Information Domain

- Software is **built** to accept the input, manipulate it on some way, and produce output.
- Software also **process** the event.
- An event represents some aspect of system control and is really nothing more than Boolean data – either on or off
- The information domain consists of three different views
- **Information content or data model** ◦
- shows the relationships among the data and control objects that make up the system
- **Information flow** ◦
- represents manner in which data and control objects change as each moves through system
- **Information structure** ◦
- representations of the internal organizations of various data and control items

Analysis Principles

1.The information domain of a problem must be represented and understood.

2.The **function** that the software is to be perform must be defined.

3. The behavior of the software must be represented

4. The model that depict information, function, and behavior must be partitioned in hierarchical fashion

5. The analysis process should move from essential information toward implementation details

Analysis Principles

- By applying these principles, the analysis principles, the analyst approaches a problem systematically .
- Information domain is examined so that the function may be understood more completely
- Models are used so that the characteristics of function and behavior can be communicated in a compact fashion.
- **Partitioning** is applied to reduce complexity

Analysis Principles

- In addition to these operational analysis, Davis suggests a set of guiding principles for requirement engineering
  - ***Understand the problem before you begin to create the analysis model.***
  - ***Develop the prototypes that enable a user to understand how human/machine interaction will occur***

Analysis Principles

- **Record the origin of and the reason for every requirement**
- **Use multiple views of the requirement**
- **Rank the requirements**
- **Work to eliminate ambiguity**

Information Domain

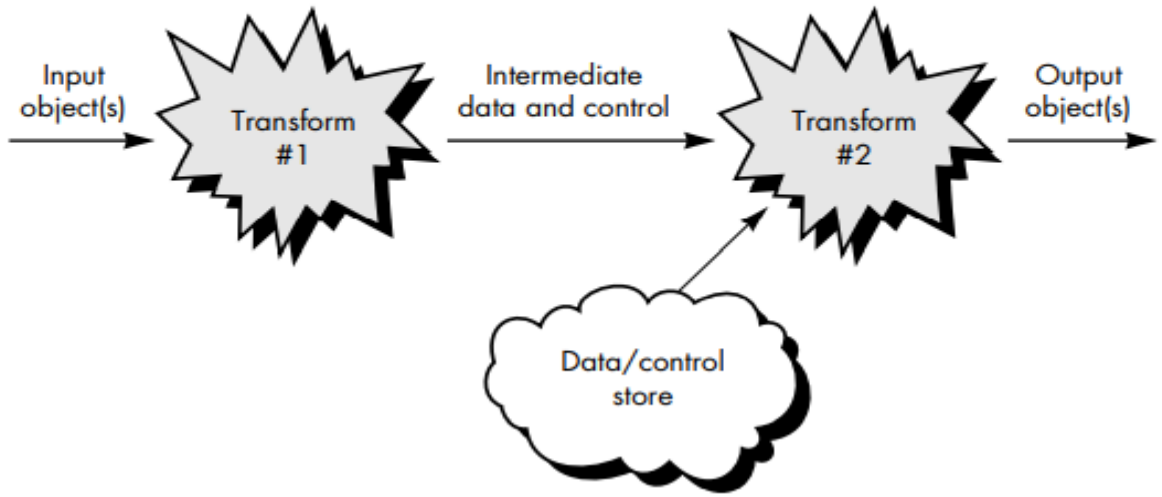
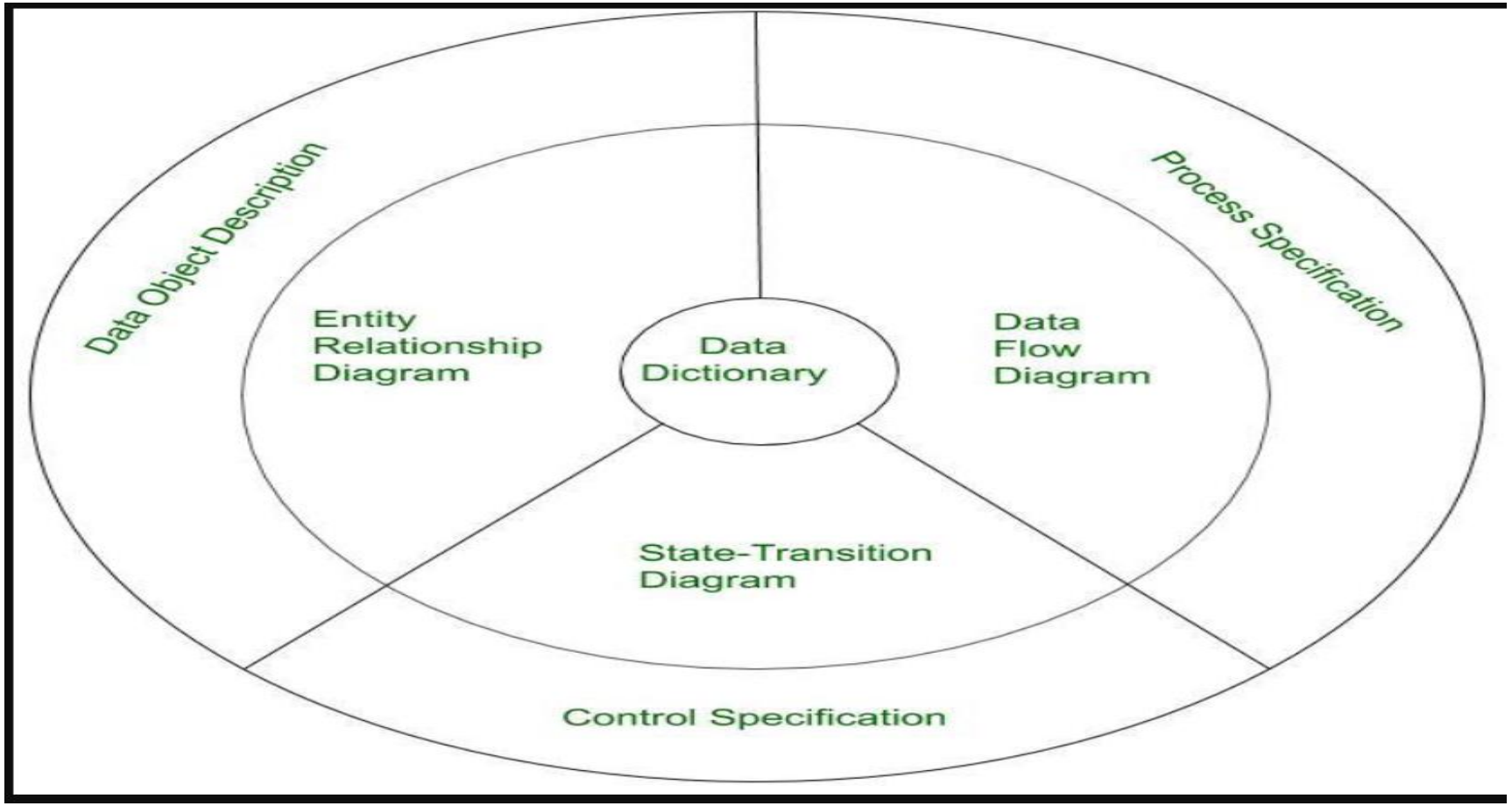


FIGURE 11.3 Information flow and transformation

Elements of Analysis Model



Objectives of Analysis Modelling:

- It must establish **a basis** for the creation of software design.
- It must **describe** requirements of customer.
- It must define set of requirements which can be validated, once the software is built.

Data Dictionary:

- A **repository** that contains description of all the data objects consumed and produced by the software.
- Three different diagrams surrounds the core.
- The **ERD** depicts the relationships between data objects
- The **ERD** is the notation that is used to conduct the data modelling activity.
- The attribute of each data object in the ERD can be described using **data object description** .

State Transition Diagram

- The state transition indicates how the **system behaves** as a consequences of external events.
- STD represents the various modes of behavior called the state.
- Information about the control aspects of the software is contained in the **control specification**

Data Flow Diagram

DFD provides the two purpose

1. To provide the an indication of how data are transformed as they move through the system.
2. To depict the function and sub functions that transform the data flow.

A description of each function presented in the DFD is contained in a **process Specification (PSPEC)**

Data Modelling -> ER Diagram

- Data Objects
- Attributes
- Relationships

Example: A **person** owns the **car**

Modeling

We create functional models to gain a **better understanding** of the actual entity to be built.

- **Data model**
- shows relationships among system objects
- **Functional model**
- software converts information and to accomplish this, it must perform at least three common tasks- input, processing and output.
- When functional models of an application are created, the software engineer emphasizes problem specific tasks.
- The functional model begins with a single reference level model (i.e., name of the software to be built). I
- n a series of iterations, more and more functional detail is given, until all system functionality is fully represented.

Models

Behavioral Modelling

- A computer program always exists in some state – an externally observable mode of behavior (e.g waiting, computing, printing) that is changed only when some events occurs.
- Describe manner in which software responds to **events** from the outside world

Partitioning

- Process that results in the elaboration of data, function, or behavior.
- **Horizontal partitioning** ◦
- breadth-first decomposition of the system function, behavior, or information, one level at a time.
- **Vertical partitioning** ◦
- depth-first elaboration of the system function, behavior, or information, one subsystem at a time.

Requirements Views

Essential view

- presents the functions to be accomplished and the information to be processed while ignoring implementation

Implementation view

- presents the real world realization of processing functions and information structures

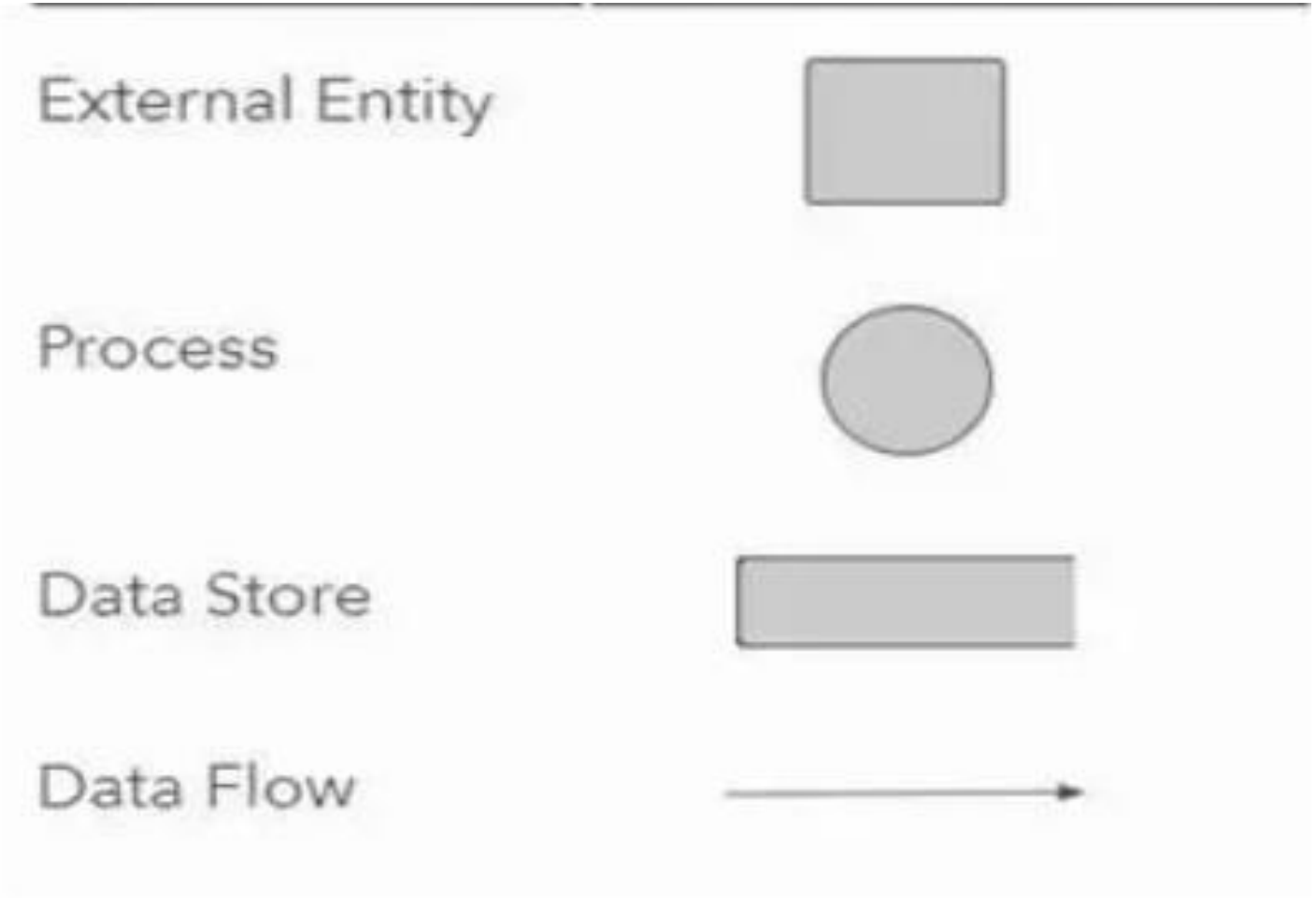
Functional Model

- The DFD many be used to represent a system or software at any level of abstractions.
- DFDS may be partitioned into levels that represents increasing information flow and functional details
- Therefore, the DFD provides a mechanism for functional modelling as we as information flow modelling
- A level 0 DFD, also called a fundamental model or a context model, represents the entire software element as a single bubble with input and output data indicated by incoming and outing arrows, respectively

Representation of Data Flow

- External Entity is :An external entity can represent a human, system or subsystem. It is where certain data comes from or goes to system Represented by Rectangle
- Process : A process is a business activity or function where the manipulation and transformation of data take place. Represented by circle
- Data Store :A data store represents the storage of persistent data required and/or produced by the process.
- Data Flow :A data flow represents the flow of information, with its direction represented by an arrowhead that shows at the end(s) of flow connector.

Symbols used in Data Flow Diagram



Levels of DFD Diagrams

1.Level 0 DFD

Also known as Context Diagram or **Functional System Model :**

Represents the entire software elements as a **single bubble** with input and output data indicated by incoming and outgoing arrows.

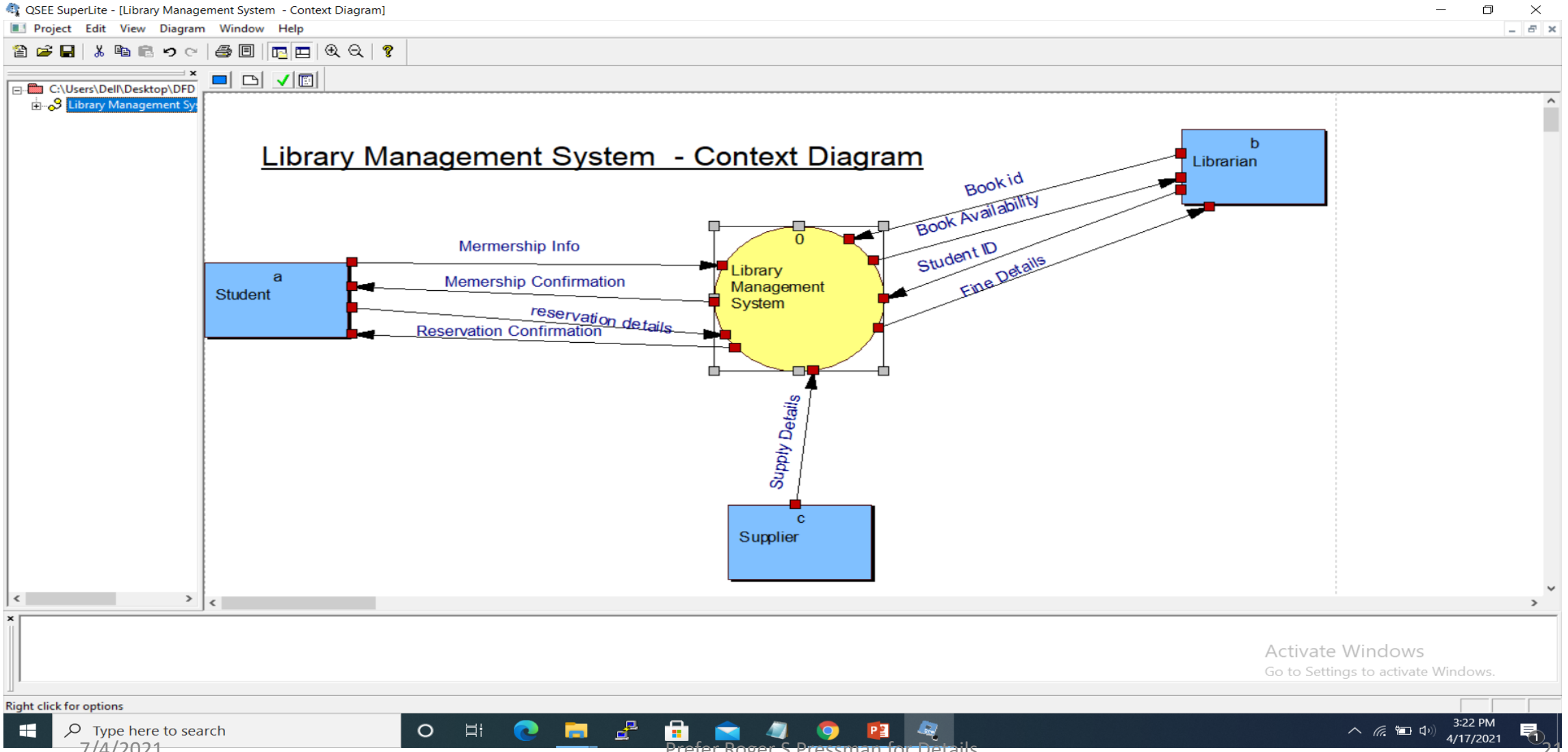
2. Level 1 DFD :

- are still a general overview, but they go into more detail than a context diagram.
- In a level 1 data flow diagram, the single process node from the context diagram is broken down into sub processes.
- As these processes are added, the diagram will need additional data flows and data stores to link them together.

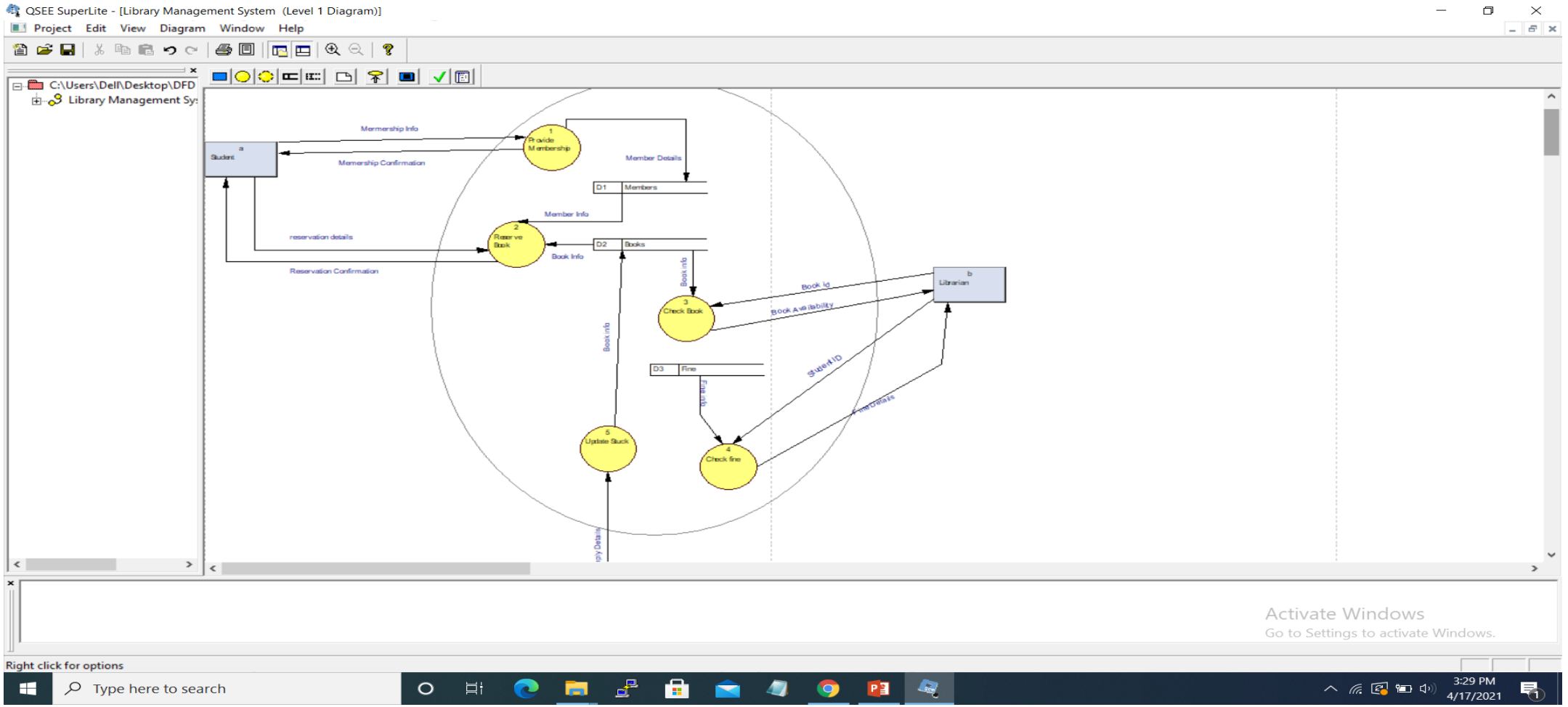
3. Level 2 DFD

- simply break processes down into more detailed sub processes.

DFD 0 or Context Diagram



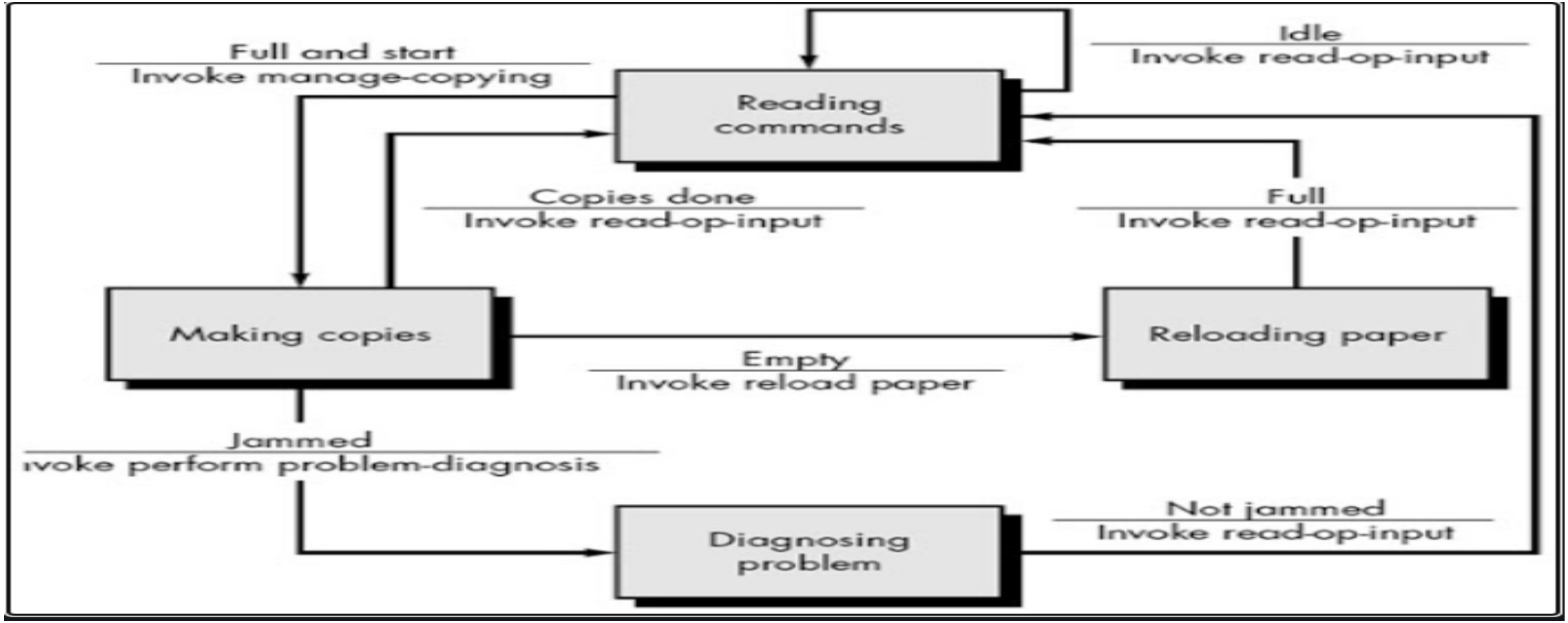
DFD level 1


QSEE SuperLite - [Library Management System (Level 1 Diagram)]
Project Edit View Diagram Window Help
C:\Users\Dell\Desktop\DFD
Library Management Sy
Mermership Info
Mermership Confirmation
Student
1 Provide Membership
Member Details
D1 Members
Member Info
reservation details
2 Reserve Book
D2 Books
Book Info
Reservation Confirmation
Book info
b Librarian
Book Id
Book Availability
3 Check Book
Book info
D3 Fine
Fine info
StudentID
Fine details
5 Update Stuck
4 Check fine
Activate Windows
Go to Settings to activate Windows.
Right click for options
Type here to search
3:29 PM
4/17/2021

Behavioral Modelling

- **state transition diagram** are used to represent the behavior of the system to various external conditions and inputs.
- It graphically represents how a system in one state switches to another state on the action of external stimuli and control signals.
- A **state** is any observable mode of behavior.

Behavioral Modelling



Behavior Modelling

- System states are represented by **a rectangular shape by rounded corner** and arrows are used to represent transitions between states.
- Each arrow is labelled as a ruled expression A/B.
- Top value represents the event responsible for the transition and bottom value represents the action that occurs as a consequent of the event during transition.

Behavioral Modelling

- Fig shows a state transition diagram for photocopier software.
- Photocopies software will make photocopies only when it not in jammed state and exists in start state.
- If it is in jammed state, then the problem is diagnosed and removed.
- If photocopier is empty means do not have blank papers then the papers are reloaded.
- In the above two conditions. Start commands are given again.

Data Dictionary

- The data dictionary is an organized listing of all data elements that are pertinent to the system.
- Data are organized with precise, rigorous definitions
- Both user and system analyst will have a common understanding of inputs, outputs, components of stores and even intermediate calculations

Data Dictionary

- **Name**—the primary name of the data or control item
- **Alias**—other names used for the first entry.
- **Where-used/how-used**—a listing of the process and how it is used
- **Content description**—a notation for representing content.
- **Supplementary information**—other information about data types, preset values, restrictions or limitations, and so forth.

Data Dictionary

- The notation used to develop a content description is noted in the following table:

Data Construct	Notation	Meaning
	=	is composed of
Sequence	+	and
Selection	[|]	either-or
Repetition	$\{ \}^n$	*n* repetitions of
	( )	optional data
	* ... *	delimits comments

Example of Data Dictionary

- **name**: telephone number
- **aliases**: none
- **where used/how used**: dial phone (input)
- **description**: telephone number = [local number / long distance number]
- **local number** = prefix + access number
- **long distance number** = 1 + area code + local number
- **area code** = [800 | 888 | 561]
- **prefix** = *a three digit number that never starts with 0 or 1*
- **access number** = * any four number string *