Order Slip

Copy of Order Slip

Phoned Customer Info

Customer Record

Phoned Customer Order

Copy of Order Slip
Another Example

Perfect Pizza: Proposed Physical Level 0 Diagram

1.0 System Finds Customer Record

2.0 Clerk Enters Customer Order (by phone)

3.0 System Prints Delivery Order

5.0 Clerk Adds Customer Record

7.0 System Prints Weekly Totals (batch)

D1 Customer DB

D2 Customer History DB

D3 Sales DB

Customer

Phoned Customer Order

Customer Information

Order Info

Discount Info

Order Info

Customer History Record

Sales Info

Weekly Report

Phone #

Customer History DB

Sales DB

Delivery Printout

Delivery Person

Clerk

Management

Cook

Customer Info

Phone Number

Phoned Customer Order

Customer Info

D1 Customer DB

D2 Customer History DB

D3 Sales DB

Weekly Report

Sales Records
Another Example

Perfect Pizza: Partitioned Physical Level 0 Diagram

1.0 System Finds Customer Record

2.0 Clerk Enters Customer Order (by phone)

3.0 System Prints Delivery Order

5.0 Clerk Adds Customer Record

Customer

Phoned Customer Order

Order Info

Customer Info

Clerk Enters Customer Order

Delivery Order

Number

D1 Customer DB

Phone #

D2 Customer History DB

D3 Sales DB

Cook

Management

Phone

System Prints Weekly Totals (batch)

Frequency

Weekly

Totals

(batch)

Discount Info

Sales Records

Weekly Report

Delivery Printout

Phone #

Customer Order

Discount Info

Order Info

D3 Sales DB

Sales

Records

Weekly Report

Delivery Printout

Delivery Person

System Finds Customer Record

Phoned Customer Info

Customer Record

Phoned Customer Info

Customer Order

Order Info

D2 Customer History DB

D3 Sales DB

Discount Info

Order Info

Customer History Record

Customer Information

Cust. Info.

D3 Sales DB

Weekly Totals (batch)

System Prints Weekly Totals (batch)
Data Flow Diagramming Rules

• Processes
  – a process must have at least one input
  – a process must have at least one output
  – a process name (except for the context level process) should be a verb phrase
    • usually three words: verb, modifier, noun
    • on a physical DFD, could be a complete sentence
1.0 Gather Data

2.0 Compile Statistics

3.0 Analyze Responses

Demographic Data

Survey Responses

Final Report
Data Flow Diagramming Rules

• Data stores and sources/sinks
  – no data flows between two data stores; must be a process in between
  – no data flows between a data store and a source or sink; must be a process in between
  – no data flows between two sources/sinks
    • such a data flow is not of interest, or
    • there is a process that moves that data
2.1 Store Customer Data

2.2 Extract Customer Preferences

D1 Customer Data

Customer Preferences

D2 Customer Preferences

Customer Information

D1 Customer Data

Customer Data

D2 Customer Preferences
Data Flow Diagramming Rules

• Data flows
  – data flows are unidirectional
  – a data flow may fork, delivering exactly the same data to two different destinations
  – two data flows may join to form one only if the original two are exactly the same
  – no recursive data flows
  – data flows (and data stores and sources/sinks) are labelled with noun phrases
2.0
Total Daily Sales

3.0
Print Delivery Instructions

1.0
Take Customer Order

Customer Order

Order Total

Order Information
1.0 Take Customer Order

Customer Order

Customer Address

Customer Information

2.0 Lookup Customer Record

3.0 Print Delivery Instructions

1.0 Take Customer Order

Customer Order

Customer Address

2.0 Lookup Customer Record

3.0 Print Delivery Instructions
Calculate Weekly Sales

Cumulative To-Date Sales

Daily Sales
Data Flow Diagramming Guidelines

• The inputs to a process are different from the outputs

• Every object in a DFD has a unique name
1.0 Get Customer Data

Customer Data

Only if these are exactly the same

2.0 Take Customer Order

3.0 Validate Customer Data
Data Flow Diagramming Guidelines

• A data flow at one level may be decomposed at a lower level
• All data coming into and out of a process must be accounted for
• On low-level DFDs, new data flows can be added to represent exceptional situations
1.0 Get Customer Address

1.1 Get Customer Phone

1.2 Lookup Customer Address

1.3 Request Customer Address

Customer Information ➔ 1.0 Get Customer Address ➔ Customer Address

Customer Phone ➔ 1.1 Get Customer Phone ➔ Customer Phone ➔ 1.2 Lookup Customer Address ➔ Customer Address

Customer Address ➔ 1.3 Request Customer Address ➔ Customer Address
1.0
Get Customer Address

1.1
Get Customer Phone

1.2
Lookup Customer Address

1.3
Request Customer Address

Customer Information → Get Customer Phone → Customer Phone

Customer Address → Request Customer Address → Customer Address

Invalid Phone Number Message

Customer Phone → Get Customer Phone → Invalid Phone Number Message

Customer Address
Data Elements

• Indivisible pieces of data
• Data flows and data stores are made up of data elements
• Like attributes on an ER diagram
• The data elements of a data flow flowing in or out of a data store must be a subset of the data elements in that data store
1.0 Calculate Gross Pay

2.0 Calculate Withholding Amount

3.0 Calculate Net Pay

4.0 Print Employee Paycheck

5.0 Create Time Record

6.0 Reconcile Pay Check

D1 Employee Master

D2 Employee Time Record

D3 Check Reconciliation Record

D4 Withholding Tables

Employee

Hours Worked

Employee Time File

Gross Pay

Number of Dependents

Withholding Rates

Employee Time Record

Gross Pay

Withholding Amount

Net Pay

Paycheck Information

Employee Paycheck

Employee Paycheck

Employee Reconciliation Record

Employee Time Record

Employee Record

Gross Pay

Withholding Amount

Paycheck

Employee

Check Reconciliation

Reconcile Pay Check
DFDs and ERDs

- DFDs and ERDs are both used to model systems, but they show two very different perspectives on the system.
- A DFD shows what the system *does* as well as the *data* that the system manipulates.
- An ERD shows *only* the *data* that the system manipulates.
DFDs and ERDs (cont.)

• **Entities** on an ERD often (but not always) correspond to **data stores** on a DFD.

• **Attributes** on an ERD usually correspond to **data elements** (listed in the data dictionary) that make up the data store and data flows on a DFD.

• **Relationships** on an ERD **do not** correspond to **processes** on a DFD.

• **Sources and sinks** on a DFD usually **do not** show up as **entities** on an ERD.
Example DFD and ERD

DFD

1.0
Take Order

2.0
Convert Order to Cooking Instructions

Cooking Instructions

3.0
Convert Order to Ingredient List

Processed Order

Ingredients

Inventory Processing

D1
Order Log

Cook

Incorrect ERD

Cook

Places Order

Customer

Inventory

Inventory Processing

Cooking Instructions

Ingredients

Item

Quantity

Name

Hours

Name

Address
Example DFD and ERD

**DFD**

1.0 Take Order

2.0 Convert Order to Cooking Instructions

3.0 Convert Order to Ingredient List

**Correct ERD**

Order

- OrderId
- Time
- Date

Contains

- ItemQuantity

Item

- ItemId
- ItemName

Includes

- Ingredient
- Ingredient Quantity

Requires

- Cooking Instructions
- Index
- StepId
- Description