National University
of Computer & Emerging Sciences

SOFTWARE ENGINEERING
(Week-3)

USAMA MUSHARAF

MS-CS (Software Engineering)

LECTURER (Department of Computer Science)

FAST-NUCES PESHAWAR

AGENDA OF WEEK # 3

- Case Study Discussion
- Class Activities
  - Discussion on Assignment # 1
  - Quiz # 1
- Requirement Engineering
- Requirement Analysis (Structured Analysis)

BANKING SYSTEM CASE STUDY

- A bank has several automated teller machines (ATMs), which are geographically distributed and connected via a wide area network to a central server.

- Each ATM machine has a card reader, a cash dispenser, a keyboard/display, and a receipt printer.

- By using the ATM machine, a customer can withdraw cash from either checking or savings account, query the balance of an account, or transfer funds from one account to another.

BANKING SYSTEM CASE STUDY

- A transaction is initiated when a customer inserts an ATM card into the card reader. Encoded on the magnetic strip on the back of the ATM card is the card number, the start date, and the expiration date.

- Assuming the card is recognized, the system validates the ATM card to determine that the expiration date has not passed, that the user-entered PIN (personal identification number) matches the PIN maintained by the system, and that the card is not lost or stolen.

- The customer is allowed three attempts to enter the correct PIN; the card is confiscated if the third attempt fails. Cards that have been reported lost or stolen are also confiscated.

BANKING SYSTEM CASE STUDY

- If the PIN is validated satisfactorily, the customer is prompted for a withdrawal, query, or transfer transaction.

- Before withdrawal transaction can be approved, the system determines that sufficient funds exist in the requested account, that the maximum daily limit will not be exceeded, and that there are sufficient funds available at the local cash dispenser.

- If the transaction is approved, the requested amount of cash is dispensed, a receipt is printed containing information about the transaction, and the card is ejected.

- Customer records, account records, and debit card records are all maintained at the server.

AGILE DEVELOPMENT USING BANKING CASE STUDY

Assignment no 1:

Create a working plan for banking system case study using agile process model (Extreme programming). You are required to

- Identify functional requirements (FR’s) from the case study and write User Stories for each FR in order to have detail understanding.
- Create Iteration Plans.
- Perform Test First Development (Write test descriptions for user story cards).

WHAT IS REQUIREMENT?

What is requirement?

- **The descriptions of what the system should do**
  - **services that it provides and the constraints on its operation**

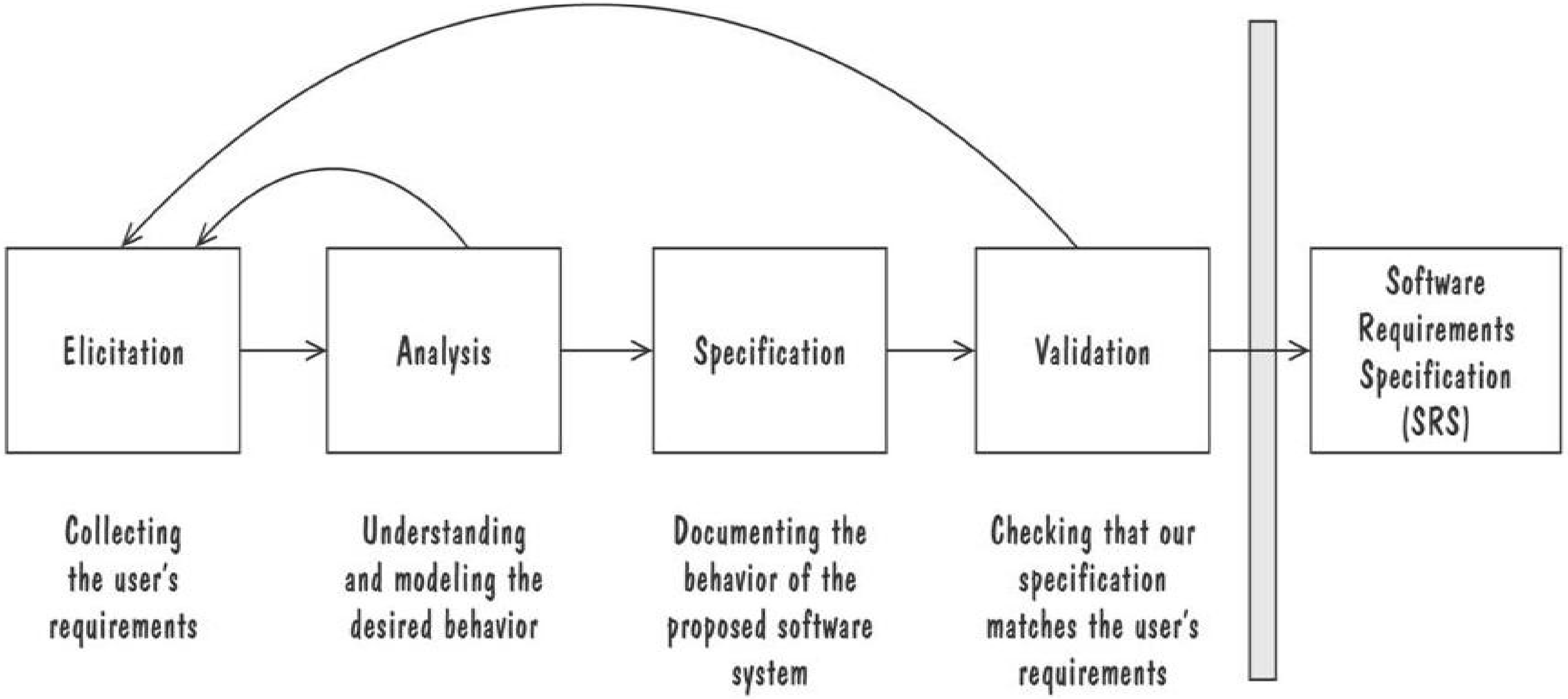
TYPES OF REQUIREMENTS

- **Functional requirements:**
  - statement of **services**
  - how system **reacts** to input
  - how system **behaves** in particular situation
- **Non-functional requirements:**
  - constraints on services **(timing, quality, security etc.)**

- Domain requirements
- Inverse requirements
- Design and implementation constraints

THE REQUIREMENTS PROCESS
(PROCESS FOR CAPTURING REQUIREMENTS)

- Performed by the req. analyst or system analyst
- The final outcome is a Software Requirements Specification (SRS) document


Elicitation
Analysis
Specification
Validation
Software Requirements Specification (SRS)
Collecting the user's requirements
Understanding and modeling the desired behavior
Documenting the behavior of the proposed software system
Checking that our specification matches the user's requirements

REQUIREMENTS ELICITATION
STAKEHOLDERS

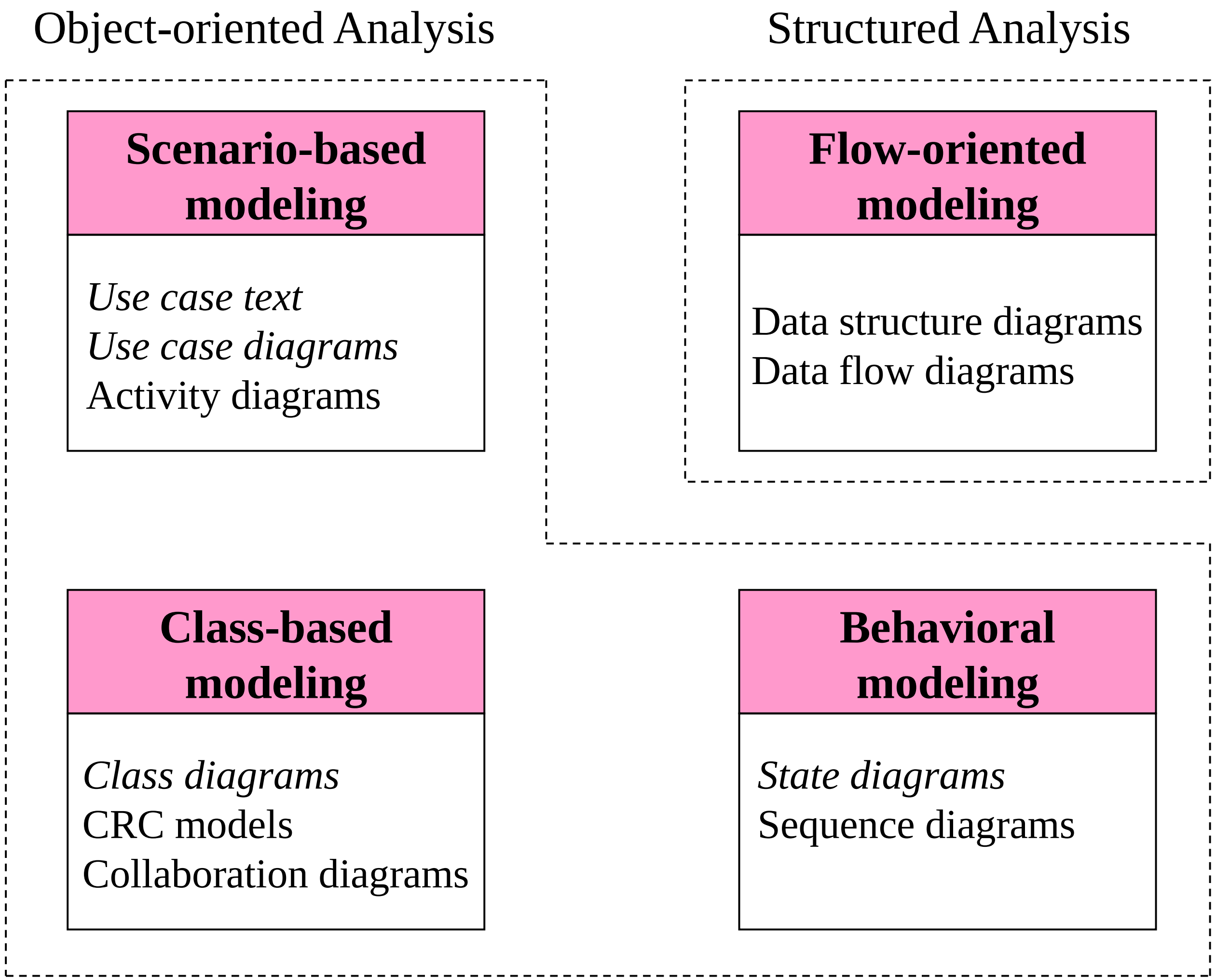
- Clients: pay for the software to be developed
- Users: use the system
- Domain experts: familiar with the problem that the software must automate

REQUIREMENTS ELICITATION
MEANS OF ELICITING REQUIREMENTS

- Interviewing stakeholders
- Reviewing available documentations
- Observing the current system (if one exists)

ANALYSIS MODELING

ELEMENTS OF THE ANALYSIS MODEL



Object-oriented Analysis

Structured Analysis

Scenario-based modeling

Use case text
Use case diagrams
Activity diagrams

Flow-oriented modeling

Data structure diagrams
Data flow diagrams

Class-based modeling

Class diagrams
CRC models
Collaboration diagrams

Behavioral modeling

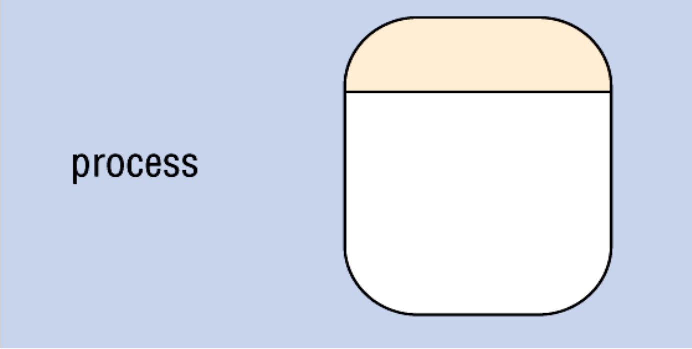
State diagrams
Sequence diagrams

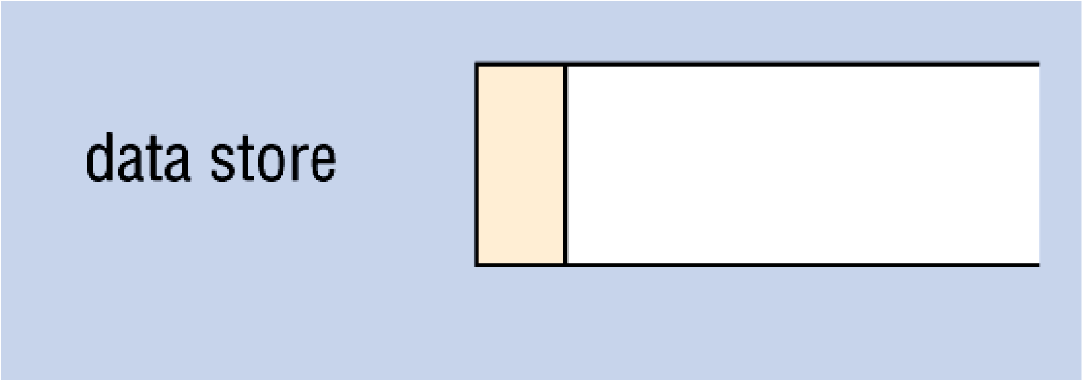
FLOW-ORIENTED MODELING

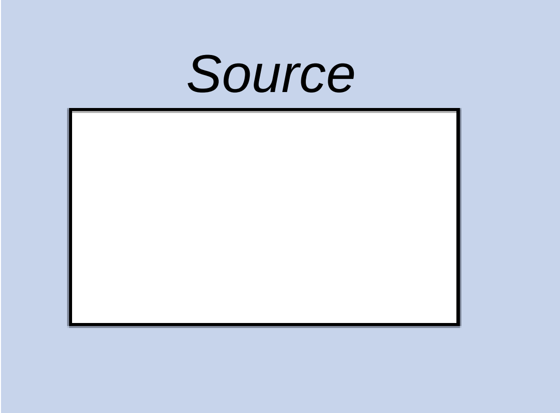
WHAT IS A DATA FLOW DIAGRAM?

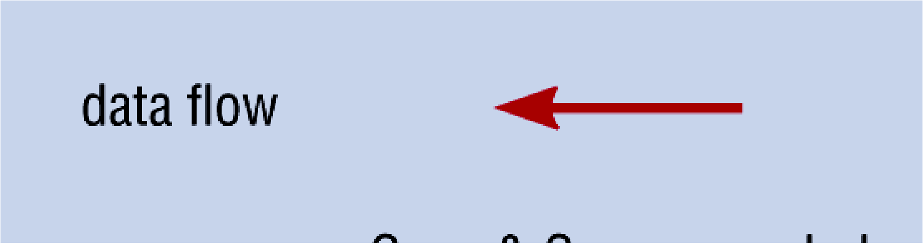
- A data flow diagram (DFD) is a graphical tool that allows system analysts (and system users) to depict the flow of data in an information system.

DATA FLOW DIAGRAM SYMBOLS





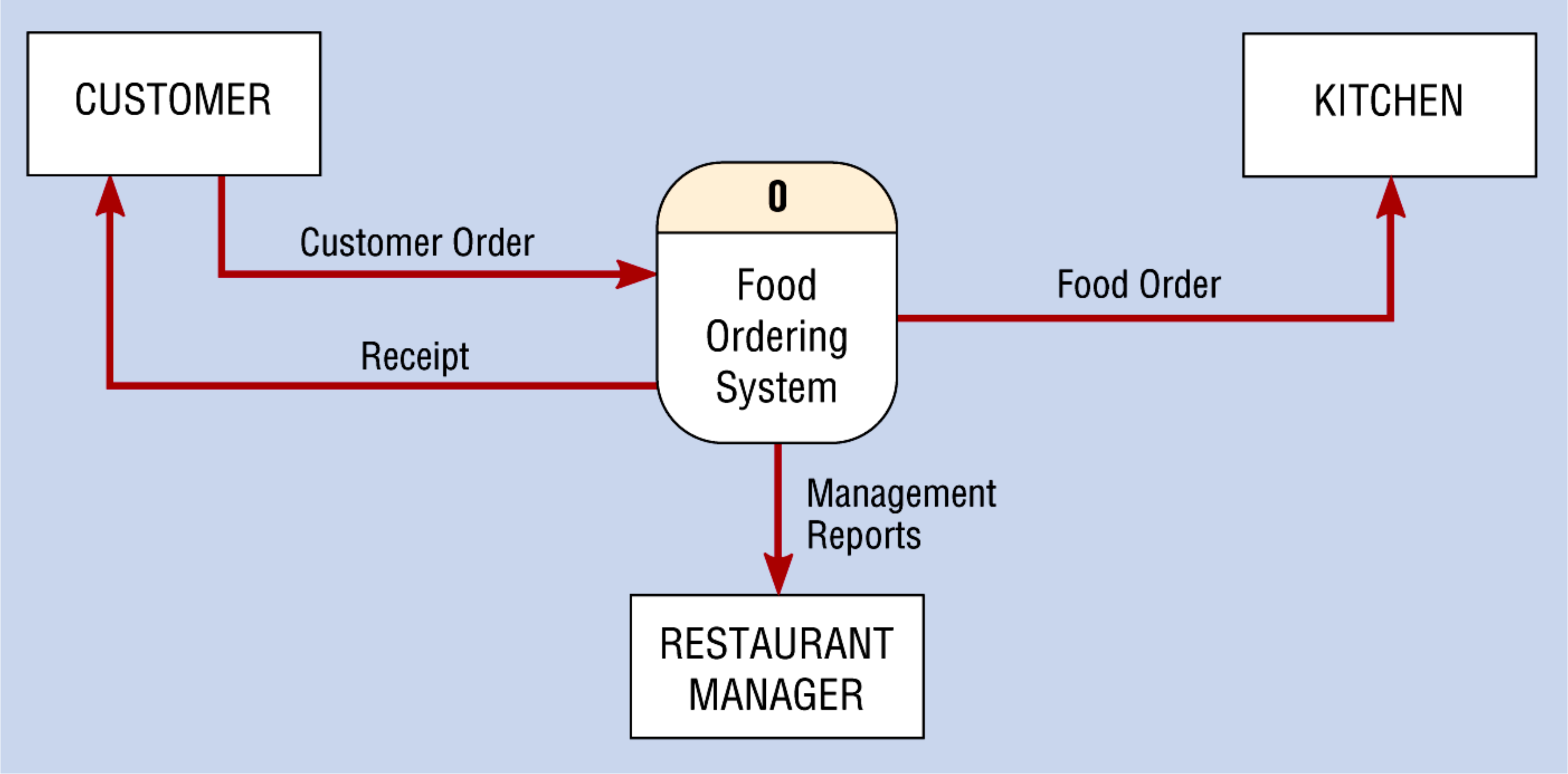




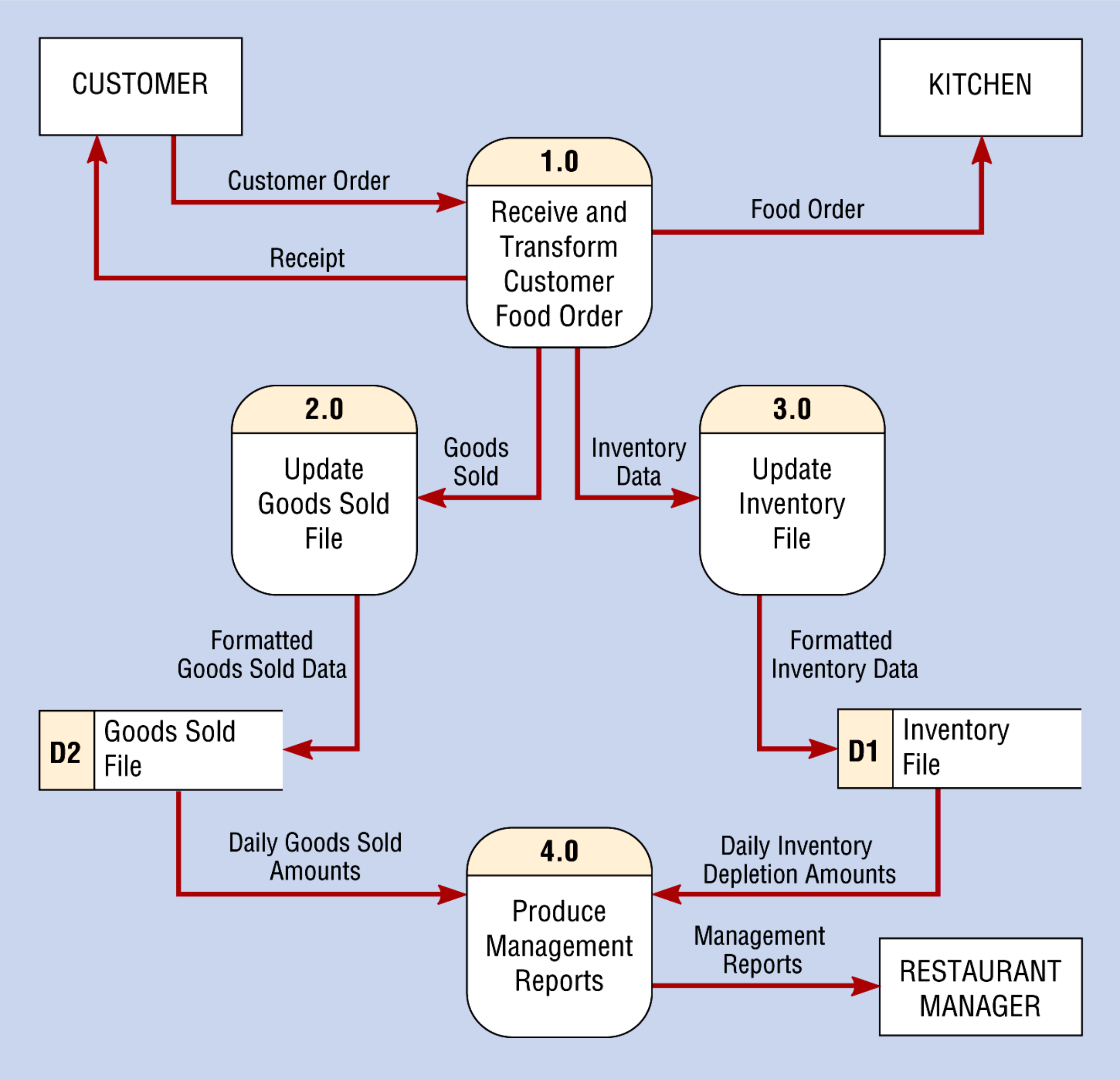
STEPS IN BUILDING DFDS

- Build the context diagram
- Create DFD fragments
- Organize DFD fragments into level 0
- Decompose level 0 DFDs as needed

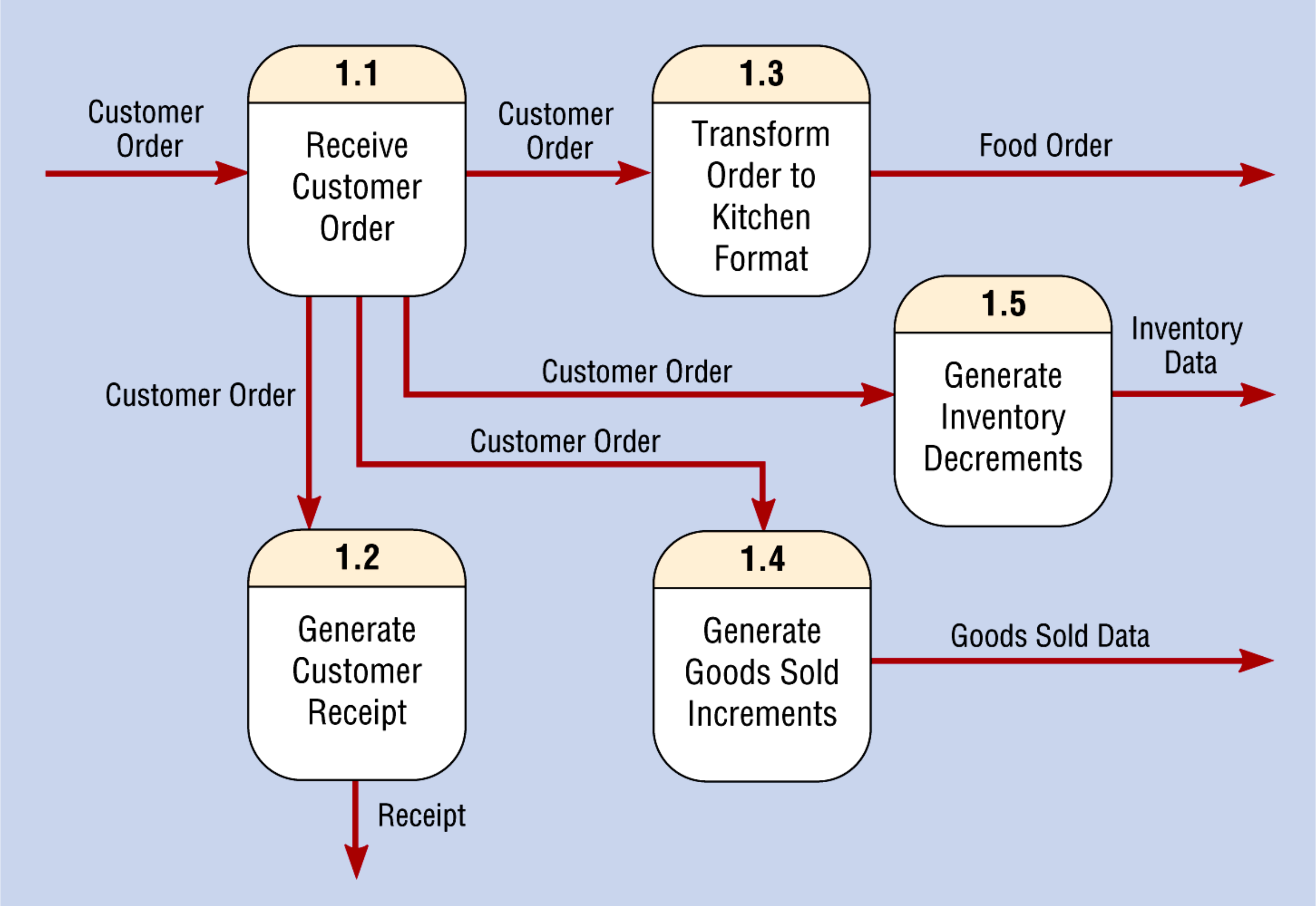
CONTEXT DIAGRAM OF FOOD ORDERING SYSTEM



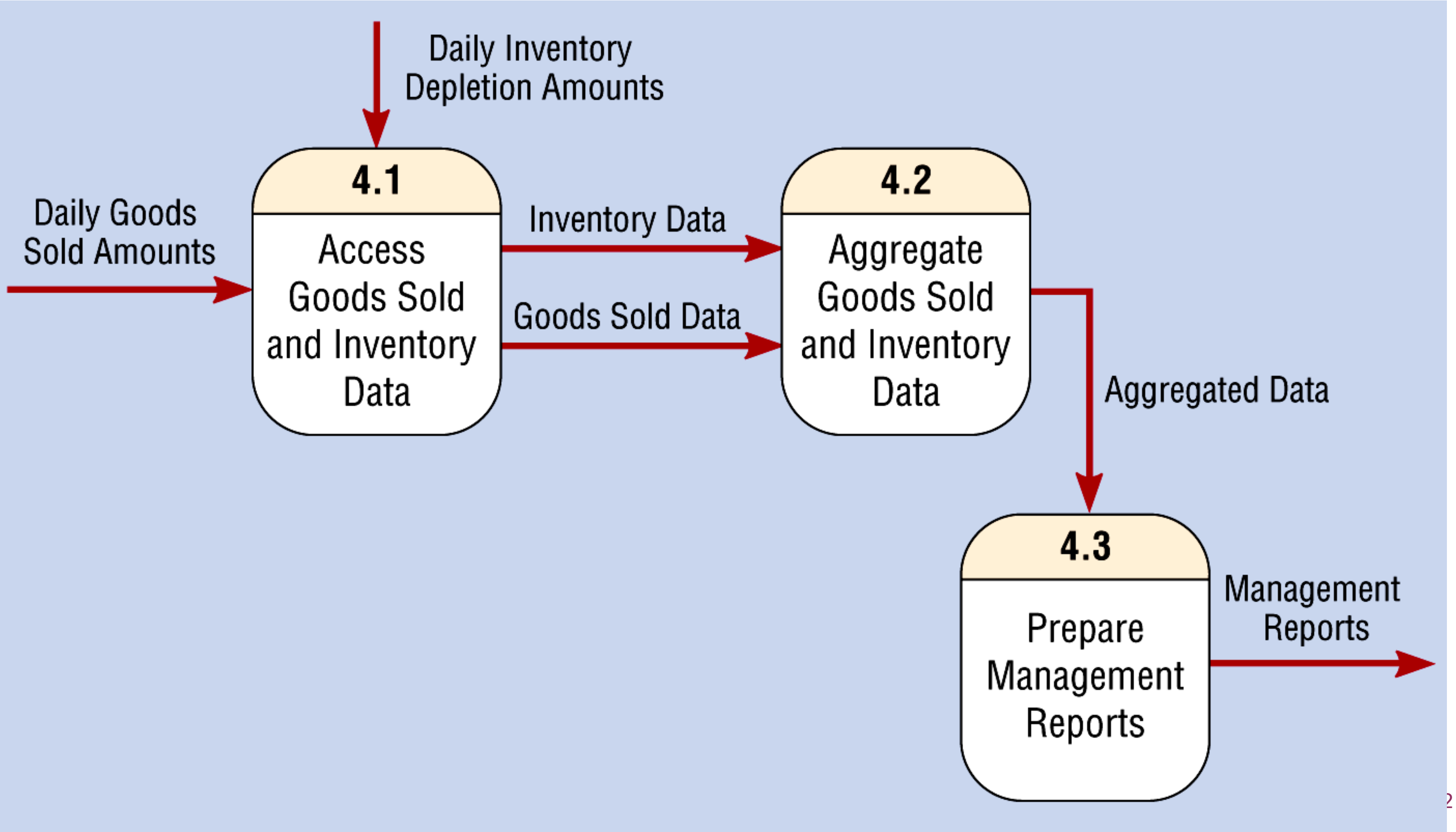
LEVEL-0 DFD OF FOOD ORDERING SYSTEM



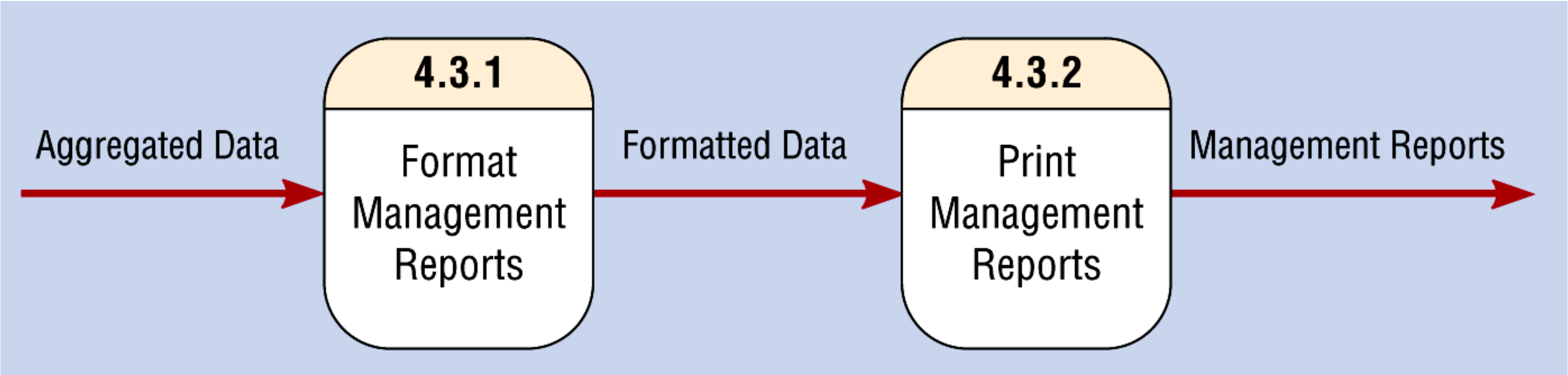
LEVEL-1 DIAGRAM SHOWING DECOMPOSITION OF PROCESS 1.0 FROM THE LEVEL-0 DIAGRAM



LEVEL-1 DIAGRAM SHOWING THE DECOMPOSITION OF PROCESS 4.0 FROM THE LEVEL-0 DIAGRAM

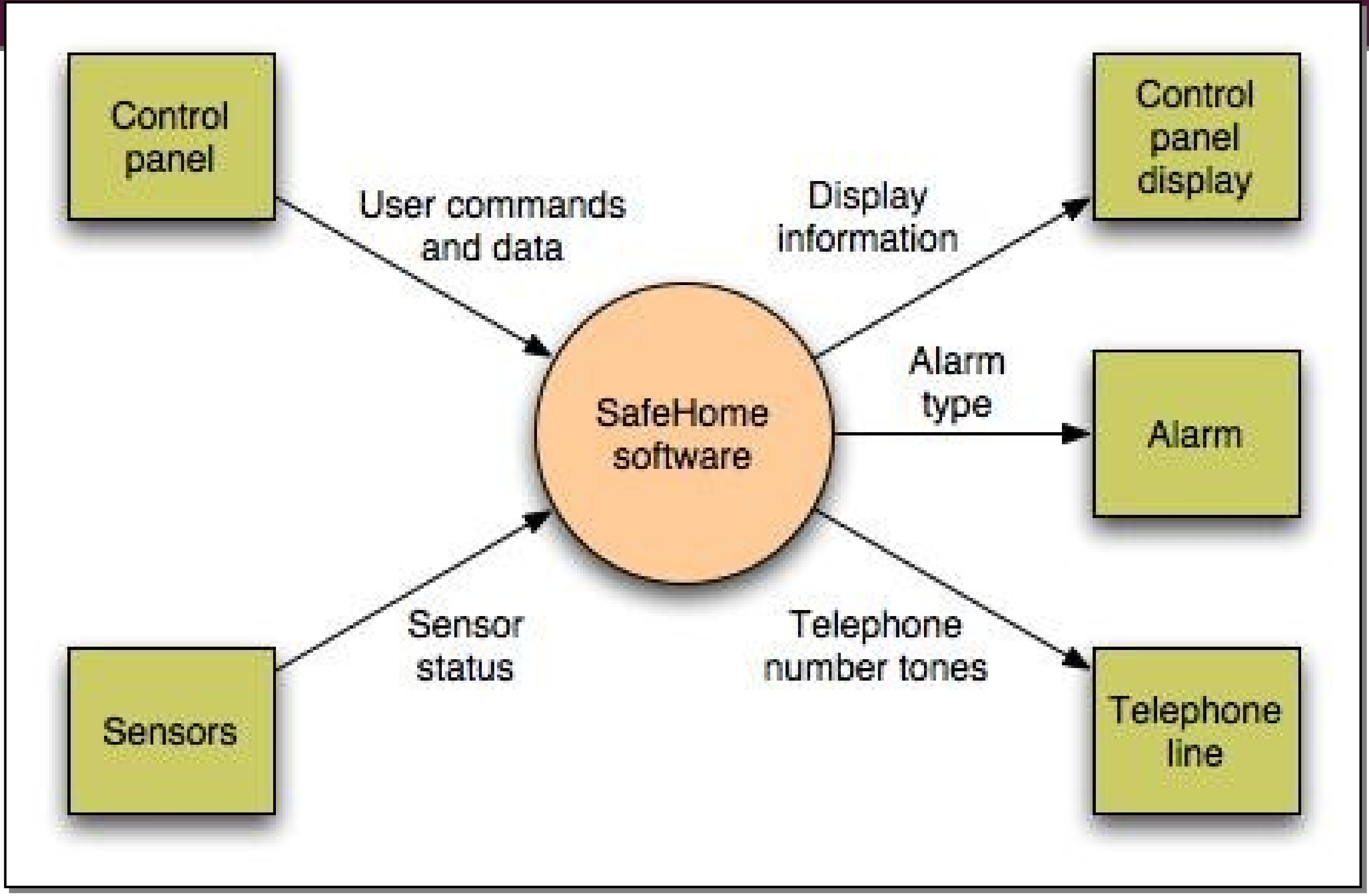

Daily Inventory Depletion Amounts
Daily Goods Sold Amounts
4.1
Access Goods Sold and Inventory Data
Inventory Data
Goods Sold Data
4.2
Aggregate Goods Sold and Inventory Data
Aggregated Data
4.3
Prepare Management Reports
Management Reports

LEVEL-2 DIAGRAM SHOWING THE DECOMPOSITION OF PROCESS 4.3 FROM THE LEVEL-1 DIAGRAM FOR PROCESS 4.0

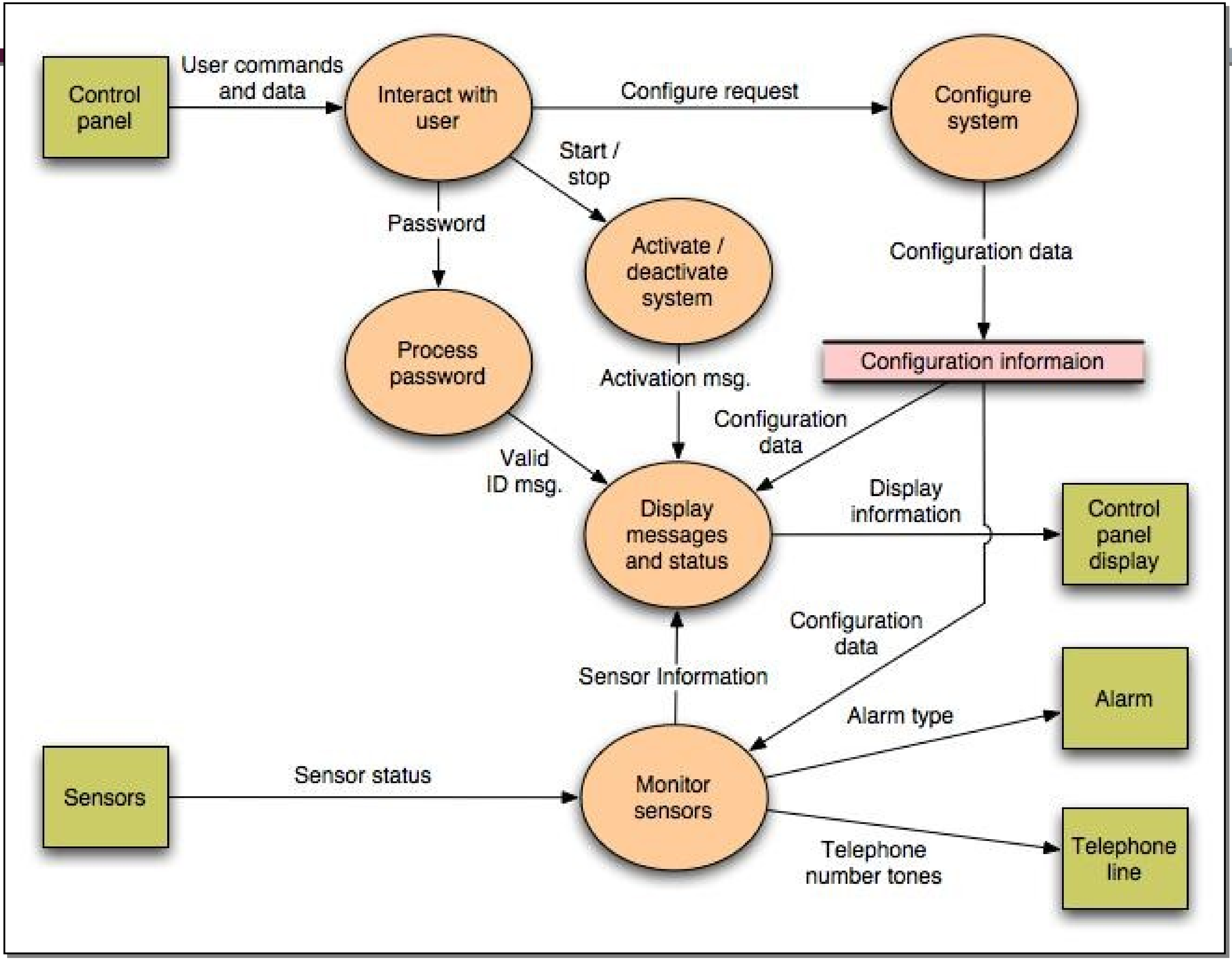


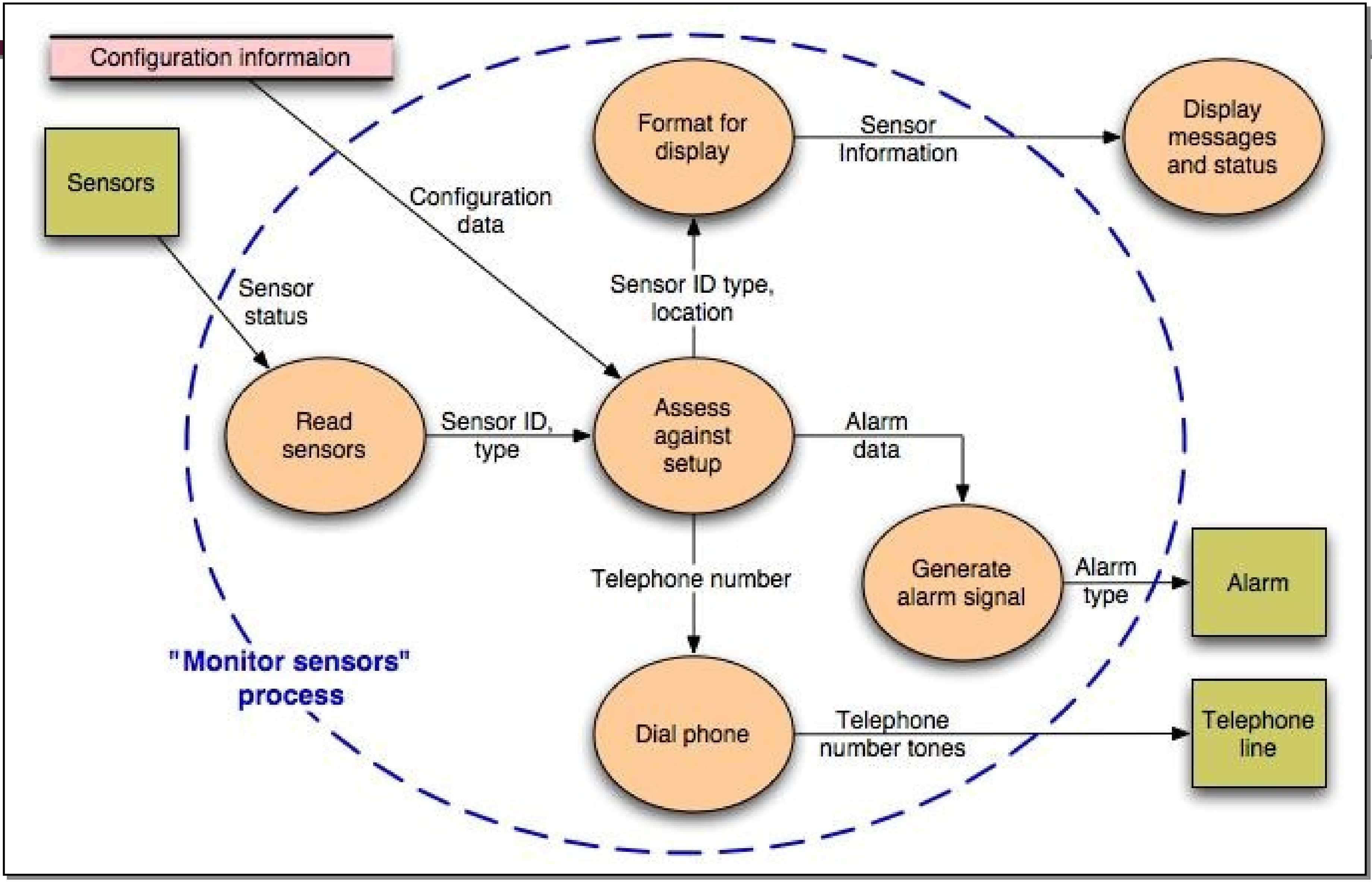
DATA FLOW DIAGRAM OF SAFE HOME SYSTEM

DATA FLOW DIAGRAM



Context-level DFD for *SafeHome* security function


Control panel
User commands and data
Interact with user
Configure request
Configure system
Start / stop
Password
Activate / deactivate system
Configuration data
Process password
Configuration informaion
Activation msg.
Configuration data
Valid ID msg.
Display messages and status
Display information
Control panel display
Configuration data
Sensor Information
Alarm type
Alarm
Sensors
Sensor status
Monitor sensors
Telephone number tones
Telephone line



Level 2 DFD that refines the monitor sensors process

HAVE A GOO DAY!