

Data Flow Diagrams

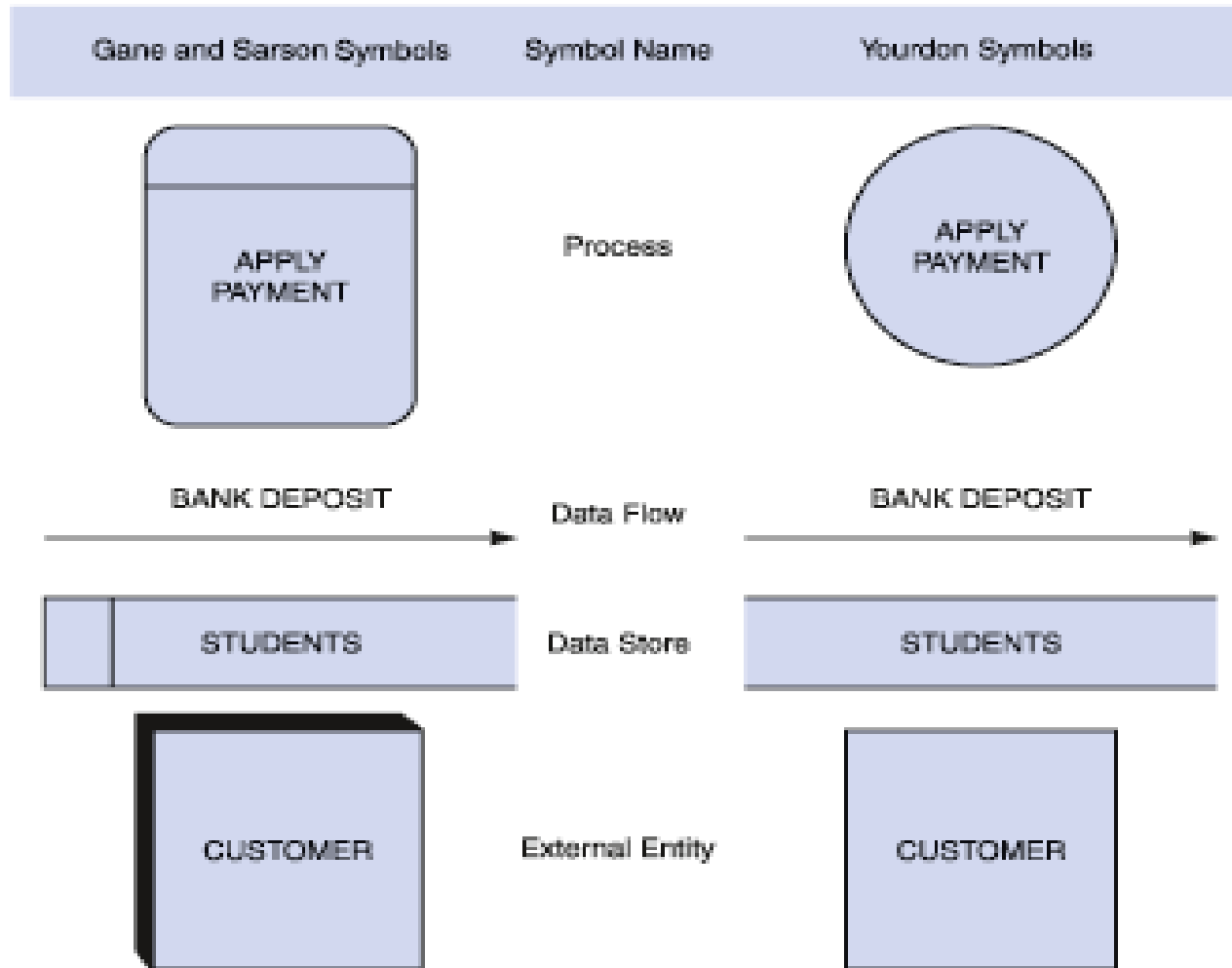
What is a Data Flow Diagram?

- ▶ A data flow diagram (DFD) is a graphical representation of the movement of data between external entities, processes and data stores within a system.
- ▶ Simply put, DFD's show how data moves through an information system.

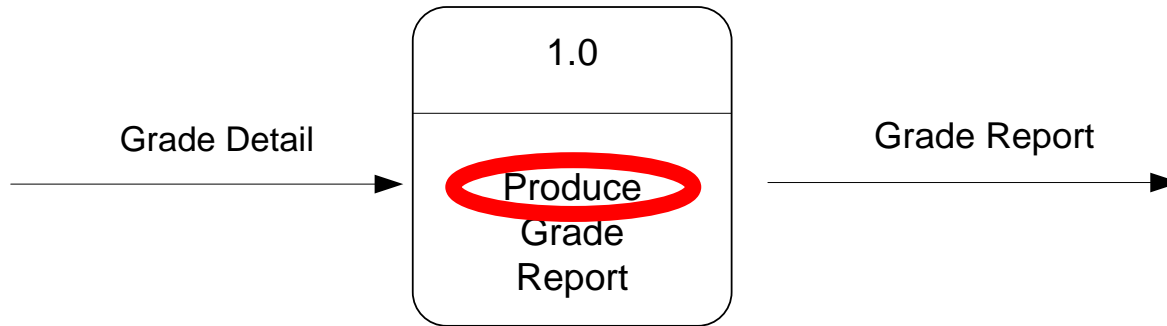
Advantages of a DFD

- ▶ It aids in describing the boundaries of the system.
- ▶ It is beneficial for communicating existing system knowledge to the users.
- ▶ A straightforward graphical technique which is easy to recognise.
- ▶ DFDs can provide a detailed representation of system components.
- ▶ It is used as the part of system documentation file.
- ▶ DFDs are easier to understand by technical and nontechnical audiences
- ▶ It supports the logic behind the data flow within the system.

DFD Symbols

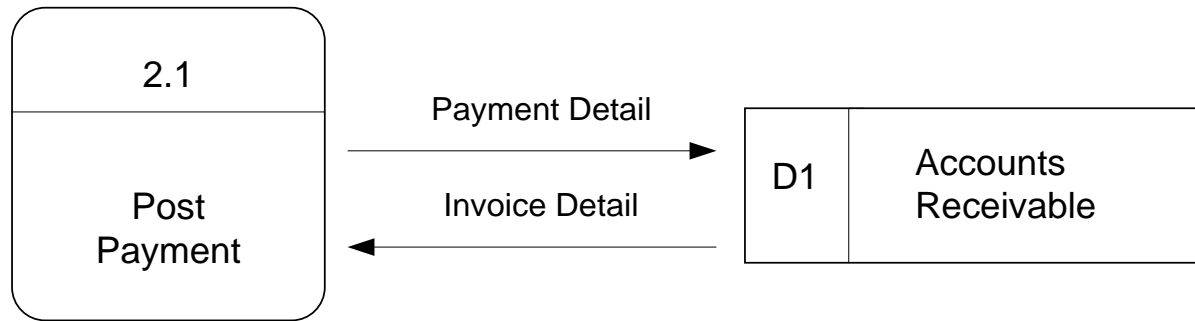


Process



- ▶ The work or actions performed on data so that they are transformed, stored, or distributed.
- ▶ Process labels should be **verb phrases!**

Data Flow



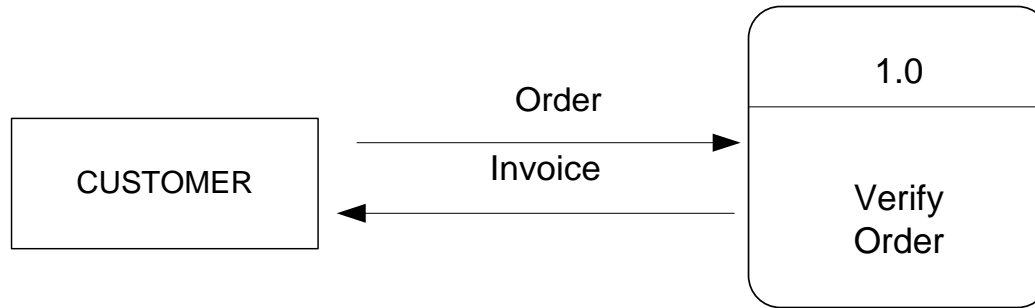
- ▶ A path for data to move from one part of the system to another.
- ▶ Data in motion!
 - ▶ Arrows depict the movement of data.
- ▶ **NO VERBS**

Data Store

D1	Students
----	----------

- ▶ Used in a DFD to represent data that the system stores
- ▶ Data at rest!
- ▶ Labels should be noun phrases
 - ▶ **(NO VERBS)**

External Entity aka Source/Sink



- ▶ The origin or destination of data!
 - ▶ This represents things outside of the system.
- ▶ Source – Entity that supplies data to the system.
- ▶ Sink – Entity that receives data from the system.
- ▶ The labels should be noun phrases!

DFD Naming Guidelines

- ▶ External Entity → Noun
- ▶ Data Flow → Names of data
- ▶ Process → verb phrase
 - ▶ a system name
 - ▶ a subsystem name
- ▶ Data Store → Noun

General DFD Rules

	YES	NO
A process to another process	✓	
A process to an external entity	✓	
A process to a data store	✓	
An external entity to another external entity		✓
An external entity to a data store		✓
A data store to another data store		✓

Advantages of DFDs

- ▶ Simple graphical techniques which are easy to understand
- ▶ Helps define the boundaries of the system
- ▶ Useful for communicating current system knowledge to users
- ▶ Explains the logic behind the data flow within the system
- ▶ Used as the part of system documentation file

Creating Data Flow Diagrams

Steps:

1. Create a list of activities
2. Construct Context Level DFD
(identifies external entities and processes)
3. Construct Level 0 DFD
(identifies manageable sub process)
4. Construct Level 1- n DFD
(identifies actual data flows and data stores)
5. Check against rules of DFD

Creating Data Flow Diagrams

Food Cart Example

Creating Data Flow Diagrams

Example

The operations of a food cart will be used to demonstrate the creation of dataflow diagrams.

Steps:

1. Create a list of activities
 - Old way: no Use-Case Diagram
 - New way: use Use-Case Diagram
2. Construct Context Level DFD (identifies sources and sink)
3. Construct Level 0 DFD (identifies manageable sub processes)
4. Construct Level 1- n DFD (identifies actual data flows and data stores)

Creating Data Flow Diagrams

Example

Think through the activities
that take place at a food cart

1. Create a list of activities

Customer Order
Serve Product
Collect Payment
Produce Product
Store Product

Creating Data Flow Diagrams

Example

Also think of the additional activities needed to support the basic activities.

1. Create a list of activities

Customer Order
Serve Product
Collect Payment
Produce Product
Store Product
Order Raw Materials
Pay for Raw Materials
Pay for Labor

Creating Data Flow Diagrams

Example

Group these activities in some logical fashion, possibly functional areas.

1. Create a list of activities

Customer Order
Serve Product
Collect Payment

Produce Product
Store Product

Order Raw Materials
Pay for Raw Materials

Pay for Labor

Creating Data Flow Diagrams

Example

Create a context level diagram identifying the sources and sinks (users).

Customer Order
Serve Product
Collect Payment

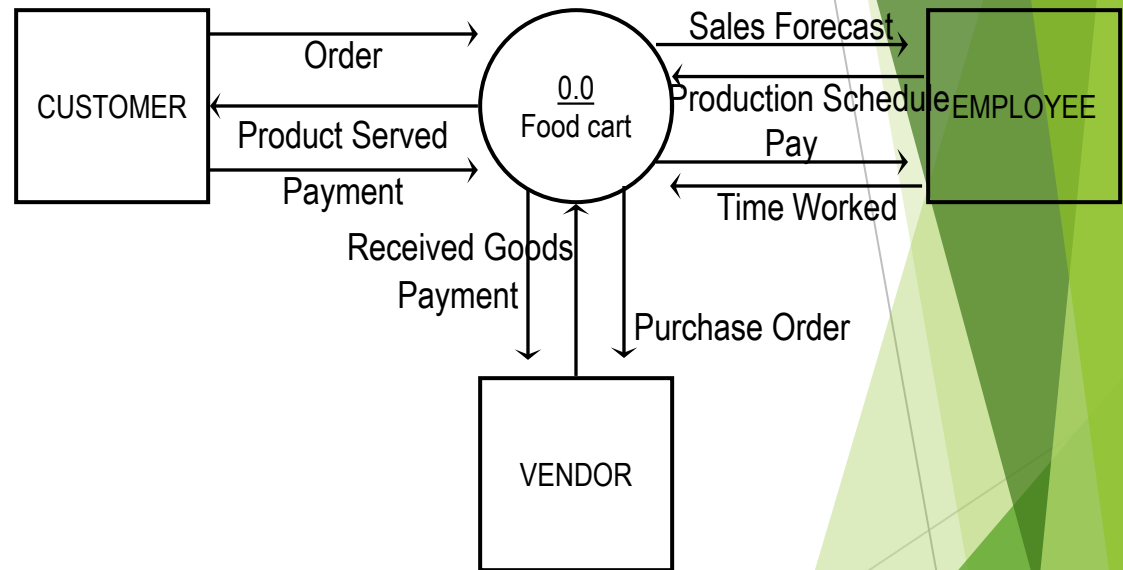
Produce Product
Store Product

Order Raw Materials
Pay for Raw Materials

Pay for Labor

2. Construct Context Level DFD (identifies sources and sink)

Context Level DFD



Creating Data Flow Diagrams

Example

Create a level 0 diagram identifying the logical subsystems that may exist.

Customer Order
Serve Product
Collect Payment

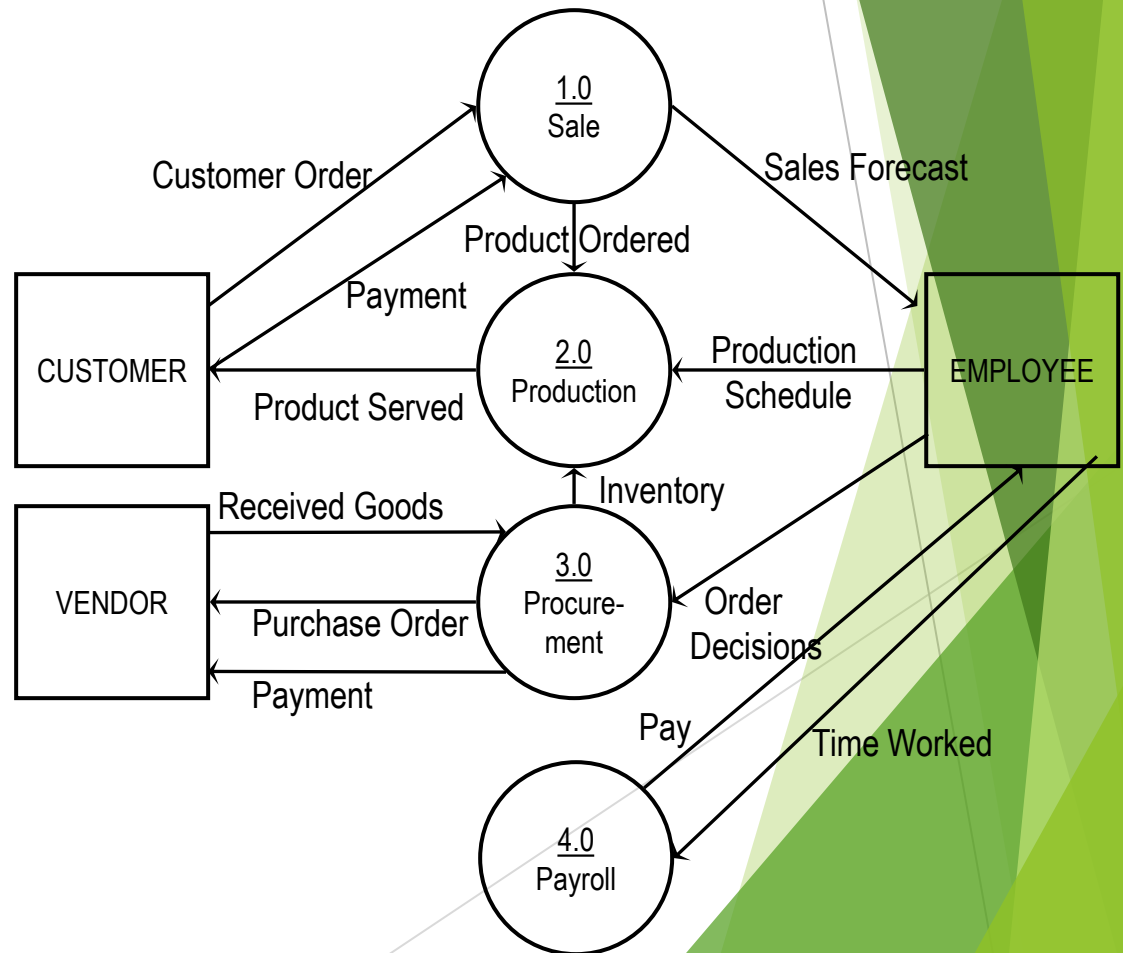
Produce Product
Store Product

Order Raw Materials
Pay for Raw Materials

Pay for Labor

3. Construct Level 0 DFD (identifies manageable sub processes)

Level 0 DFD



Creating Data Flow Diagrams

Example

Create a level 1 decomposing the processes in level 0 and identifying data stores.

Customer Order
Serve Product
Collect Payment

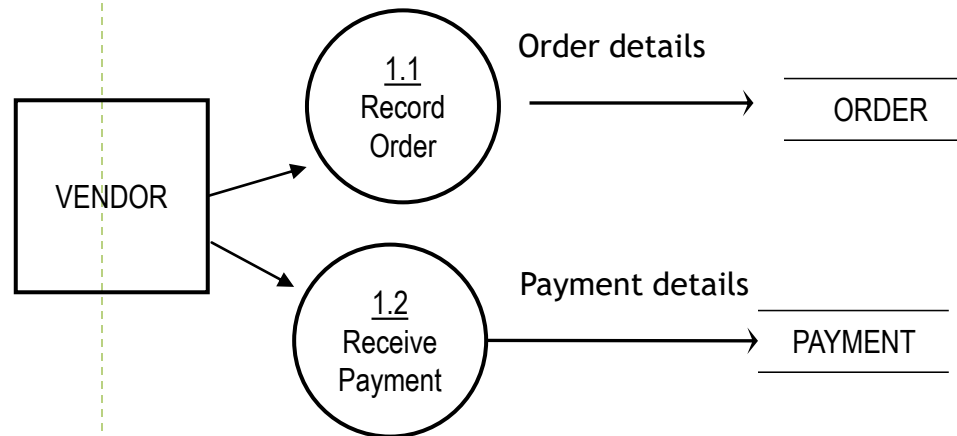
Produce Product
Store Product

Order Raw Materials
Pay for Raw Materials

Pay for Labor

4. Construct Level 1- n DFD (identifies actual data flows and data stores)

Level 1 DFD for Sale



Creating Data Flow Diagrams

Example

Create a level 1 decomposing the processes in level 0 and identifying data stores.

Customer Order
Serve Product
Collect Payment

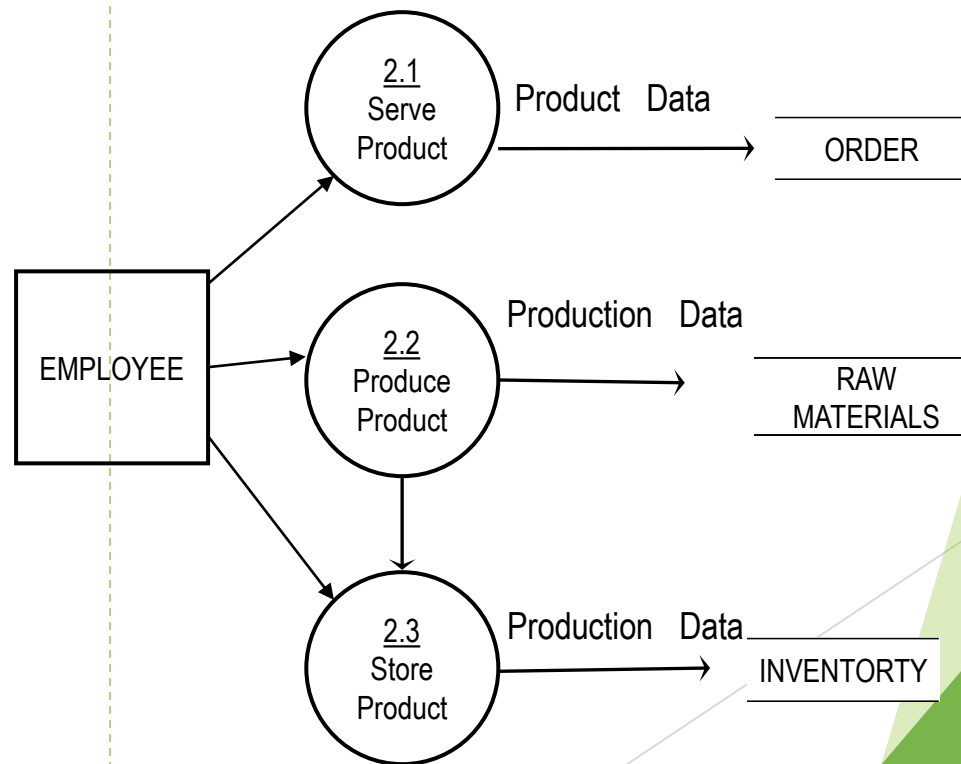
Produce Product
Store Product

Order Raw Materials
Pay for Raw Materials

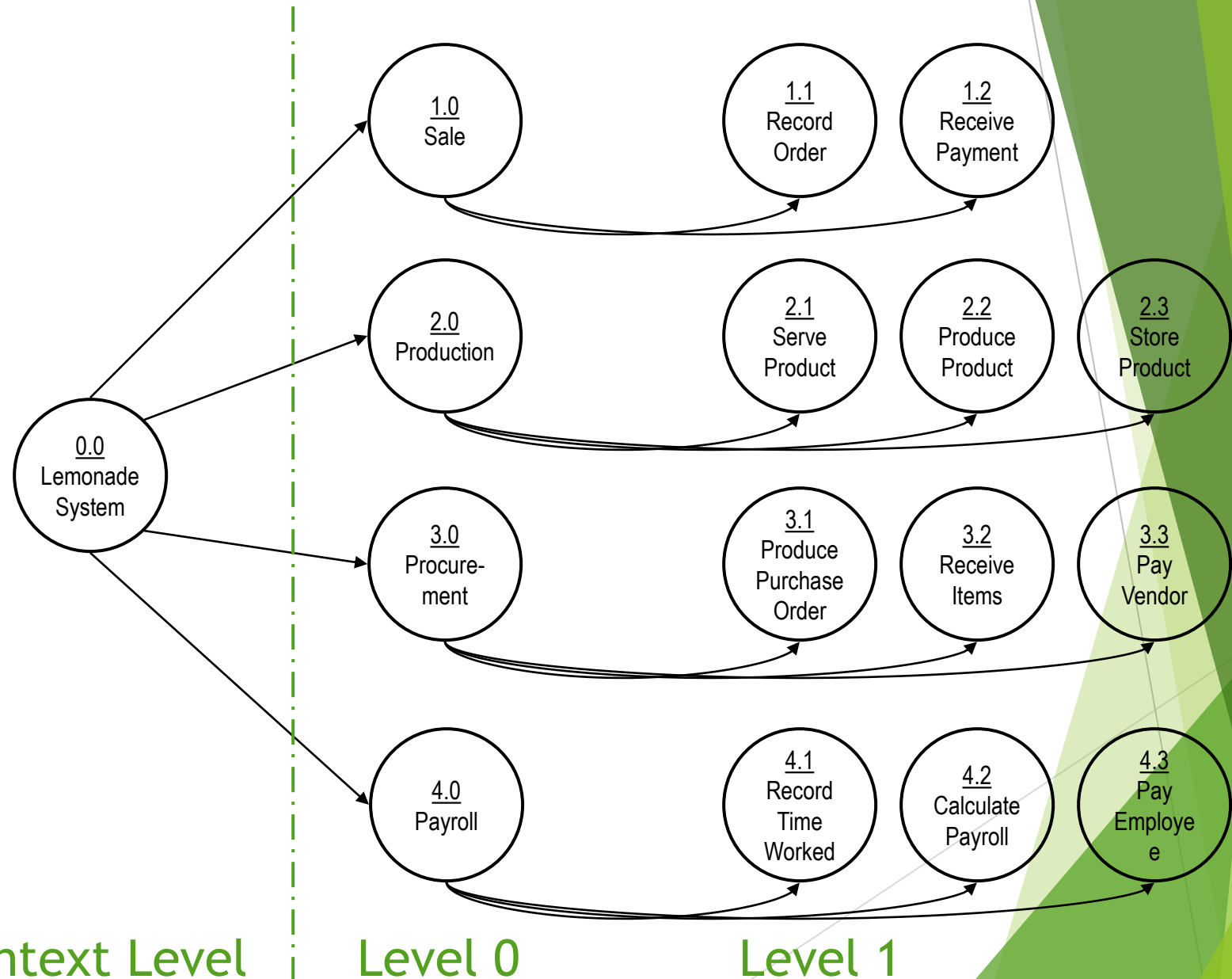
Pay for Labor

4. Construct Level 1 (continued)

Level 1 DFD for Production



Process Decomposition



DFD Example: Bus Garage Repairs

- ▶ Buses come to a garage for repairs.
- ▶ A mechanic and helper perform the repair, record the reason for the repair and record the total cost of all parts used on a Shop Repair Order.
- ▶ Information on labor, parts and repair outcome is used for billing by the Accounting Department, parts monitoring by the inventory management computer system and a performance review by the supervisor.

DFD Example: Bus Garage Repairs (cont'd)

- ▶ *External Entities:* Bus, Mechanic, Helper, Supervisor, Inventory Management System, Accounting Department, etc.
- ▶ *Key process* (“the system”): performing repairs and storing information related to repairs
- ▶ *Processes:*
 - ▶ Record Bus ID and reason for repair
 - ▶ Determine parts needed
 - ▶ Perform repair
 - ▶ Calculate parts extended and total cost
 - ▶ Record labor hours, cost

DFD Example: Bus Garage Repairs (cont'd)

▶ *Data stores:*

- ▶ Personnel file
- ▶ Repairs file
- ▶ Bus master list
- ▶ Parts list

▶ *Data flows:*

- ▶ Repair order
- ▶ Bus record
- ▶ Parts record
- ▶ Employee timecard
- ▶ Invoices